Administration Guide

The ExtraView Administrator's Guide is designed to give users of the ExtraView web-based business issue tracking system the knowledge and proficiency needed to accomplish two general goals:

1. Customize ExtraView to conform to their company's workflow processes, business rules, and inter-organizational vocabulary, with all the appropriate data fields and security privileges that are required.
2. Give Administrators the ability to successfully administer ExtraView on an ongoing basis in order to efficiently respond to user inquiries, concerns, and requests.

In writing this guide, we anticipate that the reader is at least somewhat familiar with many of the standard issue tracking and business management functions found in ExtraView. Accordingly, this guide will assume this familiarity, and will focus on describing the administrative functions of ExtraView.

Downloadable PDF

This Administration Guide is downloadable as a single PDF by clicking here. You will need the Adobe Acrobat Reader to view this.

ExtraView Versions

There are 3 ExtraView versions, the **Standard**, the **Professional** and the **Enterprise** versions. with the **Enterprise** version all features may be configured without limitation. The **Professional** version has some limitations, and the **Standard** version has the most limitations. With the payment of appropriate license fees, you may upgrade your **Standard** version to either the **Professional** or the **Enterprise** version. The **Professional** version may be upgraded to the **Enterprise** version. It is not possible to downgrade versions. The limitations of the different versions can be seen here.

Key Concepts

ExtraView is Web-based issue-tracking and workflow software that is designed to meet the following objectives:

- Easy to install, configure and administer, minimizing your organization's setup and ongoing cost of ownership
- Provide functionality that is easily extensible over time
- Able to support your business processes and your workflow, with straightforward configuration
- Capable of implementing multiple tracking systems, all within a single database, with each tracking system operating independently, or inter-related with the other tracking applications (multiple tenants)
- Scalable to support large numbers of users and issues
- Easily configured to reflect your company's terminology, and data hierarchies, and able to provide extensive validation for data that describes your organization, products, and services
- Provide varied and extensive reporting and charting
- Easily integrate with other enterprise software systems
- Be extensible by adding additional code into the environment, **without changing the base product**

To understand the key concepts in ExtraView, it is recommended that you read through the pages referenced below.

Installation & Configuration

This process is best executed with advanced planning. The purpose of this guide is to give you complete details on the administrative portion of the initial setup as well as ongoing support for your installation. Please consult the ExtraView Installation Guide for your specific platform for full details on the installation of the servers and ExtraView application. ExtraView Corporation's technical support personnel can help with recommendations for suitable platforms. You can view the server requirements by clicking here.

New Installations Downloaded Directly from the ExtraView Website

If you are using the standard ExtraView product, downloaded directly from our website installation is very simple and includes all the components required for a fully functioning system, including a database. Just follow the download instructions.

ExtraView Installed with your own Server and Database

If you are installing and configuring ExtraView with your own database and servers, you can view the Installation Guide here.

Configuration

This Administrator's Guide covers the design of your system and how to provide the features you need for your company. Briefly, the tasks that need to be accomplished are as follows. Note that many of the tasks do not need to be performed in sequence, but can be executed in an arbitrary order.

- Install the license key provided by ExtraView
- Create a user account with administrative privileges for your own use
- Set up a small number of global behavior settings. These will be the foundation to the successful running of ExtraView in your environment
- Configure any special connectivity needs to remote databases, such as LDAP or Active Directory for remote directory services and / or SSO for single sign on authorization
- Define and create the user defined fields in your system, that will complement the inbuilt fields
- Define the relationships between record types and fields, such as how different record types are related and where parent child-relationships exist between list values
- Define and implement the various user roles, or categories of users who will access the system
- Design and lay out screens to support the fields you create
- Create a structure of permissions that support access to each screen for each user role that was defined
- Set up the workflow that will control the processes in your company. For complex workflows this may involve the design and programming of customized "user exit" routines, written in the Java or JavaScript languages
- Design standard reports that will be "public" for your users
- Add user accounts to the system
- Test the completed system

Defining Your Process

ExtraView allows the System Administrator to define a process that conforms to the way the company works. It does not impose a fixed methodology on the company. The administrator can, without programming, set up rules appropriate to the company's needs.

Each issue you submit can be moved between any number of status values that you define, with each status being visible only to the user role that is permitted to work on the individual status. For example, an *Open* issue may only be changed by the Engineering group, who may only mark it *Fixed* or *Issue* not found after working on it. This same Engineering group may not *Close* the issue since that is a state that only a different role such as Quality Control may access.

A user role is created for all people who should follow the same rules. Typically, these would fall along the lines of customers, support staff, engineering, quality assurance, managers, administrators, etc., but complete flexibility exists to define what user roles you create and how many you create.

In addition, you can program a significant amount of logic into each **Add** or **Edit** screen form within ExtraView. For example, you can set some fields to only display dependent on the contents of a specific field, or you can specify sub-layouts within the form that are conditional upon the value of a specific field.

Should you have a workflow process that cannot be accommodated within ExtraView's standard functionality, the product can be extended with additional code written in the Java language and inserted into a "user exit" routine using ExtraView's UserCustom Class. ExtraView was designed to make it easy to alter or add functionality within the source code, but without needing to modify the source code of the base product. In this way, your investment is protected when you update or upgrade ExtraView.

Customizable User Interface

ExtraView can be modified in a number of ways in order to tailor its look and feel to any company's needs.

The following changes can be made simply, either by you or by ExtraView Corporation's Professional Services team:

- Alter screen colors and fonts
- Add your company logo
- Edit all text labels to reflect your own terminology
- Rename menu items
- Create new navigation buttons in any style / color
- Create alternative screen layouts for different groups of users
- Create alternative report and email layouts for different user roles

To simplify the creation of a good-looking interface, there is an administrative utility that allows you to select a theme from a library of preconfigured options.

Note that ExtraView has been designed to support users with disabilities. In particular, there is broad support for the US Section 508 of the Rehabilitation Act of 1973, as amended in 1998. However, it is the administrator's responsibility to configure ExtraView to their organization's needs to support Section 508. For example, when an administrator creates a new field, they should ensure that it has a tooltip with appropriate content that can be used by screen readers. Similarly, the administrator should select or create color themes that provide for color-blindness.

For more information, please see the following pages:

- Section 508 - Disability Support
- VPAT for ExtraView

Users

Users are specific people who have an authorized sign on capability to ExtraView. Each user has an individual account and the user or a system administrator can modify personal behavior settings within the account.

The administrator has additional privileges to manage user accounts. For example, they can create new accounts, disable existing accounts and can add and remove users from user roles and privacy groups.

User accounts may be created by a self-registration process. However, if a user self registers, they are only given basic access until a system administrator grants them additional rights.

There are two special accounts within the system, named *admin* and *system*. The *admin* account has the following characteristics:

- It should not be used to create or manage any issues
- Issues cannot be assigned to the *admin* user
- The *admin* user is not counted within the number of available licenses
- The *admin* user does not view the sign on message area, allowing them to bypass any potentially bad HTML that was inserted into this area by an administrator
- The *admin* user has additional privileges in the data dictionary, where they may enter and update the SQL statements that are used by ExtraView to extract and update information in the database
- The *admin* user bypasses all status change rules.

The *system* account is reserved for use by ExtraView personnel, in the eventuality that any customer of the product has locked themselves out of their system, and ExtraView personnel need to restore access. If you accept the risk, and understand that ExtraView support may not be able to access your system quickly, you can ask ExtraView support to give you ownership of this account.

Users may be defined and maintained in an external directory, such as an LDAP directory or a Microsoft Active Directory. Various policies for synchronizing the external directory and ExtraView's internal directory may be adopted.

Users may sign on more than once and they will still occupy a single license within ExtraView. However, there is a limit for each unique user in that they may only sign in and have up to five active sessions. For each five active sessions, one license is consumed.

User Roles and Security

Individual users belong to one or more user roles and share the same characteristics and permissions. For example, one user role may have read and write access to a particular field, while another role may only be able to view the same field. Another user role may not even be able to see the field.

With the Enterprise version, you may create an unlimited number of user roles. Other versions have constraints on how many roles may be created.

All fields, menus and screens have security keys to protect the object. For example, a security key exists for accessing the security module itself. Another security key exists for the *product* menu item on the Administration screen. Another example is that a security key exists to control access to the *description* field. All of the fields that are visible can be turned on or off based on user role privileges. There are literally hundreds of security keys within ExtraView, and each time you create a new user-defined field, two new security keys are automatically created to allow you to protect the newly created object. These two keys allow you to control access to the field when adding a new item to the database and to control access to the field when either updating or reporting.

The Grant Security Privileges section controls all these accessibility features in your version of ExtraView. In a matrix view, the intersection of the security key and the user role is a read and write switch. Therefore, for every item that has a defined security key, you can allow or prohibit any user role to access that feature.

Business Areas

This key concept is used to define a structure where you may define multiple tracking systems within a single instance of ExtraView. Each of these tracking systems is termed a Business Area, and each of these Business Areas may have its own set of screens, fields, processes, rules and workflow. Business Areas may also share these objects. An administrator may have control over the entire installation or sub-administrator roles may be created, each having control over a part of the installation. With this capability, ExtraView is termed a multi-tenanted database.

For example, the Business Areas can correspond to issue tracking processes such as defect tracking, requirements planning, customer issues, adverse event tracking, change management and/or safety issues. Subject to any restrictions of the ExtraView licensing scheme purchased, the administrator may create as many Business Areas as required. Within each Business Area, multiple Projects can also be defined, again subject to any restrictions of the ExtraView license purchased. These Projects have the same characteristics as Business Areas. Projects will typically be used to provide alternative processes or screens or workflow within a defined Business Area.

The administrator can limit a user's access to the issue-tracking database to individual Business Areas, and to individual Projects.

Different screen layouts and default reports with different fields can be designed for each Business Area and each Project, as well as different workflow and business rules, thus ensuring the optimization of ExtraView for each part of the organization.

Using common fields, management information can still be consolidated across Business Areas or the entire organization.

The fundamental principle that governs the use of Business Areas and Projects is inheritance. The top-level Business Area within ExtraView is termed the Global Area. The Global Area has a single Project, named the Master Project. Layouts and security permissions always exist for the Global Area and its Master Project. Beneath the Global Area, further Business Areas and Projects can be defined, where inherited values apply. Each of these child Business Areas has a Default Project, and may have any number of individual Projects. At any level, alternative layouts, fields and security permissions for fields can be defined, over-riding the inherited values.
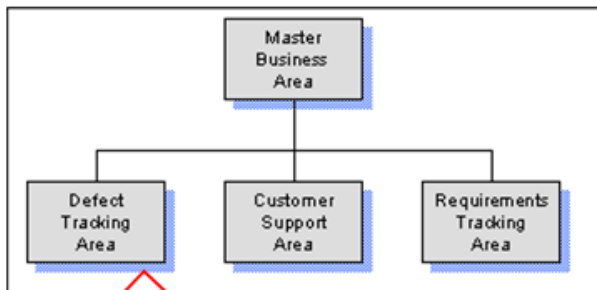
Example Business Area Structure
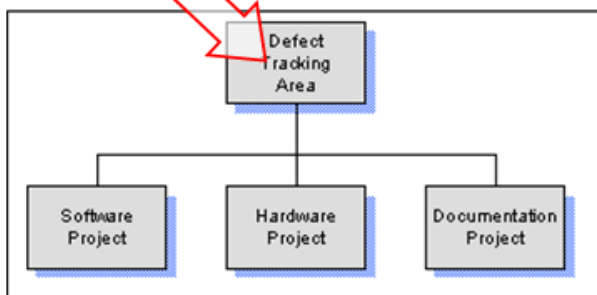
Example Project Structure within a Business Area

All ExtraView's inbuilt fields are global in scope. Fields you define yourself as an administrator may be global in their scope or may be defined for a single Business Area, or defined for a single Project within a single Business Area.

Many of the options in this guide are dependent upon Business Area and Project. However, for clarity, reference is only made within this guide, when it is important to explain a significant fact. Simply, the Business Area and Project will appear on administration screens such as security permissions and layouts, when they are required, and will not be present if the Business Area and Project capability is turned off.

Another key attribute of Business Areas is that you may create relationships between items (or issues) stored in different Business Areas. This is an extremely powerful feature that allows you to define multiple record structures that can interoperate with each other. These relationships can extend many levels in a hierarchical structure. A simple example may be that you store "customers" in one

Business Area, and that you store "customer issues" in a separate Business Area. You may then relate these two areas together, based upon a common attribute such as the customer name. I.e, when a customer issue is entered, the customer name is selected and all the other attributes of the customer are then visible within the customer issue record. When reporting, you can use the relationship to show fields from the "customer" record along with fields from the "customer issues".

Note: You should not store data for your installation within the Global Business Area and Master Project. These should be reserved for the setting of default security permissions and layouts to provide the inheritance to all other Business Areas and Projects. This gives you the most flexibility in maintaining and extending your system.

Note: If you are configuring ExtraView for a single purpose, such as defect tracking, we do not recommend that you simply configure the Global Business Area, and use that to store your data. Although this will work, it will make future changes difficult, should you decide to introduce additional tracking systems.

The Enterprise version of ExtraView allows an unlimited number of Business Areas and Projects to be created. Other versions have constraints on the numbers that may be created.

Business Areas and Projects can be turned on and off with the behavior settings named ENABLE_AREAS and ENABLE_PROJECTS. These are found on the Environment Settings menu within Administration. **It is recommended that you never alter the default settings for these behavior settings.**

Queries

Queries may be formed and executed on any issue data stored within ExtraView.

Queries are composed using either standard or advanced filters. The results are presented using either a Quicklist or a Detailed Report. The results may be output to the user's browser, to Microsoft Word, Microsoft Excel, Adobe PDF or to text.

The standard filters require the composition of a layout of the type SEARCH_QUICK. This should contain the most frequently used fields for queries. Users can select any number of filters on this layout to compose a query. When multiple filters are selected, the query engine places an "AND" conjunction between the filters. The user can select expanded query filters, in which case, multiple values can be selected within each field. This is an "OR" operation with the values of the filter field.

Advanced filters require the selection of the filters to be used in the query, one-by-one. However, the user can select from the entire range of fields to which they have read permission. The user may also set a conjunction between each filter, using "AND", "OR", "UNION", "INTERSECT" or "MINUS". Advanced filters take more time to set up, but offer significantly more capability.

Once prepared, filters sets may be stored and recalled.

Within the administration section of ExtraView the most important aspect that affects querying is to correctly set the following Data Dictionary fields, as explained in the Data Dictionary section of this guide:

- Allow selection on reports
- Total field on reports
- Filter criteria
- Is sortable

Reports

Reports are composed of two basic elements, a set of query filters, and a definition of the data to be output. There is a wide range of report types, as explained in the End User Guide. The results may be output to the user's browser, to Microsoft Word, Microsoft Excel, Adobe PDF or to text.

Each report has filters, which are composed using either standard or advanced filters. The standard filters require the composition of a layout of the type SEARCH_QUICK. This should contain the most frequently used fields for queries. Users can select any number of filters on this layout to compose a query. When multiple filters are selected, the query engine places an "AND" conjunction between the filters. The user can select expanded query filters, in which case, multiple values can be selected within each field. This is an "OR" operation with the values of the filter field.

Advanced filters require the selection of the filters to be used in the query, one-by-one. However, the user can select from the entire range of fields to which they have read permission. The user may also set a conjunction between each filter, using "AND", "OR", "UNION", "INTERSECT" or "MINUS". Advanced filters take more time to set up, but offer significantly more capability.

For most report types there is the capability to define hierarchies for the filters, allowing the selection of parent and child records for output to the report.

Within the administration section of ExtraView the most important aspect that affects querying is to correctly set the following Data Dictionary fields, as explained in the Data Dictionary section of this guide:

- Allow selection on reports
- Total field on reports
- Filter criteria
- Is sortable

If you are using relationships within your installation, reporting hierarchies may be created, allowing parent / child / grandchild type reporting.

Fixed Names & Screen Titles

These terms are used widely throughout the administration guide.

Fixed names are defined as terms that are used within the ExtraView database to refer to a field or object. Once created, names for an object do not change and they are a fixed reference. Each name will have a corresponding screen title. If you have turned on the localization feature of ExtraView, there may be more than one title for any named object, i.e. there may be one title for each locale or language. Screen titles for any fixed name may be changed by the administrator.

The screen title is defined as the reference to an object by which it is referred to throughout the user interface to the end user of ExtraView. Thus, every title within ExtraView may be altered, but the underlying name that it refers to will not alter.

In this way, the title (or label) that refers to each field and object can be changed at will by the administrator, but the underlying data remains without change. For example, the field with the name of **ID** may have its title changed from *Defect #*, to *Tracking Number*. From the moment of this change, all screens, reports and other screens that refer to **ID** will use the new title.

Changes made in this way to metadata by administrators are logged by ExtraView, giving an audit trail of who made what change, and when they made it.

Fixed names used for any object type are unique within ExtraView Names can only consist of the characters A to Z, 0 to 9, and '_'. The first character of a fixed name must be alphabetic and the name can be up to 30 characters in length. Names cannot contain characters from non-English alphabets. Further restrictions are that you may not have two underscore characters together in a name and the last character may not be an underscore.

Screen titles are not required to be unique for an object type across an ExtraView installation. However, consideration should be given to using non-unique titles. In some installations, this makes perfect sense, in others it may not. Titles can consist of most characters, but "special" characters such as '!', '"', '#', '$', '%', '&', '"', '(', ')' '@', '~', ':', ';' and '`' may not work in all places. For reliability, you should use alphabetic characters only in titles. However, titles can be localized and may contain characters from any alphabet, including double-byte character set alphabets.

With a small number of exceptions, you may not insert HTML into a screen title. This is to preserve security where a user may have the ability to alter a title and uses this ability to inject HTML that consists of a script into a screen. Such scripts may not be innocuous and therefore the ability to introduce HTML of any type into a title is restricted.

Note that if you take advantage of this feature, then any output through the API or CLI will contain and display the embedded HTML within the screen title.

Data Dictionary

The Data Dictionary is the central place where all field definitions are stored and maintained. All User Defined Fields (UDF's) are defined in the Data Dictionary. In addition, this core component of ExtraView controls many of the attributes of each field, such as where it is used, its display type, display title, whether the field is selectable on reports, its default value, default attributes and help text. Global attributes may be defined for a field. These optional attributes determine properties of the field which can be inherited wherever the field is used. For example, you might provide a SIZE global attribute to a text field. Wherever this field is used as input, it will then adopt this SIZE, although the size might be overridden on an individual form.

Correct settings in the Data Dictionary are essential to a smooth running system. Although it is possible to alter every field within the ExtraView, it is recommended that you only make changes when you thoroughly understand what the consequences will be.

Design Center

This feature is the principal area where you will configure your system.

This feature allows you to configure the layout of the *Add Issue*, *Edit Issue* , *Query* and other screen layouts. In addition, key reports such as the *Quicklist* and the *Detailed Report* also use layouts defined within the Design Center. Layouts can be defined for different user roles within your system, offering a tremendous amount of flexibility. You may create and define fields within the Design Center, or you may use the Data Dictionary for this purpose.

Layouts work in conjunction with security permissions for each field. Therefore, simply placing a field on a screen does not automatically give all users the ability to read or write to the field. You can alter the permissions to each field within the Design Center, or you may use the **Grant Security Privileges** option to define which fields are visible and updateable to each group of users. The security privilege for the field overrides the fact that a field may be placed on a screen or report.

One layout may be embedded within another layout. In addition to this, you may specify alternate layouts that appear, dependent upon the value of a specific field. For example, you may have a category list field that has the values of *Software*, *Hardware* and *Documentation*. Depending on which value is chosen, a sub-layout can be displayed that contains the fields pertinent to gathering the information needed about each of these categories. These sub-layouts or embedded layouts may only be embedded within an *Add* or *Edit Issue* layout. There is no need for them on other layouts.

Fields within each layout may have one or more attributes defined. These layout cell attributes affect the display of the field, or the way in which it is processed. For example, an attribute may provide the field with an alternate title, just for the one layout, or the attribute may define that the field is only visible if another field is of a specific value.

There are two special type of layouts which may only be embedded within other layouts. These allow the definition of **Related Issue Layouts** and **Repeating Row Layouts**.

Workspaces

Workspaces provide a single browser window within which you can run all of ExtraView's end-user functions. Within the browser window, a separate panel will be opened for each function. For example, you might open an *add* screen, an *edit* screen, and several reports, all at the same time. Each panel has a title bar that contains buttons to control the functions within the window, as well as buttons to minimize, maximize, and close the panel.

A typical workspace

A user may have any number of workspaces, and each may contain different panels. The user may save the state of the workspaces, and these will be restored when they reopen each workspace. The workspace is provided as an alternative user interface to the traditional user interface. With permission, the user may start their ExtraView session in the traditional interface, or within the workspace interface. The user may also switch between the interfaces. At this time, the ExtraView Administration functions will not appear within a workspace window and will always use a separate browser window.

Email Notification

Whenever issues are inserted or updated, email notification may be triggered to send information to users connected with the issue. These notifications are typically used to inform users such as the originator, the assigned to and the person who last updated the issue.

Interest lists allow users to subscribe to issues where they have a particular connection. These are a powerful feature of ExtraView, ensuring automatic notification of events to appropriate individuals within an organization. Interest lists can be defined within the Data Dictionary on any field that may have a list of values. For example, you may define an interest list that notifies individuals on all issues that have a severity level of *critical*, or you may define an interest list that notifies individuals whenever an issue affects a particular *module*.

Rules may also be defined to override the standard notification rules.

Notification may also take place which escalates issues which have remained in a particular status for longer than a given period, or are approaching or have exceeded a date such as the due date of a repair.

Programming Interfaces

RESTful Application Programming Interface

This web-orientated API to ExtraView allows the user to extend ExtraView's functionality. The key features of the RESTful (Representational State Transfer) interface are:

- A set of URL functions utilizing the HTTP protocol that access ExtraView to perform user functions such as adding and updating issues. This eliminates the need for expensive software such as Oracle SQLNet on each client computer
- A set of URL functions that allow limited administration of ExtraView. For example, you can add users and alter user passwords
- A full Command Line Interface (CLI) implemented on top of the API, that allows users to access functions such as adding, updating, deleting and searching from a command line. This is typically used from a UNIX, Linux or Windows 2000 command shell

Web Services Application Programming Interface

This Simple Object Access Protocol (SOAP) interface provides a complete set of API functions, similar to the WOA interface, but utilizing the more standardized methods of a SOAP interface.

The interface may be accessed through both Java and .Net. A full set of examples is provided with an interface implemented on top of Visual Studio.

Command Line Interface (CLI)

The CLI is implemented using the ExtraView API. It comprises a set of Perl scripts that perform many functions such as:

- Adding or updating issues in a batch or interactive mode from a telnet session
- Performing limited administrative functions
- The CLI functions can be further scripted by an ExtraView user, to create their own functions, running in interactive or batch mode.

Please see the following guides for further information:

- [Application Programming Interface Guide](#)
- [Web Services Interface Guide](#)
- [Command Line Interface Guide](#)

Client Browser Support

Supported Desktop Browsers

ExtraView for this release is certified to support the following browsers.

- Microsoft Internet Explorer is supported using versions 9, 10, 11
- Microsoft Edge
- Mozilla Firefox is supported using version 25 and greater
- Apple Safari (on Macintosh only) is supported, using version 10.1 or greater
- Google Chrome is supported, using version 30 or greater.

These browsers are supported on the following platforms (where available) – Windows 8.1, 10.0, Apple Macintosh, and Linux. The exception is Apple Safari which is not supported on Microsoft Windows platforms.

Screen Resolution

The resolution of desktop monitors or screens on which users should use ExtraView should be a minimum of 1280 x 1024 pixels. While ExtraView will work at lower resolutions than this, users may have to scroll up, down and sideways more than they would like. The administrator can make decisions in the design phase that will affect this. For example, the settings in the style sheets that control the small, medium and large font sizes can be tailored, as can the number of columns and rows that appear on screen layouts. If the users are utilizing the Workspace feature, the recommendation is to use screens with as high a resolution as possible, to maximize the number of viewable panels. The recommendation is that administrators should design their layouts to fit within a resolution of 1280 x 1024 pixels.

Cookies in the Browser

Most browsers have cookies turned on as a default setting and this is what ExtraView expects. If cookies are not turned on, ExtraView will warn the user , and will not function until they are turned on. ExtraView primarily uses session cookies, and no cookie or other information is stored within the client browser after the ExtraView session is complete.

The only exception to this is when the administrator enables the behavior setting ALLOW_CHANGE_LOCALE_AT_SIGNON. In this case, a cookie retains the preferred locale setting for the user who signs on for a 30 day period.

JavaScript in the Browser

JavaScript must be turned on within the client browser.

The Browser Back Button

Users should never use the browser's back button within ExtraView. They should only navigate by the buttons that are displayed on ExtraView's menus. The reason is that ExtraView must maintain integrity of its information at all times. For example, if you press the button on ExtraView's *Add Issue screen* to add a new record, then pressed the back button and pressed the add button again, you will have two records inserted. Similar problems occur if the user attempts to go back to a record he has edited, or to go back to a screen that was refreshed from the server during adding or editing an issue.

The Browser Refresh Button

Similar to the Back button, the Refresh button within your browser should not be used. At the times that a Refresh is allowable, ExtraView will offer a Refresh button in its menu bar.

Character Sets within the Browser

ExtraView must work consistently with a single character set, in order that information entered within different browsers across an organization will be compatible, and can be stored on and retrieved from the ExtraView server in a consistent manner. This is less of a problem with languages based on the Roman alphabetic, but is an essential ingredient of correctly configuring a system where users use double-byte languages such as Japanese and Chinese. There is a behavior setting (see the following section) named HTTP_CHARSET that defines the character encoding for input from all browsers, for all users of the ExtraView instance. By default this is set to UTF-8, a character set that is universal and supports all languages. The recommendation is to have all users set their local browser character set to UTF-8. If this is to be changed, then it must be changed on the server and in every client browser.

Note: It is strongly recommended that HTTP_CHARSET is set to a value of UTF-8, and that all users only set their browser character set to UTF-8, so that characters will be displayed correctly and consistently.

Administrative Functions

Administrative functions are split into the following groups:

- **Operational Tasks** - These are the day-to-day tasks such as administering users and setting up escalation tasks
- **Site Configuration** - This is where the majority of operations that configure your site are managed
- **Initial Setup** - These are typically the features that need to be set up in a new installation
- **Import/Export** - This is where you export and import data to and from ExtraView
- **Advanced** - These are advanced features that only need to be accessed on an occasional basis.

There is some crossover in the groups of functions. For example, the administrator will need to occasionally access a Site Configuration feature for a day-to-day operation.

Mobile Device Configuration

ExtraView supports mobile platforms. Mobile clients can be downloaded from the Apple App Store, or from Google Play, depending on the device you are running. Given that layouts designed for desktop usage may well be too wide and too tall to be used without a significant amount of scrolling on a mobile device, ExtraView provides the capability to create layouts for different display targets as follows:

- **Desktop** - these layouts are the default and will always be presented to a user if no Tablet or Phone layout exists. Given that there are always a set of default Desktop layouts in a system, this means that a layout will always be presented to a mobile user,

even if it has not been optimized for use
- **Tablet** - these layouts will be presented to a user when they have been configured, and ExtraView detects it is running on a tablet-sized device. These layouts may have fewer rows and fewer columns than a Desktop layout, depending on how many fields need to be placed on the layout. The end user can override the Tablet display target selection and elect to use either the Desktop or Phone layouts, if they desire, within the sign on information for the connection
- **Phone** - these layouts will be presented to a user when they have been configured, and ExtraView detects it is running on a smart phone-sized device. Typically these layouts will have fewer rows and only one or two columns. The end user can override the Phone display target selection and elect to use either the Desktop or Tablet layouts, if they desire, within the sign on information for the connection.

Mobile configurations may also be implemented to be more restrictive than the desktop configuration, for example to only allow the end user to add a specific issue and no more, often without the need for the user to sign on with a valid User ID and password. See the section further down this page, entitled **Configuring Limited Access**.

Implementation Considerations

You can globally turn mobile access on and off with the behavior setting named ALLOW_MOBILE_CLIENTS

Given the smaller screen sizes of Tablet and Phone devices, you will almost certainly want to configure different layouts for these. Here are some recommendations:

- Do not configure the mobile layouts until you have completed the configuration of the Desktop target layouts. This allows you to debug your workflow and other logic first, then simply do a "save as" type of operation to create the Tablet and Phone layouts from the Desktop layouts. Then it is a relatively simple exercise to alter the presentation of the layouts, while retaining the underlying logic. If your Desktop layout does not have more than 3 columns and more than about 10 rows, then Tablet users may often be able to use this as their layout. It is recommended that you configure your Phone layouts with a single column and perhaps 5 or 6 rows per screen. See here for instructions on creating layouts
- Consider the introduction of screen pages within your mobile layouts. These are configured using the PAGE_PRE_xxx and PAGE_POST_xxx fields. These provide a neat interface to the user, only displaying a screenful of fields with *next* and *previous* page buttons to move between the different parts of a layout. These are most useful on Phone layouts. See here for instructions on using pagination
- Many reports will be overly large to display on mobile devices, particularly on mobile phones. For each report type that is enabled for mobile working, you can choose within the individual report editor screens to allow or disallow the user to see and run that report on a mobile device. While you may allow the end user to decide for themselves which personal reports they should be able to view, consider the Public reports and whether they are suitable for mobile viewing. You might create some reports just for mobile working. You can also use the **Administrative Report Management** utility to review and update all the reports to decide which you want to make available to mobile users. If you are familiar with the public reports, this is a faster way to turn on reports for mobile working, compared to individually reviewing the reports
- It is worthwhile introducing specific navigation bars for mobile working. These can direct users to the specific pages that are important for their use. In addition, you might introduce a report menu, with access to a limited number of reports specifically designed for mobile working, and not allow access to reporting in general
- Do not use the inbuilt ADMIN user account to test the mobile interface. This account should only be used on desktop interfaces
- There are some limitations with mobile clients, principally because of design limitations of the devices. For example, Apple iOS devices do not allow access to their file system to be able to upload documents. See here, for a full list of limitations
- Although there are some limitations, there are also opportunities to introduce new features into your workflow with mobile devices. For example, you can directly access the camera and photo albums within mobile devices. You can also use a canvas to capture a user's signature on the screen and upload this directly into an image display type field. The opportunity to configure these items present new features that cannot be met from a desktop computer.

Configuring Limited Access

A typical use case for limiting user access within the mobile client is to allow users to simply report issues, as opposed to having access to all of ExtraView's functionality. The following configuration options allow for up to five screens within the mobile client to be configured, with optional access to the sign on screen, to reports and to other functions:

Behavior Settings

MOBILE_DIRECT_ACCESS_1 to MOBILE_DIRECT_ACCESS_5	These are 5 separate behavior settings that allow for the optional placement of five separate buttons on the Sign On screen of the mobile client. If not configured, no button appears. If configured, the entry should be of the form:
	Button Title:URL

	The *Button Title* will appear on the Sign On screen and the *URL* must be a valid URL within ExtraView to which the user is directed when the button is pressed. It is assumed that a User ID and password are parameters within the URL. Typically this URL will also have an AREA, a PROJECT and ROLE defined and be directed to an *add* screen within ExtraView.
HIDE_MOBILE_SIGNON_SCREEN	This setting may have the value of YES, NO or SIGNON_BUTTON. When the value is NO, the User ID and Password fields are always visible. When the setting is YES, the input fields for the User ID and Password are never viewable and the client is reliant on one or more of the behavior settings named MOBILE_DIRECT_ACCESS_1 through MOBILE_DIRECT_ACCESS_5 providing direct access to screens within ExtraView. When the value is SIGNON_BUTTON, there is a button with the title **Sign On** and when this is clicked, the User ID and Password fields become visible. The default value for this setting is NO, so that the user has the ability to sign on.

Security Permission Keys

MENU_MOBILE_NAV_BAR	This allows the navigation bar that is configured to only appear when there is read and write access to the permission key. The most frequent use of this setting is to suppress the navigation menus for users within the mobile client, locking them into the functionality provided by the behavior settings named MOBILE_DIRECT_ACCESS_1 to MOBILE_DIRECT_ACCESS_5
PR_ADD_PROBLEM.CONFIRM_ALLOW_EDIT	This security permission key will remove the **Edit** button from the confirmation screen displayed to the user following the successful addition of an issue through an *add* screen.

Section 508 - Disability Support

In 1998, the US Congress amended the Rehabilitation Act of 1973 to require US Federal agencies to make their electronic and information technology (EIT) accessible to people with disabilities. The law (29 U.S.C. § 794 (d)) applies to all US Federal agencies when they develop, procure, maintain, or use electronic and information technology. Under Section 508, agencies must give disabled employees and members of the public access to information that is comparable to access available to others. The appropriate code is 36 CFAR Part 1194.22.

ExtraView provides extensive support to build Section 508 compliant applications. ExtraView's inbuilt functionality automatically provides this support, but it is the responsibility of the administrator to ensure that they configure their application specific functions to support Section 508.

As guidance, this page states the current regulations in 36 CFAR Part 1194.22 and provides guidance on how ExtraView applications support Section 508.

1194.22   Web-based intranet and internet information and applications

(a) A text equivalent for every non-text element shall be provided (e.g., via "alt", "longdesc", or in element content).

Every button, clickable image, link, help tip and other navigation elements have a help tip that can be interpreted by screen reader and other equivalent disability supporting software. ExtraView relies upon the manufacturers of supported internet browsers to support screen reading software. ExtraView may be customized extensively with the addition of User Defined Fields (UDFs). For each and every field that an administrator creates, they should create the appropriate text equivalent within the data dictionary, using the Help Text attribute.

(b) Equivalent alternatives for any multimedia presentation shall be synchronized with the presentation.

ExtraView is not designed to support multimedia presentations, hence this requirement is not applicable.

(c) Web pages shall be designed so that all information conveyed with color is also available without color, for example from context or markup.

All colors are configurable by the administrator, and it is thus their responsibility to set appropriate colors and styles that are unambiguous. For example, administrators may configure required field entries to contain markup characters as well as color. A single behavior setting can ensure that all required fields may be displayed as follows:

- Without Section 508 compliance - `Required Field Name`
- With Section 508 compliance - `*Required Field Name*`

The specific markup used is completely configurable by the administrator.

(d) Documents shall be organized so they are readable without requiring an associated style sheet.

Documents stored within ExtraView can be configured so that they do not require any style sheets. For example, administrators should configure standard Text Area fields to store documents, as opposed to configuring HTML Area fields which may contain style sheet information.

(e) Redundant text links shall be provided for each active region of a server-side image map.

There are no server-side generated image maps within the ExtraView application, hence this requirement is not applicable.

(f) Client-side image maps shall be provided instead of server-side image maps except where the regions cannot be defined with an available geometric shape.

There are no server-side generated image maps within the ExtraView application, hence this requirement is not applicable.

(g) Row and column headers shall be identified for data tables.

The ExtraView application adheres to this requirement when generating all data tables, such as those found in reports. Note that this requirement is adhered to when outputting data to the user's internet browser, but the user may also output data to spreadsheets and Adobe PDF documents that do not adhere to this requirement.

(h) Markup shall be used to associate data cells and header cells for data tables that have two or more logical levels of row or column headers.

There is no built-in feature that requires data tables with more than one logical level, although it is possible for the administrator to configure such elements. ExtraView Corporation's recommendation is that the administrator of a Section 508 site should not configure tables with more than a single logical level.

(i) Frames shall be titled with text that facilitates frame identification and navigation.

ExtraView uses one or at most two frames on every screen. These may be configured with identification text. When there are two frames on a screen, the first frame is always the navigation and is always placed at the top of the screen, and the second frame contains data, reducing any confusion for a disabled user.

(j) Pages shall be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz.

This is never an issue with the ExtraView application as it fundamentally presents alpha-numeric information and static images.

(k) A text-only page, with equivalent information or functionality, shall be provided to make a web site comply with the provisions of this part, when compliance cannot be accomplished in any other way. The content of the text-only page shall be updated whenever the primary page changes.

While it is expected that the administrator configures screens that adhere to Section 508, the administrator can also configure text-only pages whenever needed, and link these to their configuration. For example, there is an optional help button on each screen which can access text content specific to the installation, and each UDF title may become a clickable link to a text-only page.

(l) When pages utilize scripting languages to display content, or to create interface elements, the information provided by the script shall be identified with functional text that can be read by assistive technology.

Given ExtraView's presentation of HTML pages that are all generated on a server and displayed in a user's internet browser, all elements are generated with the appropriate attributes that are required by assistive screen reader technology.

(m) When a web page requires that an applet, plug-in or other application be present on the client system to interpret page content, the page must provide a link to a plug-in or applet that complies with §1194.21(a) through (l).

This is not applicable within the ExtraView product.

(n) When electronic forms are designed to be completed on-line, the form shall allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues.

All forms, whether inbuilt into ExtraView, or configured by an administrator are specifically generated by the server to meet this requirement. All fields have titles and input elements that support Section 508 requirements providing the administrator adheres to best practices when configuring the screens.

(o) A method shall be provided that permits users to skip repetitive navigation links.

The only place where repetitive navigation links are used are within forms that input and update records. By default these are at the top and bottom of the form. However, a single behavior setting allows the administrator to turn off the bottom navigation link, thereby creating a compliant form.

(p) When a timed response is required, the user shall be alerted and given sufficient time to indicate more time is required.

The only place where a timed response is required is an optional administrator defined timeout for inactive users. The timer for this input is controlled by the administrator who can ensure timeouts never occur for any user.

Voluntary Product Accessibility Template (VPAT)

In January 2009, the Information Technology Industry Council (ITI) asked for the development of a specification that could be used for ITI's Voluntary Product Accessibility Template (VPAT). This VPAT has been embraced by the U.S. General Services Administration as a way to simplify government market research on IT products with accessible features.

The VPAT for ExtraView may be seen here.

Behavior Settings

Behavior settings control basic functionality within ExtraView and allow your installation to be tailored quickly for your hardware, and your company's requirements with the platform.

The behavior settings are grouped into different categories that can be selected within the **Behavior Settings** administration function.

Behavior Setting Categories

- API
- Company Information
- Display
- Email
- Environment
- LDAP and SSO
- Reporting / Querying
- Security and Session
- User
- Workflow

Commonly used behavior settings

With more than 200 behavior settings, it can be somewhat complex to understand the purpose and behavior of each setting. The settings that are most frequently used are gathered in this section.

Category	Setting	Purpose
User	USER_SELF_REGISTRATION	You can allow your users to self-register, or you can be the registrar. When this is set to Yes, a prompt appears on the Sign On screen, allowing a user

to navigate to a page where they can register themselves as a user in ExtraView. When a user registers in this way, they will only be given the privileges of the user role defined in the behavior setting named LIMITED_USER_ROLE, but no user role is set in their account. When a user self-registers, an email is sent to an administrator, as defined by the following settings with Email Settings on the Email Notification administration menu. The user created will be in the business area and project defined by those of the ADMIN user account.

User	USERNAME_DISPLAY	You have the ability to display usernames by First, Last or by ID based on your company's methodology. First – will produce names like David Smith Last – will produce names like Smith, David ID – will produce names like dsmith
Display	ABBREVIATED_HISTORY	A value of YES will show changed fields only in history records and will not use the History layout to display the audit trail. A value of NO will use the History layout to display the audit trail. The displayed results with YES are more concise than NO, but there is not a fixed layout to easily spot the changes
Display	SUPPORT_LINK	If you have an HTML page that you would like your users to go when they need support, you can put the link plus a message here
Workflow	ENFORCE_STATE_CHANGE_RULES	Gives you the opportunity to turn Status Change Rules on or off
Email	EMAIL_ADMINISTRATOR_NAME	This is the email address or alias for the ExtraView administrator. Emails that are automatically generated by ExtraView are originated with this name. Examples are emails sent upon the self registration of a user, or an unauthorized access attempt
Email	EMAIL_ADMINISTRATOR_USER_ID	This is the email address to which emails originating within ExtraView are sent. This is usually the administrator's email address or an alias for the administrator
Email	EMAIL_FROM_USER_ID	Emails sent from ExtraView will show this as the sender's address. For example, use support@myco.com
Email	EMAIL_SUBJECT_TEMPLATE	This allows you to use fields within the record and place them within the subject line of the email that is generated. The field that you want to include is enclosed between $$ and $$. For example, the issue title is $$SHORT_DESCR$$. The value of: ExtraView Notification [$$ID$$]: $$STATUS$$ - $$SHORT_DESCR$$ will produce an email subject line like: ExtraView Notification [12345]: Open – Report of a failure
Company Information	COMPANY_NAME	The name of your company

Security Permissions

Depending on which version of ExtraView layouts are installed with your new installation, different permission settings may exist. These different layouts may be a minimal set of layouts, where you will do most of the customization yourself, or they may be one or more of a series of layouts for "best practices" implementations such as bug tracking or customer support.

The visibility of individual fields or items within your installation may not be quite what you need. Firstly, if you are not certain, note the ADMIN user supplied with all ExtraView installations overrides the security permission settings, therefore if an item is visible

using the ADMIN user, it is certain that a permission is not set the way you require. Secondly if you cannot see a field you require or you create a field and do not see it where you expect, check the field has been placed on the appropriate layouts as well as checking that the field has the correct permissions. Other sections of this guide explain all these items, but this is a short list of the permissions that frequently need to be checked and altered on a new installation:

Security permission key	Purpose
AREA.nn	nn is the internal ID of the business areas, and if you do not see a business area you expect, check these keys
CF_AREA	Visibility of business areas. This includes the ability to add or edit the list of business areas in administration
CF_PROJECT	Visibility of projects This includes the ability to add or edit the list of projects in administration
CF_PERSONAL_OPTIONS	Gives the ability to edit your personal options
STATUS.xxxxxx	xxxxxx is name of each status value in your installation. If you do not see a status value in a list on the add or edit screens, it is likely you did not give the correct permission to its security permission key

Internationalization / Localization

ExtraView has been designed to work in a global setting. The following attributes should be understood when looking at ExtraView working over multiple countries and languages:

- **Locales** - Locales identify specific languages and geographic regions. Within ExtraView you create an entry for each of the locales you require. A locale is defined as a combination of the Language and the Region. For example, here are some definitions:

Language	Language Code	Region	Region Code
English	en	Great Britain	GB
English	en	United States	US
Japanese	ja	Japan	JP
French	fr	Fance	FR
French	fr	Canada	CA

  For each locale you create, ExtraView will maintain a set of information for the messages to display, the numeric formats to use when displaying data, date formats and sort orders. The two-letter language codes are defined within an ISO standard (ISO 3166-1). The two-letter region codes extend the language code to indicate the geographic region. One language may have many regions. For example, there are aproximately 20 Spanish speaking countries or regions.

  Each user belongs to a single locale, set within their Account Administration screen. For example, you may have users in Japan, Great Britain and Germany. Each user will see date formats in their own locale, with no resulting confusion because of the differing and conflicting date formats in use over the world. If Extraview has been localized for a specific locale, then a user who sets that locale will see the localized messages for that locale. In this way different users in different countries can all see and use ExtraView within their own language. Note that the language translations must be provided for each region. If there is no language translation for any message or title, ExtraView will display the value in the DEFAULT_LANGUAGE, usually English

- **Language Translation** - Each installation has a default language, defined in the behavior setting DEFAULT_LANGUAGE. Usually, this is *en* for English. For each locale that is defined, the administrator can define an alternative to each and every system message and to each and every metadata list value entered when configuring the system. If a user is working within a locale, and no translation has been created for a specific message, ExtraView will display the message using the DEFAULT_LANGUAGE.
- **Date Formats** - For each locale, ExtraView will display date formats with the appropriate local format. For example, if the date is January 10, 2015 and you have selected a SHORT date format, the date will be displayed as 01/10/15 if the locale is *en-US* and displayed as 10/01/15 if the locale is *en-GB*
- **Numeric Formats** - Numbers will be displayed according to the conventions of the locale set for the user. For example, the number 123,456.789 in the en-US locale will be displayed as 123.456,789 in the *de-DE* German locale

- **Sort Order** - Given the different alphabets for different locales, ExtraView will sort lists and results according to the rules of the locale
- **Time zones** - There is support for every time zone around the world. Each user can select the time zone within which he is working. ExtraView will account for time zones in all its calculations. For example, if you have two users in different time zones, and one updates an issue in the Pacific Time zone, and the other views the issue in the European Central time zone, the user in the European Central time zone will see the issue expressed in their time zone. There is support for two types of date fields; fields with a display type of *date* will always be corrected as just mentioned. There are also *day* fields where no correction is made for time. This can be used for events such as a birth date where you never want to correct for time zone
- **Automatic translation of data** - It is possible to configure different text area fields where a field in one language within one field may automatically be translated to a different language in a different field. This uses Google's web service API to perform the translation

As far as possible, ExtraView relies on foundation technology such as the database and the Java language to provide many of these features. For further information on this topic, please consider visiting these reference sources:

- [Internationalization: Understanding Locale in the Java Platform](#)
- [Language localisation - Wikipedia](#)
- [Globalization support in Oracle Database](#)

ExtraView Licensing Schemes

ExtraView is a licensed product. Several types of licensing schemes are available, according to your purchase of the license from ExtraView Corporation. The license details of the ExtraView installation can be viewed in the section named **Company Information Settings** in the **System Controls** section of **Administration**. There are two versions of ExtraView - named **ExtraView Standard** and **ExtraView Enterprise**. These are the key difference in features of each version. Please see the [website](#) for complete details.

ExtraView Standard	ExtraView Professional	ExtraView Enterprise
Zero license cost for unlimited number of licenses	See [here for pricing](#)	See [here for pricing](#)
Support is not provided beyond the first 30 days		
For direct support, see [here for pricing](#)		See [here for support pricing](#)
Supports upto 3 user roles	Supports upto 5 user roles	Supports an unlimited number of user roles
Supports upto 1 Business Area	Supports upto 3 Business Areas	Supports an unlimited number of Business Areas
Supports 25 additional user-defined fields	Supports 150 additional user-defined fields	Supports an unlimited number of user-defined fields
		WOA RESTful API and SOAP Web Services API
		User Custom programming interface
		Extended reporting
		Keyword searching of documents stored in the database
		Compliance with 21CFR Part 11 regulations
		Automatic translation of text between different languages
Privacy of issues based upon the user's company name	Privacy of issues based upon the user's company name	Extended privacy with sophisticated logic

Named User Licensing Scheme

Named users are identified as single-person entities that each has a unique User ID within ExtraView. You may create more end users than the number of licenses you have purchased, but the maximum number of active users cannot exceed the number of licensed users. You may disable any account or accounts to remain in compliance with the license. Note that users who only occupy the role identified in the behavior setting named LIMITED_USER_ROLE are not counted in the licensing scheme, and they will always have access to ExtraView.

Concurrent User Licensing Scheme

You may create any number of user ID's with the concurrent licensing scheme, but not all users may be able to sign on and work within ExtraView at one time. This is because the software will restrict the number of users who can sign on at any time to the number of concurrent licenses purchased. If users in excess of this number attempt to sign on, they will receive a message asking them to contact the ExtraView administrator. You can control the length of time a license remains active, if a user becomes idle, through a behavior setting. After the expiry time, if a user remains idle, their session is expired allowing another user to occupy the license.

If a user with a concurrent license signs on from a single computer, with different browsers, they only occupy a single concurrent license, to a maximum of five connections. Thereafter, a second concurrent license is occupied. If the same user ID is used from different computers, a concurrent license is counted for each computer.

With EVMail usage, whereby an ExtraView issue is either inserted or updated from an incoming email, the sender's address of the incoming email is determined. If the sender's ExtraView account has a concurrent license, or the sender is not a user within ExtraView, a concurrent license is consumed when the email is processed and an issue is inserted or updated. The expiry time of this license is fixed at 30 minutes. If there are insufficient concurrent licenses available at the time EVMail attempts to insert or update an issue, an email is sent to the system administrator and an email is sent to the sender of the incoming email advising them that their email could not be processed.

Mixed Licensing Scheme

When a mixed license is purchased, there will be a maximum number for named users and a maximum number for concurrent users. When any new user is created, they are automatically given the ability to use a concurrent license. The administrator may use the end user account maintenance screen to move any user (up to the number of named licenses purchased) from a concurrent license to become a named user license. Note that users who only occupy the role identified in the behavior setting named LIMITED_USER_ROLE are not counted in the licensing scheme, and they will always have access to ExtraView. These users should not be given a named user license, but should occupy a concurrent license. This is the default behavior.

End-User Licenses without Cost

A single user role within ExtraView, identified by the behavior setting named LIMITED_USER_ROLE, may be used to identify users who are licensed at no cost. There are several limitations with users who belong to this user role:

* They may not update any issues
* They may not become the owner of an issue
* No issue can be assigned to a person with this role
* The user may only belong to single role and may not switch roles.

The ADMIN user account is supplied without any cost. This user account should not be used to insert or update issues as it has special properties.

Operational Task Menu

The Operational Task menu contains the majority of administrative functions that are used day-to-day, to maintain key data such as user information, and to manage logs and sessions. More frequently changing administrative data, such as that used to define business calendars, report schedules and escalation rules are also found here.

User Account Maintenance

This section of the Administration Guide discusses users, how they are defined and how they are maintained. It also explains how user roles are formed and used. Security is a strong point within ExtraView and there is an explanation of how this may be configured in many ways to enhance your ExtraView installation.

Another key concept surrounding users is privacy and how issues may use a feature known as Privacy Groups.

ADMIN User Account Properties

This section deals with all aspects of user administration. Within this section, you will learn how users are created and maintained.

The ADMIN User Account

This account exists in all ExtraView installations and is provided at no cost. This user has special properties and should not, under any circumstances, be used to create and update issues. This is because it bypasses virtually all of the security permissions and workflow controls within the system. Because of this, you should take extra care with the password for this user. It is provided primarily for three reasons:

- This is the user account you use when ExtraView is first installed, allowing you to enter and create your own user ID's.
- If you ever get into a situation where you accidentally lock yourself out of ExtraView, for example by turning off all permissions, then this account will still be active.
- On occasions ExtraView Corporation support personnel may direct you to use this account to help identify a problem.
- If you are utilizing the behavior setting named USER_SELF_REGISTRATION, then users who self register will be given the default business area and project specified in the behavior settings named SELF_REG_DEFAULT_AREA and SELF_REG_DEFAULT_PROJECT.

Accessing User Accounts Without the Needing to Know a User's Password

The ADMIN user has one other property. It allows the administrator to sign in as a different user, without knowing that user's password. This can be very useful for problem solving, or to help a user set up a feature such as their Home Page or a report. To achieve this, the ADMIN user signs on with the following convention:

ADMIN|OTHER_USER_ID

where they substitute the other user's ID for OTHER_USER_ID. They must, of course, enter the ADMIN user password.

Adding Users

Once the required user roles have been created, an administrator may add new users to any of the roles that are active within the system and to which they themselves have permission. Note that users may be assigned privileges for more than one role at a time. From the Administration menu, under the **Users** tab, click the **User Account Maintenance** link. A screen similar to the following screens appears:

User Account Maintenance

Return   Email users   Print Page

There are currently 10 activated named licensed users.
This installation is licensed for 20 users.

Add   Add a new user

Select users by activation status or role   [* Active users * ▾]   Select users by privacy group   [* In any or no privacy groups *
Bank of America
Bear Inc.
Chrysler
Cisco]

The letters in the list indicate the first character of the filter column selected. First, select the column to use as the filter. Secondly, click on a letter to produce a list of just these items, or enter a search pattern, including a wildcard (*). You may also use the Export button to produce a CSV file of the results currently displayed on the screen

①   Select filter column for report   [User ID ▾]

②   Click on a letter   A | B | C | D | E | G | I | J | M | Q | S | T | <all>

     or enter a search expression and press the **Enter** key or click the **Go** button   [Enter search expression]   Go   Export

	User ID	User name	Email Address	Company	Enabled	License type	Last access	Created	Last updated
Edit	ADMIN	System Administrator	admin@xxx.com	Superior Software Corp	Yes	Named	1/23/07 13:57	ExtraView 10/1/03 0:00	System Administrator 10/19/06 16:39
Edit	BSMITH	Bill Smith	bsmith@x.com	Superior Software Corp	Yes	Named	1/31/07 15:02	ExtraView 10/1/03 0:00	Bill Smith 1/30/07 14:25
Edit	CSR	Chris Robinson	csr@xxx.com	Superior Software Corp	Yes	Named	8/2/06 3:59	ExtraView 10/1/03 0:00	Bill Smith 7/14/06 11:18
Edit	DEV	Jimmy Duncan	jimmy@xxxx.com	Superior Software Corp	Yes	Named	1/7/07 0:02	ExtraView 10/1/03 0:00	Bill Smith 1/14/07 12:06
Edit	EVP4D	SCM Daemon		Superior Software Corp	Yes	Named	8/10/05 11:36	ExtraView 10/1/03 0:00	Bill Smith 11/22/05 23:19
Edit	GREG	Greg Goldberg	greg@extraview.com	Superior Software Corp	Yes	Named	8/10/05 11:36	ExtraView 10/1/03 0:00	System Administrator 8/9/05 10:54
Edit	GUEST	Guest User	guest@acme.com	Guest Company	Yes	Named	8/10/05 11:36	ExtraView 10/1/03 0:00	Bill Smith 7/19/06 16:57
Edit	IT	George Miller	gmillar@xxx.com	Superior Software Corp	Yes	Named	8/10/05 11:35	ExtraView 10/1/03 0:00	George Miller 8/10/05 11:35
Edit	JOHN	John Customer	john@xxx.com	ABC Software	Yes	Named	8/10/05 11:34	ExtraView 10/1/03 0:00	System Administrator 8/9/05 15:30
Edit	MARY	Mary Dickens	mary@ssc.com	Superior Software Corp	Yes	Named	8/10/05 11:31	ExtraView 10/1/03 0:00	Mary Dickens 8/10/05 11:33
Edit	QA	Mary Brown	qa@xxx.com	Superior Software Corp	Yes	Named	6/30/06 14:44	ExtraView 10/1/03 0:00	Mary Brown 6/30/06 14:44
Edit	SHUNT	Sally Hunt	shunt@extraview.com	Superior Software Corp	Yes	Named	6/4/06 18:00	ExtraView 10/1/03 0:00	System Administrator 6/4/06 18:02
Edit	TEST	Susan Green	sgreen@xxx.com	Superior Software Corp	Yes	Named	6/20/06 3:09	ExtraView 10/1/03 0:00	Susan Green 6/20/06 3:09

13 record(s) selected from a total of 13 record(s)   Return   Email users   Print Page

User Accounts screen

Note the search capabilities of the screen. Not only can you choose to search by active and inactive users, but you can choose to search by any of the columns of information in the list, such as by **User name**, or **Company**. Use the filter column for the report specified in the select list, then either click on a letter that represents the first character of each entry in the list, or enter an expression in the text input box with the entry **Enter search expression**. You can then press the **Go** button to execute the search and retrieve the records that match the filter and search expression.

Note that if there are a large numbers of users who would be returned to the screen with the filter you set, only the first set of users will be displayed, and there is an opportunity to page through the results.

If you press the **Export** button after executing the search, then all records that are displayed on the screen will be exported to a CSV (Comma Separated Value) file. You can use this file outside of ExtraView for any purpose. The export file will contain any role and attribute information in the form of delimited lists within a single cell in the spreadsheet.

Click the **Add** button to add a new user. The following screen appears:

Add New User screen

The tabs across the top of the screen are for the following purposes:

Tab	Purpose
Personal Information	The key demographic details of the user. All the essential fields to set up and maintain a user are on this tab
Personal Options	The user's personal options that they will also be able to control by themselves
Report Options	The reporting options for the user, including their Home Page reports
Notification Options	The user's email notification options
Privacy Groups	This tab only appears if the behavior setting named ENABLE_PRIVACY_GROUPS is set to YES. The tab is where you control each user's access to privacy groups

Personal Information

This is where you assign the user a unique User ID (which they will use when logging in to the system), their password and other information about the account. You must enter data into all of the required fields, plus any optional fields that you wish to complete.

Basic Information screen

User Information

User Information Fields	Purpose
User ID	The name that the new user will use to login to the system (required). User names the following letters, numbers and characters: A – Z 0 – 9 - _ . @ The user ID must be unique.
Alternative User ID	This field is turned off with the security permission keys named USER.LOGIN_ID and USER.LOGIN_ID_ON_ADD. When you enter the User ID for a new user, the alternative User ID is populated with the same value. This may be changed. Alternative User ID's are available to allow the changing of a user's sign on ID. User ID's are attached to many items of data within ExtraView and are used as foreign key constraints in the database. It would be complex to change all of these if you required to change the User ID for a purpose such as a name change for a user. The Alternative User ID allows the administrator to change the sign on credentials for a user, yet maintain underlying data integrity within the database
First Name	First name. On the **Change a user's details** screen, you can control access to this field with the security permission key named USER.USER_FIRST_NAME
Last Name	User's last name (always required). Note that if you attempt to add a user where the combination of first name and last name already exists, then you will receive a warning, but you will be able to go ahead with the update if you decided this is correct. . On the **Change a user's details** screen, you can control access to this field with the security permission key named USER.USER_LAST_NAME. If the administrator does not have write permission to this key, they are prohibited from creating new users. The field is always visible as ExtraView is unusable if a user's last name is not visible. Thus you cannot turn off read permission for the field.
Password	The user's password. You can control access to this field with the security permission key named USER.USER_SECURITY_PASSWORD. The only occasion that you are likely to make this field not visible is

when you are using a LDAP server for user authentication. There is also a security permission key named USER.USER_PASSWORD_MANAGEMENT. This controls the ability to expire user's passwords, and the time interval information associated with passwords, such as the password expiration interval and the number of days until the password expiry. Write access is needed to alter these fields. . If the administrator does not have write permission to USER.USER_SECURITY_PASSWORD, they are prohibited from creating new users.

Verify Password	Retype the User's password. The same security permission key as for the Password field is applicable.
Primary Email Address	The email address to which automatic email notification will be sent. This is required if the behavior setting named EMAIL_NOTIFICATION is set to YES. Note that if you attempt to add a user where the email address already exists, then you will receive a warning, but you will be able to go ahead with the update if you decided this is correct. The security permission key named USER.USER_EMAIL_ADDRESS controls access to this field, but only when editing a user. The user's email address is always writable when you are creating a new user. When you are updating a user, this field may be given read or read/write permission, or no permissions, in which case it will be invisible.
Job Title	The user's job title. Permission to access this field is controlled with the security permission key named USER.USER_JOB_TITLE.
Company Name	For internal users, this should reflect the same value as COMPANY_NAME in installation setup, especially if you are using the Privacy Group feature of the product as this match is used to determine whether a user may or may not see specific issues. As this field is instrumental in setting up privacy groups accuracy in setting the entry here is important. If you have several or many users in different companies that all access your ExtraView installation, you should consider using the COMPANY_NAME_LIST_UDF setting to provide a list of company names, to complement the manual entry of each company name and potentially spelling these differently for different users. The control for this is a behavior setting named COMPANY_NAME_LIST_UDF. If this is empty, then the Company Name for each user is entered and updated manually. When a new user is created, the field is given a default value of the value of the COMPANY_NAME behavior setting. If you place the name of a User Defined Field (UDF) with a display type of List into the COMPANY_NAME_LIST_UDF behavior setting, then the Company Name becomes a choice of entering a new value or selecting from the list of these values, eliminating the possibility of misspellings and simplifying the population of the field. The UDF field is a normal UDF in all respects, and you may share its use on add and edit screens as well as use it on reports. Note that the internal company name configured in the COMPANY_NAME behavior setting, does not appear in this list and is never added to the list. When a user self registers to access ExtraView, this field is set to a value of their email address, ensuring uniqueness and that the user cannot become a member of a Privacy group, thereby seeing issues which he is not authorized to view. When updating the user's account, this field can be controlled with the USER.COMPANY_NAME field.
Address	This is two-line field that is always present on the user account screen
City	The city of the user. This is always present
State / Province	The state or province of the user's address. This is always present
Zip / Postal Code	The zip or postal code of the user's address. This is always present
Country	The country of the user's address. This is always present
Work phone	The user's work phone. This is always present.
Home phone	The user's home phone. This is always present.
Cell phone	The user's cell phone number. This is always present.
Fax	The user's fax number. This is always present.
Pager	The user's pager number. This is always present.
User field 1 - User field 10	This series of ten fields are turned off by default, using the security permission keys named USER.USER_DEFINED_1 through USER.USER_DEFINED_10. You may enable the security permission keys and alter the titles of these fields with their entries in the inbuilt field section of the data dictionary, thereby using these fields for any purpose.

Security Information

Security Information Fields	Purpose

Enabled User	This checkbox controls whether the user is enabled or disabled. User's accounts are never deleted from ExtraView; instead they are disabled and the user may no longer access their account. In addition, the user's name will not appear selectable in lists in the normal fashion. If enabled with the behavior setting named ALLOW_SEARCH_DEACTIVATED_USERS, users may still use deactivated users as a filter to search for records. The reason that user's accounts are not deleted but rather deactivated is because you may still wish to view users' names who originated issues or otherwise have their name set as a value in a field within an issue.
Named User	This field appears if you have a mixed mode license to ExtraView with both named and concurrent users. If this box is checked, then the user will occupy a named license, else they will occupy a concurrent license
Expiry Interval	This is a number that defines how many days should elapse between the user being prompted to change his password.If the value is zero, the behavior setting named PASSWORD_EXPIRE_TIME_DAYS is used to determine the interval.
Expire Password Now	If the administrator checks this option, then the user's password is expired immediately. On the next occasion that the user attempts to sign on, they are forced to alter their password. Note that this option does not appear if an LDAP server is controlling user authentication.
Days to password expiry	The label to this field also shows the number of days left, before the password expires. You can set this field to any number of days to choose a new expiry date. If you set this number to -1, the current password may be used once, and the user must provide a new password on the next occasion they sign on. If the behavior setting named PASSWORD_EXPIRE_TIME_DAYS is set to zero, then user's passwords are not expired automatically.
Select User Roles	Functional teams where members have the same set of privileges. Simply click on one or more roles that you want the user to be able to adopt. If the user is not granted any role, then they will assume the role nominated by the behavior setting named DEFAULT_USER_ROLE. If they are granted more than one role, then the user will see a select list in the navigation bar, and they can assume any of their valid roles by making a selection from this list

Default Area & Project

Default Area / Project Fields	Purpose
Set Default Area	This is where you set the default Business Area for the user. This is the Business Area they will point to when they sign on
Set Default Project	This is where you set the default Project for the user. This is the Project they will point to when they sign on

Photograph

If you wish to store users' photographs, they can be uploaded. Valid image file types are GIF, JPG and PNG. Images are resized to 80 pixels wide by 100 pixels in height. Once a user's photograph is uploaded and saved, you may use it within workflow, for example to include a photograph of the user on the *add* or *edit* screen of an issue. This is accomplished using rules, which take the user's image and copy it into a field within the issue. If you have a field name USER_PHOTO on a sreen, you can populate it within rules as follows:

```
<== load ==>
USER_PHOTO = USER.{image};
```

Signature

If you wish to store users' signatures, they can be uploaded. Valid image file types are GIF, JPG and PNG. Signatures are resized to 750 pixels wide by 300 pixels in height. Once a user's signature is uploaded and saved, you may use it within workflow, for example to print the user's signature on outgoing correspondence. Assume the correspondence to be printed is held within an HTML Area field on the *edit* screen of your issues. Placing the signature within the correspondence is a two-step process. First, create and hide a field on the *edit* screen which will receive the signature. This field is populated with rules, such as:

```
<== load ==>
USER_SIGNATURE = USER.{signature};
```

Within the HTML Area field, include a token for the signature field. In this case, the token would be $$USER_SIGNATURE$$.

When the HTML Area field is used as the content for an email (probably using a rule that sends the HTML Area field as an email, the token is replaced with the image of the user's signature.

Signatures are typically scanned into your computer, and then uploaded into ExtraView. It's important to upload a signature of a usable size. There is no point scanning an entire sheet of paper with the signature somewhere in the middle, as the white space surrounding the signature will be included in the uploaded image. Scanning software often includes tools that crop images, and most desktop operating systems such as Microsoft Windows include tools that crop images. There are also many third-party utilities that perform the function. To assist you in scanning a signature of the correct size, use the template that you may download here, to aid in scanning an image to the correct size, before cropping. Print the template and ask users to sign within the box before the signature is scanned.

Personal Options

The personal options screen allows the administrator or user to set their basic parameters for their use of ExtraView.

Personal Options screen

Personal Option Fields	Purpose
Text size	Choose Small, Medium or Large from the list, according to the size of the font to be used for the user's display.
Language	Choose from the list of available languages for the installation. The user's user interface to ExtraView will then use all localized messages available for the installation.
Time Zone	Time zone selection applies for the user system-wide
Date Format	Date format selection applies for the new user. All places where the date is visible to the user will use this format. Alternatively a custom date mask can be set, and a date format designed to the specification required
Custom Date Format	The date mask allows the setting of a date format, should the inbuilt formats not be sufficient for your needs. See Appendix A for a detailed explanation of custom date formats.
Time in 24 Hour Format	Should be Yes or No
Chart / PDF output font	The font saved in the user's profile must be a valid chart font for the locale in which the user is

configured. If it is not valid, then the behavior setting for DEFAULT_CHART_FONT is inspected; if the default chart font is not valid, then the first valid font from the list of available fonts on the application server is used. If ExtraView cannot find the font or a font without the correct character set is defined, it is likely that both charts and reports sent to PDF output will not display correctly. This is most typically an issue with Asian users who require double-byte character sets.

Start page	The user may select a start page. This is the page to which a user is directed, when they first sign on to ExtraView. By default, this is the Home Page, but it may be selected from the list of available start pages in this list. This option may be turned off completely, with the behavior setting named USER_DEFINED_START_PAGE. The administrator may add additional start pages with the advanced administration feature named **Manage the list of start pages available to users**. The user must also have permission to use a start page, before they are allowed to view the page. The initial values available, when this feature is turned on are the Home Page, the Search / Report page, the Add Issue screen and the Administration screen
Default Workspace	This provides a choice between all the saved workspaces for a user, as to which one will be opened when the user first signs on to a workspace. This only appears if the user has permission to access workspaces
Drilldown from Reports	This provides an option for the user when they are using workspaces. The user may either choose to drillwown from queries and reports to a single Quicklist panel within the workspace, or to open a different workspace for each drilldown operation. This option only appears if the user has permission to access workspaces
Browser Character Set	This only appears to the user account named ADMIN. Choose the character set for the user's display. Typically this is set to **UTF-8 Unicode 8-bit Transfer**.
MS Office Char. Set	This only appears to the user account named ADMIN. Choose the character set for exporting information from ExtraView to Microsoft Office products. The default is **UTF-16LE Unicode 16-bit LittleEndian**. This setting works with most language versions (and certainly all English language versions) of Microsoft Office products. If the user has difficult reading some characters on reports exported to Microsoft Excel or other Office products, they should find out which character set their Microsoft product is set to use, and make the setting in this list the same.
File Attachment Char. Set	This only appears to the user account named ADMIN. This is the default character set for attachments that the user will upload when they are managing issues. The initial value of this default is provided in the behavior setting named DEFAULT_ATTACHMENT_CHARSET.
Email character set	This only appears to the user account named ADMIN. This is the default character set that is used for sending email to the user. The default for this is UTF-8. The default can be changed with the behavior setting named EMAIL_CHARSET
Workspace Settings	If you have been granted permission to utilize workspaces, then you may set some options for your personal preferences.

- **Default Workspace** - you can set the default workspace to be loaded when you first enter the workspace feature
- **Size of icons on panels** -Choose small or large icons in the panel menubars
- **Drilldown for Reports** - you can select whether your personal preference is to minimize the number of Quicklist panels opened when you drill down from reports. You may prefer to keep this number to a minimum, or to create a new panel with each new drilldown
- **Auto-scroll to Panels** - you may choose what action that ExtraView takes when you click on a panel within a workspace. The default is that the panel obtains the focus and you work within that. However, you can choose to have ExtraView auto-scroll as much of the panel as it can, into view, from the top left-hand corner of the workspace that's visible. The first choice is better if you do a lot of dragging and dropping of issues between panels, as the panels will remain in their place within your browser. The second choice is better if you work mainly in *add* or *edit* screens, and want to see as much of their contents as possible, when you click on a panel. In both cases, clicking on a panel's icon within the navigation bar, will auto-scroll the viewing area of the workspace to make as much of the panel visible as is possible
- **Scroll Panels** - setting this to **Panels Do Not Need Focus** allows you to scroll the workspace canvas or any workspace panel or element within that panel when the mouse pointer is over the respective area. Panels do not get focus when this happens. The **Panels Need Focus** option signifies that you can only use your mouse to scroll the panel and its elements, if the panel has focus. You need to sign off and on again for this to take effect
- **Vertical Offset for New Panels** - If this is set to **Yes** then new panels are opened to the right and down from the position where the previous panel was opened. If you set this to **No** then new panels are only opened to the right of the previous panel, and at the same vertical offset.

Report Options

Report Option Fields	Purpose
Drilldown Report Format	Quicklist or Detailed Report
Home Page report #1	Choose the report that you want to appear first on the Home Page for the user. This option only appears when editing the user account.
Home Page report #2	Choose the report that you want to appear second on the Home Page for the user. This option only appears when editing the user account.
Home Page report #3	Choose the report that you want to appear third on the Home Page for the user. This option only appears when editing the user account.

Note that the ability to select and set the reports on a user's Home Page is controlled by a security permission key named SR_SET_HOME_PAGE_REPORTS. If the user does not have read access to this permission key, then they may not change their Home Page reports. However, this ability is usually retained by the administrator so it is possible for the administrator to select and freeze the reports a user sees on their Home Page.

Report Options screen

Notification Options

These options control how various notifications are made to the user.

Notification Option Fields	Purpose
Receive notifications at your primary email address	If this is set to Yes, the user will receive email at his primary email address. The primary email address is set on the **Personal Details** tab
Receive notifications at alternative email address	If this is set to Yes, the user will receive email at his alternative email address. The alternative email address is set on the next option. The user role must have read/write permissions to the security permission key named USER.USER_EMAIL_ADDRESS in order to control this address
Alternative email address	This is the alternative email address, controlled by the previous prompt. The user role must have read/write permissions to the security permission key named USER.USER_EMAIL_ADDRESS in order to control this address
Receive notification on	Enabled will receive automatic email Disabled will not receive automatic email

own updates

Receive attachments on own updates	This allows a user account to be configured such that attachments to email notifications will be suppressed. The user will need to sign into their account to view attachments
Email format	HTML, Plain Text (brief), or Plain Text (full). The brief option provides just a few fields, enough to recognize the issue
Interest lists	Allows the opt-in or opt-out to interest lists. See the section in this guide on interest lists for full information. The key reason for interest list management on this screen is to allow users to opt-in and opt-out of interest lists to which they have permission
Escalation rules	Allows the opt-in or opt-out to escalation rules. See the section in this guide on escalation rules for full information. The key reason for escalation rule management on this screen is to allow users to opt-in and opt-out of escalation rules to which they have permission

Security permission keys separately control access to the interest list on the notification option tab for end-users. The CF_INTEREST_LIST permission key is used for this purpose. Without read/write access to this key, the end user cannot see or alter the interest list.

Notification Options screen

Privacy Groups

Personal Option Fields	Purpose
Privacy Groups	Privacy Groups enable certain groups within ExtraView to view different specific sets of issues without making the issue PUBLIC (all users can see the issue) or PRIVATE (only internal users can see the issue)

Privacy Group screen

When you have completed the entries for the user on each of the above screens, click the **Add User** button.

Note: If users are assigned to more than one user role, they should be informed they should click the **User Role** link to toggle amongst available group designations.

Note: If a new user is created and not added to any User Role, ExtraView will automatically assign them to the role defined in the behavior setting named LIMITED_USER_ROLE. This is typically the customer or guest user role.

Note: When you create a new user, the only Home Page Reports that are offered within the list boxes are Public Reports.

Note: If you attempt to create a new user and either the combination of the first and last name already exists, or the email address is already in use with another user, you will receive a warning and you are asked to confirm that you want to create the new user account.

User Attributes

User Attributes are a configurable feature which must be supported by custom code. Without custom code, this feature may display fields on the User Administration screens, but no actions are taken.

How fields that are set up as user attributes are used is totally dependent upon the custom code. A typical use case may be to control which issues a user may read and update, by using field values within the issue to identify the user's eligibility to view and update the issue. Issues which do not have the appropriate values that correspond with the user's attributes will be totally invisible to the user, both for updating and reporting.

Setting up user attributes requires the following steps:

* Provide permission to the user role(s) that will be allowed to administer the user attributes. This is accomplished with the permission key named SE_USER_ATTRIBUTES
* Use the above role to modify the fields in the data dictionary which will be used to control the user attributes. It is not recommended that you set up a mechanism with too many fields, as it will become cumbersome for the administrator. Typically two to four fields are used. Edit the fields, and set a value of *Yes* in the field **Used for user attributes**. Optionally set

the **Sort sequence for user attributes** field to sort the fields on the user account administration screen. Note that you may only configure List, Popup List, Tab and Radio Button display types to use this feature

- Go to the user account administration screen and edit a user. Note there is a new tab on the display, with the default title of **ATTRIBUTES**. When you click on this tab, you will see a screen similar to this:

When complete, the administrator uses the fields on the screen to select the values that are used within the custom code to set the desired functionality

- Develop the custom code to support the fields, utilizing the following user custom exits. These are documented in the User Custom Guide.
  - ucUAGetAttributesDescription
  - ucUAGenerateHTML
  - ucUAValidateForm
  - ucUATransactUser
  - ucUAPostUpdate
  - ucUAGetTabs

Updating User Accounts

On the User Account Maintenance screen use the filters to select the user account you wish to modify.

- Click the **Edit** button associated with the user account, and a screen similar to the following appears:

Change a User's Details screen
A full explanation of all the fields is available under the **Adding Users** section of this guide.

- Edit user information on any of the tabs as needed, and then click the **Update User** button.
- Note that when editing a user's account information, you can select which reports to add to their Home Page, by choosing up to three reports from the lists towards the bottom of the screen. You can select from public reports, or the user's personal reports.

Note: To deactivate a user account, click the **Personal Information** tab and use the checkbox named **Enabled User** to deactivate or reactivate an account. Once an account is deactivated, the user will no longer be able to login to the system, and their User ID will no longer appear in Assigned To or Owner pick list options, etc. To reactivate a user account, select *All inactive users* from the **Select Users by activation status** or role pick list on the Edit User Accounts screen. Once you have drilled down on the desired inactive account, use the **Enabled User** checkbox that appears on the **User Roles/Security** tab of the Change a User's Details screen. When you deactivate a user, they are removed from all interest lists, escalation rules as well as being removed from being the module owner of any module field value. If you reactivate a user, you will need to review all their settings for accuracy as ExtraView may reset some values such as the license type, when the account is restored.

When you manually deactivate a user account, a check is made to see whether the account is the owner of scheduled reports that are delivered via email to other users, or are sent to the document repository. If the user account owns such reports, you are offered the ability to select a different user who will become the owner of these schedules. You should select a user who has the appropriate security permissions to own and run the report schedules. If you do not want to pass the schedule for the reports to another user, use the administration utility **Manage Scheduled Reports** to remove the account owner from the schedule, or use the administration utility **Manage Reports** to delete the reports entirely.

Note: The entry **Alternative User ID** allows a user's ID to be altered. This does not change the initial ID used to create the account, but provides an alternative that can be used by the user to sign on. This is most useful for the occasions when a user's name changes, such as when a female changes their name upon marriage. When the alternative User ID differs from the User ID, either may be used to sign on.

Note: If the behavior setting named ENFORCE_DETAILED_USER_INFO is set to "Yes", then additional fields will become required. This is used when you want users that self-register on the system to provide a significant level of personal details.

Note: If you alter a user's access to specific user roles, and remove their access to a user role, this will only take effect the next time they sign on. If they are signed on at the time you change their access, they will continue to have the access you are inhibiting.

Note: Users are not physically deleted from ExtraView. The reason for this is that once most users have used ExtraView, they will be recorded as the Owner, Assigned To, or other such person against issues. To preserve data integrity, ExtraView requires the user details for everyone, even though they may not continue to use and access the system.

Note: If you alter an interest list, an escalation rule entry or change the user's language setting, there is an immediate update to the account. Pressing **Cancel** does not reverse this change.

Unauthorized User Access Attempts

If a user attempts to sign on repeatedly, without success, ExtraView will disable their account as a security precaution. The controls for this feature are within the administrative Site Configuration menu, on the **Security and Session Settings** menu.

Behavior Setting	Default Value	Description
MAX_SIGNON_ATTEMPTS	3	The maximum number of consecutive failed sign on attempts allowed by an individual user before their account is disabled. The number of failures is measured in the period defined in SIGNON_PERIOD_MINUTES
SIGNON_PERIOD_MINUTES	5	Once a user fails to sign on because of an invalid password, they will be allowed to make a total of MAX_SIGNON_ATTEMPTS in this period

If a user's account is disabled, the system administrator is notified by email. The system administrator is also notified if a person from a specific IP address attempts to sign on repeatedly and unsuccessfully from the same address according to the above settings. If the user is using the User ID of ADMIN and attempting to sign on, then the MAX_SIGNON_ATTEMPTS is internally set to two times the value in the behavior setting. This is because it can be a significant maintenance problem if the ADMIN account is disabled and cannot be re-enabled. Thus ExtraView allows some extra sign on attempts before disabling the ADMIN account, but security considerations still ensure that there is not the ability to try to sign on repeatedly.

If the behavior setting named CUSTOM_AUTHENTICATION is set to YES, then the user's account is not disabled if there are repeated failed sign on attempts. The assumption is that the custom authentication method, such as the LDAP server, will perform the action of disabling the user's account and it is superfluous to disable the account in two places.

The settings for notification are in the Email Settings menu within **Email Notification** and are:

Behavior Setting	Description
EMAIL_ADMINISTRATOR_NAME	This is the email address or alias for the ExtraView administrator. Emails that are automatically generated by ExtraView are originated with this name. Examples are emails sent upon the self registration of a user, or an unauthorized access attempt
EMAIL_ADMINISTRATOR_USER_ID	This is the email address to which emails originating within ExtraView are sent. This is usually the administrator's email address or an alias for the administrator

Hierarchies of Admin User Roles

In larger organizations implementing multi-tenanted ExtraView installations, it may be useful to have multiple administrators, each of whom only have access to specific administrative functions, or only have access to modify the characteristics of specific business areas. For example, you may want to create a sub-administrative role that can only add new users and modify these user's settings and options.

To create a hierarchy of user roles take the following steps:

- With a user who has the *Administrator* user role, enter **Administration** ==> **Site Configuration** ==> **User Roles**. Within the **User Roles** utility you can create the new user role or roles you require. You can also copy the initial permissions for the role(s) from the existing *Administrator* role.
- Within the **Administration** ==> **Site Configuration** menu, use the **Grant Security Privileges** function to modify the administrative permissions for the new role or role you created. Use the general category named * **All administration options keys** * to see all the keys that control access to the administrative functions
- Remember that you can restrict or grant access to the user roles for any administrator role

Company Name Security

There is a behavior setting named COMPANY_NAME. ExtraView recognizes this as the main company name. Given that this is the case, a new user added with a company name that is different from the value in the field associated with COMPANY_NAME will not be able to see PRIVATE issues submitted by users in your company. This is especially beneficial if you have customers using the system, and you only want them to have limited privileges to view their own issues. If you have enabled users or customers to self-register their selves within ExtraView, they will able to sign on to ExtraView, and they will automatically be assigned to the user role entered as the LIMITED_USER_ROLE. The LIMITED_USER_ROLE is set in the Workflow Settings administration menu. This group normally has minimal user privileges. By being added to this role, the user will only be able to view PUBLIC records until a System Administrator re-assigns them to one or more User Roles.

Note: When you use the company name as security, and a user moves from one company to another, and you update their record with the new company name, then all the users at that new company will have visibility of the issues that were entered by that person at their original company. You can avoid this by giving the new user a new User ID at the new company, or being diligent to make sure that all their original issues are updated to no longer refer to that person.

Note: If you are signed on with the user ID of admin, the company name security is bypassed. The admin user will see all records.

Note: There is a behavior setting named ENABLE_PRIVACY_GRP_OVERRIDE. If the value is set to YES, then internal users can see all issues regardless of the value of the PRIVACY field. Internal users are defined by the user's personal Company Name being identical to the company name defined by the behavior setting named COMPANY_NAME. If the value is set to NO, users may only see issues when they are a member of the privacy group to which the issue is assigned.

The Company Name and the Privacy of Issues

There is a behavior setting on the administration menu, System Controls tab under Company Information Settings that controls the interaction between the privacy of issues and the different users within different companies that enter these issues. This is entitled ENABLE_COMPANY_NAME_ACCESS. When this setting is YES, different users who have the same company name will be able to view all issues entered by any member of that company. This overrides the Privacy setting of PRIVATE. When the value of ENABLE_COMPANY_NAME_ACCESS is NO, the Privacy setting overrides the behavior, and all issues will be held as being strictly PRIVATE as described in the section on Privacy. The most typical use of this setting is to allow you to give your customers access to ExtraView, and to allow any member of an individual customer to see all the issues entered by his / her colleagues.

Note: A user will always be able to view an issue that they originated, irrespective of their company name.

There is a behavior setting named COMPANY_OVERRIDE_FIELDS that contains a comma separated list of field names with a display type of User. These fields must exist in the data dictionary. These fields contain the users that the company name security feature works with. By default, company name security works with only the ORIGINATOR field. This setting extends the feature to these other fields. The use case for this is as follows: The COMPANY_NAME is set to *Host Company*. They are the host of the ExtraView site. An issue is submitted by a user in Company A. The *Company A* user can see the issue as they are the Originator of the issue. The issue is visible to all *Host Company* users, because the default company name security allows them to see all issues. A *Host Company* user is set as the Owner of the issue. They want to assign the issue to a user who is neither a *Host Company* user, nor are they a *Company A* user. They want, for example, to assign the issue to a user who is a member of a company named *Company B*. The goal is to extend company name security in such a way that the *Company B* user is able to view the issue, originally created by the *Company A* user, because they are assigned to the issue. The *Company B* user will not see any other *Company A* originated issues, nor will they be able to see any other *Host Company* issues, because they are not the ASSIGNED_TO person on other issues.

The Company Name and Privacy Groups

Company name security with ENABLE_COMPANY_NAME_ACCESS and privacy group security are independent features of ExtraView, yet they combine together. The following example summarizes the behavior and interaction of company name security and privacy groups.

Privacy Group Example
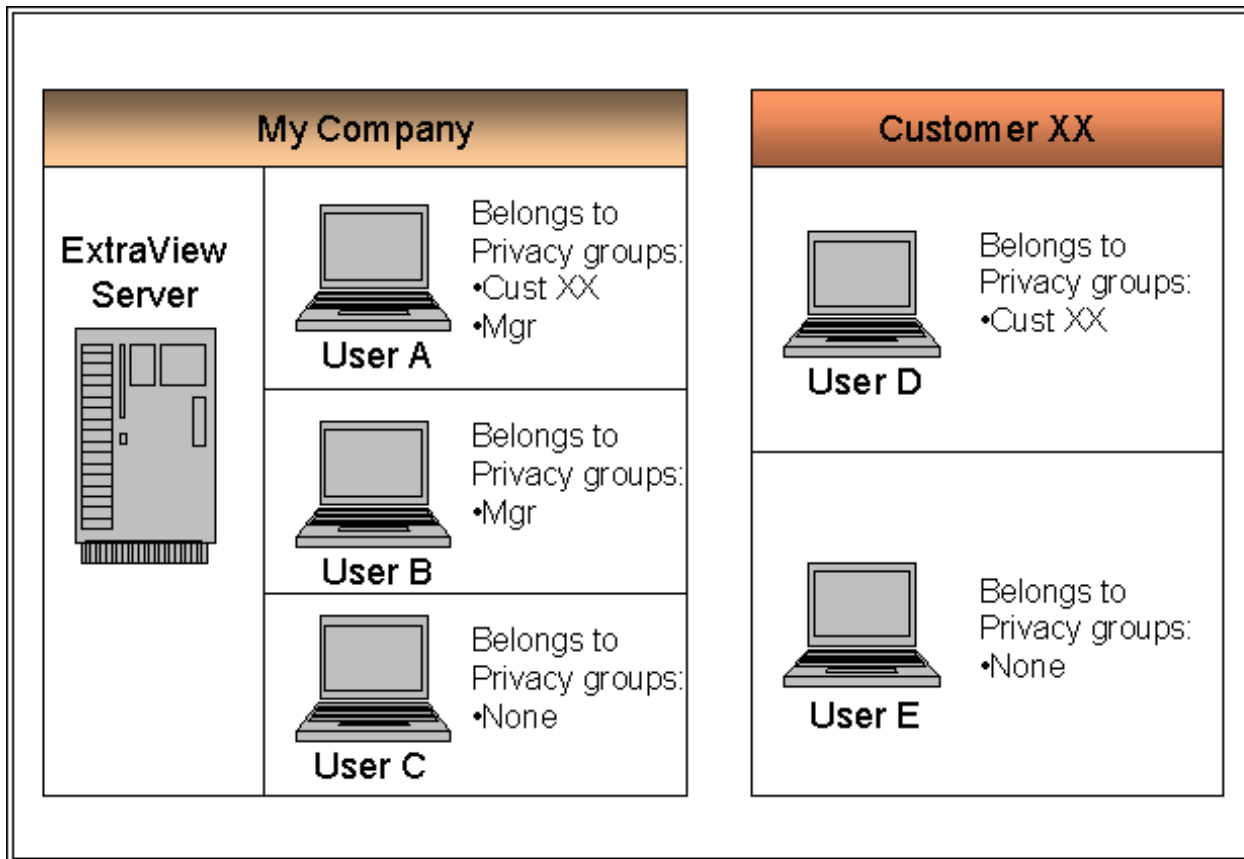
Privacy of issues

- The COMPANY_NAME behavior setting is *My Company*
- ENABLE_COMPANY_NAME_ACCESS is set to YES
- ENABLE_PRIVACY_GROUPS is enabled
- Privacy groups have been created, named *Cust XX* and *Mgr*

User A

- Employee of *My Company* (Company Name on their user screen is set to *My Company*)
- Member of privacy group *Cust XX*
- Member of privacy group *Mgr*

User B

- User B is an employee of *My Company*
- Member of privacy group *Mgr*

User C

- User C is an employee of *My Company*
- Not a member of privacy group *Mgr*

User D

- User D is an employee of *Customer XX*, and has been granted access to *My Company's* ExtraView installation
- Member of the *Cust XX* privacy group

User E

- User E is an employee of *Customer XX*, and has been granted access to *My Company's* ExtraView installation
- They are not a member of any privacy group

Who sees which issues? This table shows issues created in different privacy groups and indicates which users will be able to see the issues. The 'Y' indicates that the user will be able to see the issue. Remember that security permission settings and a default setting can be used to control who can see and who can update the privacy field.

	My Company			*Customer XX*	
Privacy Group User	User A	User B	User C	User D	User E
PUBLIC	Y	Y	Y	Y	Y
PRIVATE	Y	Y	Y	-	-
Cust XX	Y	-	-	Y	-
Mgr	Y	Y	-	-	-

To complete the explanation, note that if it is User D or User E that originates the issue, and the issue remains PRIVATE, then all users in the above scenario will be able to access the issue (but not the users from any other companies). This is because ENABLE_COMPANY_NAME_ACCESS is set to YES.

Updating your Personal Account Details

This utility allows any user to update their account details to control various display options, such as how you want to enter dates into the system, what your password will be, user information that others will see, and notification via e-mail,

To update your account details

1. Click the link with your user name:
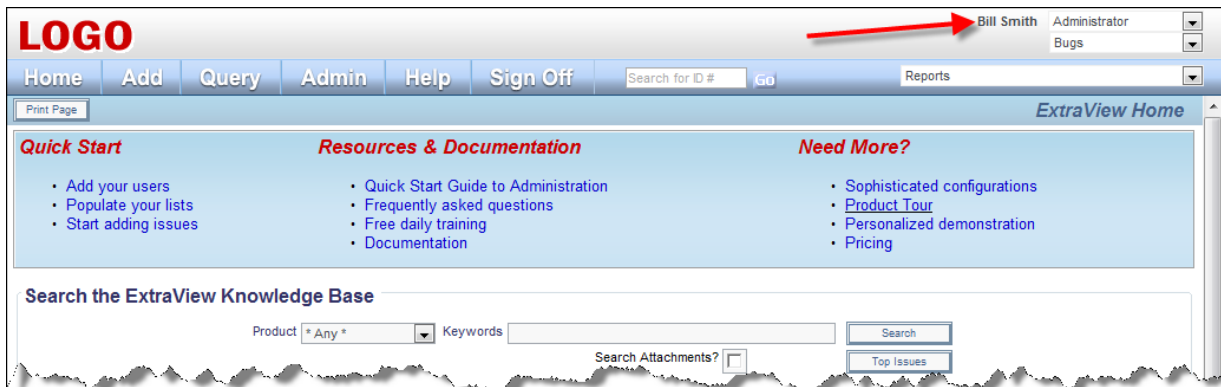
Accounts menu on Home Page
The following screen appears:

Personal Options Password screen

2. Once you have verified your password, enter any new information on the Change a User's Details screen. You can move to other tabs in the utility and update fields there
3. When done, click the **Update User** button.

Notes:

Change Personal Details screen

Customizing User Account Details

ExtraView offers the ability to customize user accounts through a number of pre-defined fields which can be used to add new fields to each user's record. Outside of key fields which are mandatory within each user's record, each of the following fields has a security permission key which can be used to hide or enable the field.

Each of the pre-defined keys is protected with a security permission key, so you can set up different access profiles to the different user fields for different user roles.

The fields that can be customized are shown below:

Field	Purpose
USER.ACCESS_UDF_DATA	This permission key controls overall access to the complete list of user defined fields USER.USER_DEFINED_1 - USER.USER_DEFINED_10
USER.ADDRESS	The two fields that comprise the user's address
USER.CELL_PHONE	The cell phone number of the user
USER.CHART_FONT	The font used to create charts
USER.COMPANY_NAME	The user's company name
USER.CUSTOM_DATE_MASK	The custom date mask used in preference to the USER.DATE_FORMAT
USER.DATE_FORMAT	The user's date format

USER.EMAIL_FORMAT	The key used to set the format of the email notifications that a user receives
USER.FAX	The user's fax number
USER_HOURS_12_24	The key that allows a user to choose the time format for input and display
USER.LANGUAGE	The user's language
USER.LOGIN_ID	An alternative to the User ID that can be used to sign on to ExtraView. Once a user account has been established, the USER_ID cannot be altered as it is used in many places, such as recording who creates and who updates issues. In some organizations, there might be a need to provide an alternative, for example after a company takeover. Also, users may change their name, so this facility allows the user to set up an alternative to their original USER ID
USER.LOGIN_ID_ON_ADD	This key controls whether the USER.LOGIN_ID is shown on the *Add User* screen. This is not typically needed, but is available if necessary
USER.NOTIFICATION_ALTERNATIVE	The key that allows a user to choose their alternative email address
USER.NOTIFICATION_PRIMARY	The key that allows a user to choose their primary email address
USER.NOTIFY_ATTACHMENTS	Allows the user role to set the notification option to exclude attachments in email notifications
USER.IMAGE	Allows the user role to upload their photograph
USER.PAGER	The pager number of the user
USER.SIGNATURE	Allows the user role to upload their signature
USER.START_PAGE	This permission controls the ability of the user to set their start page when entering ExtraView, to a commonly used page such as the Home Page, the Add Issue screen or the Search / Report screen. This is set on their personal preferences page
USER.TEXT_SIZE	Controls the text sizes that a user may select
USER.TIME_ZONE	The key to allow the changing of the user's time zone
USER.USER_DEFINED_1 - USER.USER_DEFINED_10	These are ten user defined fields for the user record. You can set the security permission to any one of these to read and write, and alter the title of the field in the data dictionary as needed. This provides the capability to have ten user defined text fields for your installation
USER.USER_EMAIL_ADDRESS	This is used to control read/write access to the email address of the user. This only has an affect on the update screen for the user. There is always write access to the field when creating a new user
USER.USER_ENABLE	Write access is needed to this field in order to be able to enable and disable user accounts
USER.USER_FIRST_NAME	When updating the details of a user, this permission key controls read and write access to the field. If there is no read or write access when updating a user, the field is not visible. The field is always writable when creating a new user
USER.USER_INTEREST_LIST	Giving read and write access to this key allows each end user to opt-in and opt-out of interest lists to which they may belong
USER.USER_JOB_TITLE	The user's Job title
USER.USER_LAST_NAME	The user's last name
USER.USER_NAMED	This permission controls the ability for an account administrator to set a user to use an available NAMED user license when the system is in MIXED mode. The default for newly created users is that they use a CONCURRENT license. This is not used if you have only NAMED or only CONCURRENT users in your installation
USER.USER_PASSWORD_MANAGEMENT	This controls the ability to expire users' passwords, and the time interval information associated with passwords, such as the password expiration interval

and the number of days until the password expiry. Write access is needed to alter these fields

| USER.USER_SECURITY_PASSWORD | The user's password |
| USER.WORK_PHONE | The work phone number of the user |

User Sign On Log

The user sign on log allows the administrator to determine all sign on, sign off and unsuccessful accesses by users. When you access the sign on log, you are offered a choice of filters. The sign on log is accessed within **Administration**, **Operational Tasks**.

If you choose the defaults, you will display a listing that shows all entries for all users for the last month. You can filter by the log entry type (sign on, sign off, unsuccessful attempts to sign on) or by a single user, or by any date range.

Choosing the filters for the sign on log screen

A sample of the sign on log is shown below. You can sort the list by any of the columns, Log entry #, Date, User ID, Entry type, Log entry, IP address, Active Users or Unlicensed Active Users. Sort the columns, by clicking on the header. An arrow shows the column currently used for the sort. Note that if you click on the column that is currently selected, the report is resorted, but in descending rather than ascending order.

This Active Users column shows how many users were connected at the moment that the sign on took place, allowing you to monitor the number of users connected to ExtraView over time. This is an aid to capacity planning.

Note that the User ID field may appear differently, according to the behavior setting of USERNAME_DISPLAY. If this is set to **POPUP**, then you may either enter the name in the User ID field, according to the format in the behavior setting named USERNAME_DISPLAY, or you may press the button beside the field, giving you a search capability to look for the user's name or user's ID that you want to select.

Log Entry #	Date	User Name	Type	Log Entry	IP Address	Active Users	Unlicensed Active Users	Reason Code
622969	3/3/08	HOJO	System Signon Success		202.232.197.33	5	0	
622974	3/3/08	RAJESHY	System Signon Success		196.12.38.195	6	0	
622979	3/3/08	CHINNAMK	System Signoff Success		196.12.38.195	6	0	
622984	3/3/08	STEVE.GADOURY	System Signon Success		209.202.70.226	6	1	
622989	3/3/08	JONATHAN.CLUGSTON	System Signon Success		157.234.254.24	7	2	
622994	3/3/08	JONATHAN.CLUGSTON	System Signoff Success		157.234.254.24	7	2	
622999	3/3/08	ALAN.ATKINS	System Signon Success		207.59.153.82	7	2	
623004	3/3/08	DCRANE	System Signon Success		65.209.222.142	8	2	
623009	3/3/08	ANNE.HOPKINS	System Signon Success		65.209.222.142	9	2	
623014	3/3/08	JKIENITZ	System Signon Success		65.209.222.142	10	2	
623019	3/3/08	GARY.RATHER	System Signon Success		65.209.222.140	10	2	
623024	3/3/08	JKIENITZ	System Signon Success		65.209.222.142	11	2	
623029	3/3/08	WANGSH	System Signon Success		65.209.222.142	12	2	
623034	3/3/08	WHEATLEY	System Signon Success		155.53.42.121	13	3	
623039	3/3/08	CARL.KOPPEL	System Signon Success		65.209.222.142	14	3	
623044	3/3/08	JONATHAN.CLUGSTON	System Signon Success		157.234.254.24	15	4	
623049	3/3/08	BRANDI.JONES	System Signon Success		67.98.28.122	16	5	
623054	3/3/08	BRANDI.JONES	System Signoff Success		67.98.28.122	16	5	
623059	3/3/08	TOM.HOGYE	System Signon Success		65.209.222.142	16	4	
623064	3/3/08	ENEARY	System Signon Success		72.139.90.138	17	4	
623069	3/3/08	ALLAN.ROFER	System Signon Success		65.209.222.142	18	4	
623074	3/3/08	null	System Signoff Success		null	18	4	
623079	3/3/08	MOHAMMED	System Signon Failed to sign on user	158.96.4.1		18	4	Incorrect Password
623084	3/3/08	LEEANN	System Signon Success		65.209.222.142	19	4	
623089	3/3/08	ALAN.ATKINS	System Signoff Success		207.59.153.82	19	4	
623094	3/3/08	RAJ.KANDADA	System Signon Success		64.164.109.2	19	4	
623099	3/3/08	KEITH.KRIEWALL	System Signon Success		150.215.91.219	20	5	
623104	3/3/08	KEITH.KRIEWALL	System Signoff Success		150.215.91.219	20	5	
623109	3/3/08	ALEX.PINELL	System Signon Success		146.184.0.114	20	5	
623114	3/3/08	DBORLEAN	System Signon Success		65.209.222.142	21	5	
623119	3/3/08	SSUBJECT	System Signon Failed to sign on user	63.175.134.130		21	5	Invalid User ID
623124	3/3/08	SSUBJECT	System Signon Failed to sign on user	63.175.134.130		21	5	Invalid User ID
623129	3/3/08	SSUBJECT	System Signon Failed to sign on user	63.175.134.130		21	5	Invalid User ID
623134	3/3/08	ADMIN	System Signon Failed to sign on user	63.175.134.130		21	5	Incorrect Password
623139	3/3/08	KEITH.KRIEWALL	System Signon Success		150.215.91.219	22	6	
623144	3/3/08	KEITH.KRIEWALL	System Signoff Success		150.215.91.219	22	6	
623149	3/3/08	DCRANE	System Signon Success		65.209.222.142	21	5	
623154	3/3/08	LEEANN	System Signon Success		65.209.222.142	21	5	

ExtraView Sign On Log

The definitions of the columns in this report are:

Column Name	Definition	License Model
Log Entry #	Serially generated ID number	All
Date	Date and time of sign on/sign off	All
User Name	User ID entered	All
Type	Sign on or sign off	All
Log Entry	Success or Failure	All
IP Address	Address of sign on client	All
Active Users	Number of currently used concurrent licenses, not including the ADMIN or SYSTEM users if they are signed on. This is calculated as follows: • For each distinct, concurrent non-GUEST user, count 1 license used • For each user that is logged in 6 times or more, count 1 additional license used for each five sign on connections beyond the first sign on	All
License Count	The number of configured licensed users from the NUM_LICENSE_USERS behavior setting	Concurrent/Mixed
Active	Number of signed on users who are consuming license, i.e. non-GUEST users, and not the	Mixed

Concurrent Users	ADMIN or SYSTEM user	
Concurrent License Count	Number of configured concurrent licenses from the NUM_CONCURRENT_USERS behavior setting	Mixed
Unlicensed Active Users	Number of signed on users who are not using a license, i.e. GUEST users	All
Reason Code	One of the following:	All

- License restriction
- Incorrect Password
- Expired Password
- Invalid User ID
- Custom authentication rejection
- User is not enabled
- License restriction

If allowed to grow with no constraints, the sign on log can grow over a period of time to an enormous size, retaining information that may no longer have value. A behavior setting named SYSTEM_LOG_EXPIRE_TIME_DAYS controls how long information is retained in the log. The default for this time is 30 days, but can be changed by the administrator. An internal system task within ExtraView deletes sign on log messages older than this time each hour, to avoid a buildup of the task that would take a considerable amount of time.

System Log

The system log allows the administrator to determine all significant metadata transactions, except for user sign on and sign off. Sign on and sign off activity can be observed in the ExtraView Sign On log. This feature also gives access to the application server log file. When you access the system log, you are offered a choice of filters.

If you choose the defaults, you will display a listing that shows all entries for all users for the last month. You can filter by the log entry type (sign on, sign off, unsuccessful attempts to sign on) or by a single user, or by any date range.

Choosing the filters for accessing the system log

A sample of the system log is shown below. You can sort the list by any of the columns, Log entry #, Date, User ID, Type, Log entry. Sort the columns, by clicking on the header. An arrow shows the column currently used for the sort. Note that if you click on the column that is currently selected, the report is resorted, but in descending rather than ascending order.

System Log   Return   Print Page

Log entry #	Date	User ID	Type	Log entry
174010	February 16, 2006	ADMIN	Update Existing SECURITY_USER	update security_user set ADDRESS_LINE1 = ?, ADDRESS_LINE2 = ?, ALPHA_TIMEZONE = ?, AREA_ID = ?, CELL_PHONE = ?, CHART_FONT = ?, CITY = ?, ('','','America/Los_Angeles','1','','SansSerif','','Toyo Corporation','','JP','SYSTEM','2006-02-16 09:41:39.0','MEDIUMDATE','QUICKLIST','','HTML','Y','','Taro','','','ja',''
174015	February 16, 2006	ADMIN	Delete SECURITY_GROUP_USER	delete from security_group_user where security_user_id = ? ('TARON')
174020	February 16, 2006	ADMIN	Add SECURITY_GROUP_USER	Add to security_group_user, insert into security_group_user(security_user_id, security_group_id) values(?, ?) ('TARON','ADMIN')
174025	February 16, 2006	ADMIN	Add SECURITY_GROUP_USER	Add to security_group_user, insert into security_group_user(security_user_id, security_group_id) values(?, ?) ('TARON','ADMIN') ('TARON','GUEST')
174030	February 16, 2006	ADMIN	Update Existing SECURITY_USER	update security_user set ADDRESS_LINE1 = ?, ADDRESS_LINE2 = ?, ALPHA_TIMEZONE = ?, AREA_ID = ?, CELL_PHONE = ?, CHART_FONT = ?, CITY = ?, ('','','America/Los_Angeles','1','','SansSerif','','Toyo Corporation','','JP','SYSTEM','2006-02-16 09:41:39.0','MEDIUMDATE','QUICKLIST','','HTML','Y','','Taro','','','ja',''
174035	February 16, 2006	ADMIN	Delete PRIVACY_GROUP_USER	delete from privacy_group_user where security_user_id = ? ('TARON')
174040	February 16, 2006	TARON	Update Existing DATA_DICTIONARY	update data_dictionary set allow_add_to_list = ?, child_key = ?, column_name = ?, default_format = ?, default_size = ?, default_value = ?, dependent_sql = ? ('N','','LAST_UPDATED_BY_USER','null','null','','N','null','LABEL','N','Y','The last person who updated the attachment','','N','0','TARON','','','','','','','','',''
174045	February 16, 2006	TARON	Update Existing DATA_DICTIONARY	update data_dictionary set allow_add_to_list = ?, child_key = ?, column_name = ?, default_format = ?, default_size = ?, default_value = ?, dependent_sql = ? ('N','','LAST_DATE_UPDATED','null','null','','N','null','LABEL','N','Y','The last date the attachment got updated','','N','0','TARON','','','','','','','','N','','','','','','N','N',''
174050	February 16, 2006	TARON	Update Existing DATA_DICTIONARY	update data_dictionary set allow_add_to_list = ?, child_key = ?, column_name = ?, default_format = ?, default_size = ?, default_value = ?, dependent_sql = ? ('null','','','null','','null','N','null','LABEL','N','N','Where you edit an existing issue','index.html','N','0','TARON','null','null','null','null','null','null','','N','','','nu
174055	February 16, 2006	TARON	Update Existing DATA_DICTIONARY	update data_dictionary set allow_add_to_list = ?, child_key = ?, column_name = ?, default_format = ?, default_size = ?, default_value = ?, dependent_sql = ? ('N','','','null','null','','null','N','null','LABEL','N','N','Where you are able to run complex queries and reports','index.html','N','0','TARON','null','null','null','null','null',
174060	February 16, 2006	TARON	Update Existing DATA_DICTIONARY	update data_dictionary set allow_add_to_list = ?, child_key = ?, column_name = ?, default_format = ?, default_size = ?, default_value = ?, dependent_sql = ? ('N','','','null','null','N','null','LABEL','N','N','','','N','0','TARON','null','null','null','null','null','null','','N','null','null','','null','N','N','N','','ExtraView Cu
174065	February 16, 2006	TARON	Update Existing DATA_DICTIONARY	update data_dictionary set allow_add_to_list = ?, child_key = ?, column_name = ?, default_format = ?, default_size = ?, default_value = ?, dependent_sql = ? ('N','','','null','null','','null','N','null','LABEL','N','N','The summary of a newly entered issue','index.html','N','0','TARON','null','null','null','null','null','null','','N','null','nu
174070	February 16, 2006	TARON	Update Existing SECURITY_MENU	update SECURITY_MENU set SECURITY_MENU_ID = ?, SECURITY_MENU_TITLE = ?, SECURITY_MENU_GROUP = ?, ADMIN_TOP_LEVEL_MENU_ORDER = ?, t ('3','Fields & Layouts','admin','3','4816','3')
174075	February 16, 2006	TARON	Update Existing SECURITY_MENU	update SECURITY_MENU set SECURITY_MENU_ID = ?, SECURITY_MENU_TITLE = ?, SECURITY_MENU_GROUP = ?, ADMIN_TOP_LEVEL_MENU_ORDER = ?, t ('8','Display','admin','4','4821','8')
174080	February 16, 2006	TARON	Update Existing SECURITY_MENU	update SECURITY_MENU set SECURITY_MENU_ID = ?, SECURITY_MENU_TITLE = ?, SECURITY_MENU_GROUP = ?, ADMIN_TOP_LEVEL_MENU_ORDER = ?, t ('6','Import/Export','admin','8','4819','6')

15 record(s) selected from a total of 15 record(s)   Return   Print Page

ExtraView Sign On Log

The log entries show the actual SQL used to update the ExtraView database. Note that the SQL statement contains a series of question marks. Each question mark in turn is replaced at execution time with a value from the parameter list.

If allowed to grow with no constraints, the system log can grow over a period of time to an enormous size, retaining information that may no longer have value. A behavior setting named SYSTEM_LOG_EXPIRE_TIME_DAYS controls how long information is retained in the log. The default for this time is 30 days, but can be changed by the administrator. An internal system task within ExtraView deletes system log messages older than this time each hour, to avoid a build up of the task that would take a considerable amount of time.

If you would like to view a copy of the application server log file, press the **View Application Server Log** button on the System Log screen. When you do this, a new window with the contents of the log file appears. This file may be very large, and it may take some time to download into this window. It may be more convenient for you to view this file with an editor directly on the file system of the server. This button only appears on the System Log administration screen if you are granted the read permission to the security permission key named CF_VIEW_APP_SERVER_LOG

Session Management

Sessions are managed for each user with a variety of behavior settings. The key settings are found in the **Security and Session Settings** category of the behavior settings.

Sessions do not have a one to one correspondence with the number of users who are signed on. ExtraView creates many sessions for each user, roughly in proportion to the number of tasks or reports a user executes at one time. Drilldowns from reports require additional sessions and when users do not close windows or sign off when they have completed their tasks, the amount of session information held in memory grows as ExtraView will not discard session information that a user may require again at some time in the future.

A table is used to store user sessions in the database when there is no more room in the application server memory (RAM) to store them. If a user session is not active, i.e. the user has closed their browser window without logging off, or if their computer is sitting with nobody using it but the browser is logged in, then the session is saved in the database table to make room for new sessions which are active.

Sessions can only be used by ExtraView when they are in memory, therefore if a session in the database needs to wake up and be used, the session must be recalled into memory. If there is not enough room, another session must be taken out of memory and put in the database, to make room for the first session.

If there are more active sessions needed than there is available memory, sessions will be continuously swapped between the database and memory – this is very time consuming and affects the overall performance. The parameters SPILL_SESSION_COUNT and NOSPILL_SESSION_COUNT control how many sessions are kept in memory. NOSPILL_SESSION_COUNT defines how many

sessions will stay in memory. Once the number of sessions in memory grows beyond NOSPILL_SESSION_COUNT, no more sessions will be added to memory until ExtraView can move some sessions out of memory and to the database.

SPILL_SESSION_COUNT must be a larger number than NOSPILL_SESSION_COUNT. SPILL_SESSION_COUNT indicates the maximum number of sessions that will ever be placed into memory. If the number of sessions in memory reaches SPILL_SESSION_COUNT ExtraView stops creating new sessions until the number of sessions in memory is reduced by saving sessions to the database.

For example set NOSPILL_SESSION_COUNT to 1000 and SPILL_SESSION_COUNT to 1100. With these settings, sessions are created in memory until there is a total of 1,000 sessions. When the 1,001st session is requested, one session in memory must be placed to database before the session can go into memory. This process can take some time and more requests for sessions may come in while this is occurring.

If there are 1,100 requests for sessions in memory received, no more sessions will be created, i.e. the system will not respond, until the writer process catches up and removes sessions from memory to get SPILL_SESSION_COUNT below the 1,100 threshold. At least 101 sessions need to be removed from memory in order to get the number of sessions below 1,000 in order for the system to catch up completely.

How many sessions are in memory at a point in time? This number is written to the application server log with the completion of each service. The number is the cc parameter. If this number is higher than NOSPILL_SESSION_COUNT then sessions are spilling over into the database and it's worth considering whether you should set the NOSPILL_SESSION_COUNT and SPILL_SESSION_COUNT to higher amounts.

Other factors can also influence how many sessions are in memory. Sessions are removed from memory and from the database once a user's session has expired, so keeping the SESSION_EXPIRE_TIME_HOURS to a lower amount will reduce how long older sessions are kept in memory, reducing the number of sessions taking up space.

One way to improve performance through session management is to increase the number of application server instances used. In a clustered application server environment, each instance will hold NOSPILL_SESSION_COUNT number of sessions in memory – so 2 application servers will hold 2,000 sessions, and 3 application servers will hold 3,000 sessions before any sessions are saved to the database. Load-balancing multiple application servers provides more memory to hold more sessions.

Conditions that sign off users automatically

The most common reason that ExtraView signs off users automatically is because they reach the timeout period you set for the license to expire. For users with named licenses, this is the time period specified in the setting SESSION_NAME_EXPIRE_TIME_HOURS. For users with concurrent licenses, the time period is set with SESSION_EXPIRE_TIME_HOURS. If the user is idle (i.e. they do not submit a request or a form to the server from their browser) for more than this period, then ExtraView only allows the user to continue after signing on again.

When ExtraView signs a user off automatically, they are asked to re-enter their user ID and password to continue working. When they were signed off the system by ExtraView, the alert shows a **Session expired or removed** message, by Administrator followed by the code.

Alert Code	Meaning
RC1	The referenced session no longer exists. The most likely reason is that the user was idle longer that the time allotted by the system administrator and the user was automatically signed out of the system. This condition may also be caused by the administrator restarting the server while you were in the middle of a task
RC2	The cookies returned by the browser don't match session cookies held on the server. There is a possibility that the user signed on with the same browser twice
RC3	The session expired in the foreground
RC4	The session was removed by administrator
RC5	Your IP address changed during the session and this is disallowed by the system administrator. Normally ExtraView performs a check for security purposes, to ensure that your IP address remains consistent. However, this check can be disabled if you part of a network where your IP address is automatically translated for any reason. This is most often seen when accessing a corporate network via a VPN or proxy server.

Note that although ExtraView attempts to keep the user signed on when their session has not expired, there are some circumstances beyond its control. If you find that user's session expire at times you cannot account for, then it is likely to be for one of the following reasons:

- If the user is behind a Proxy server or accessing ExtraView via a VPN, sessions can be disconnected early due to timeouts on other parts of the connection that ExtraView has no control over
- If a user signs on to ExtraView using multiple windows or tabs in their browser, the session expiration time generally applies from the session that the user first signed on to
- If a user signs on to ExtraView and during their work they generate new tabs and / or new pop up windows and then close the "parent" window or tab, the child windows and tabs will lose track of the parent session, and therefore lose track of the correct expiry time.

Named End-User Session Management

The setting named SESSION_NAME_EXPIRE_HOURS controls the period of time for the expiry of a named user license. By definition, there is not a strict need from a licensing perspective for end users who occupy a named license to ever have their session terminated automatically. However, every session requires resources on the server and these resources are finite. Apart from the user who forgets to sign off after an extended period, there is also the occasional computer or browser crash that will leave sessions hanging. After a period of time these events take a toll and performance can be reduced to other users. ExtraView's recommendation is that there is a 24 hour setting on named licenses. Provided the application server is configured correctly, this should clean up unused sessions without an undue burden on the server.

Concurrent End-User Session Management

Note: This feature is only activated when you have purchased concurrent or hybrid licenses of ExtraView, as opposed to purchasing named user licenses. It is not necessary to have this capability with named user licenses.

This feature gives the Administrator complete knowledge of who is signed on to the system at the current time. The Administrator with permission to this feature is also permitted to sign off any user. This may be needed if you have reached the limit of the count of licensed users, and you want to sign off some individual users, in order to let other users access the system.

The time that an individual user's session remains open, when there is no further activity is controlled with the behavior setting named SESSION_EXPIRE_TIME_HOURS. This is found within the Security and Session Settings administration menu, on the Systems Controls tab. The administrator can adjust this time to give a reasonable balance between a user's session expiring, the need to maintain available licenses to other users, and security. For efficiency, the session termination time shown on the screen for each user is only updated internally within ExtraView every few minutes, so the time shown is only approximate.

In addition, you can observe the IP address of any connected user. This can be useful for troubleshooting on networks where IP addresses are translated for any purpose. ExtraView has a behavior setting within the Security and Session Settings section that dictates whether a client connection must maintain a constant IP address during a session. This is named CLIENT_IP_ADDRESS_CHECK. Usually this is set to YES, but it may need to be set to NO if your server is accessed via a proxy server, and the IP address of an individual user may change through time.

A further behavior setting controls how ExtraView reacts when the user either closes their last remaining open window or closes their browser, with ExtraView being the last window open. If the behavior setting named AUTO_SIGNOFF_ON_USER_EXIT is set to NO (this is the default), then ExtraView will (by using a session cookie) remember that it is signed on, and an open license is retained. If the setting is YES, then when the user presses the browser back button, or other means, to go back to the ExtraView session, the user will need to sign on again, taking a fresh license.

Note that two timestamps are displayed in the report. One timestamp is shown in the personal date/time format of the current administrative user. The other is shown in an extended, unambiguous format.

Check box to disconnect	User ID	Alternative User ID	User name	User email	Company name	Session Termination Time User Format	Session Termination Time Global Format	IP Address	Keep Alive	License Type	Node ID
☐	CARL.KOPPEL	C	Carl Koppel	carl@extraview.com	ExtraView Corp	Sep 3, 2012	2012-09-03 15:55:01.000 -0700	127.0.0.1	Yes	Concurrent	WS_A:
☐	CARL.KOPPEL	C	Carl Koppel	carl@extraview.com	ExtraView Corp	Sep 3, 2012	2012-09-03 15:57:31.000 -0700	127.0.0.1	Yes	Concurrent	WS_A:
☐	DBORLEAN	DBORLEAN	Daniel Borlean		My Company Name	Sep 3, 2012	2012-09-03 15:58:09.000 -0700	127.0.0.1	Yes	Concurrent	WS_A:
☐	GWU	GWU	Geo WU	geow@the-mcorp.com	ExtraView Corp	Sep 3, 2012	2012-09-03 15:59:02.000 -0700	127.0.0.1	Yes	Concurrent	WS_A:

End-user Session Management

Directions

The following users are currently signed on to ExtraView. Check the box against any user you want to disconnect, to release the license back to the common pool. Once you disconnect users, they will need to sign on again before using ExtraView.

Disconnect Users Return Print Page

Manage Connected Users screen

On the screen is also the column **Keep Alive**. This signifies whether a user's session is still current and whether it may be killed due to exceeding the idle time expiry period. If the value is No, then the session has exceeded the value of USER_EXPIRE_TIME_HOURS, but has yet to be terminated by ExtraView. If the user is idle, but is accessing an add or edit screen, then the value may still be **YES**, as the server will preserve the session information for the user for as long a period as possible, up until the value of time expressed by SESSION_EXPIRE_TIME_HOURS is reached.

Session Expiry

Session expiry is a two-stage process. For users occupying a concurrent license, when the user has been idle, (i.e. they have not accessed the server from their browser for the time defined in USER_EXPIRE_TIME_HOURS), they will receive a session expiry message from ExtraView when they next access the server. If the user was in the middle of a session creating a new issue or editing an existing issue, then ExtraView will attempt to restore all the data held in the user's browser from the time they last entered data. This happens after they re-authenticate their sign on credentials.

This capability comes at a price, in that the server must hold all the data necessary to restore the session within the application server's memory. With a significant number of users, this can mount up, and so there is a second time out defined in SESSION_EXPIRE_TIME_HOURS. If the user attempts to restore their session after this time, no data can be restored, and the user will be taken to their usual destination page upon signing on, usually the Home Page.

For users who occupy a named user license there is a separate behavior setting for control. This is named SESSION_NAME_EXPIRE_TIME_HOURS.

When a user receives an alert informing him that their **Session expired or was removed**, it may be for one of several reasons –

Alert Code	Meaning
RC1	The most likely reason is that the user was idle longer that the time allotted by the administrator in the behavior setting named SESSION_EXPIRE_TIME_HOURS. ExtraView then signs the user out of the system. This condition may also be caused by the administrator restarting the server while you were in the middle of a task
RC2	The cookies returned by the browser don't match session cookies held on the server. There are two possible reasons for this. First, the expiry may happen if the user attempts to sign on to ExtraView, from two different instances of the same browser. Secondly, if the server is first accessed by a specific URL from a user's browser, the same URL must then be used for all subsequent accesses by all users.
RC3	The session expired in the server. This is most often due to an error condition that could not be handled
RC4	The session was removed by administrator. This happens when the administrator uses the Manage Connected Users administrative function to cancel a user's session.
RC5	Your IP address changed during the session and CLIENT_IP_ADDRESS_CHECK is "YES". Normally ExtraView performs a check for security purposes, to ensure that your IP address remains consistent. However, this check can be disabled if you part of a network where your IP address is automatically translated for any reason. This is most often seen when accessing a corporate network via a VPN with a proxy server.

You may turn on a warning to the user, that their session is about to expire, using the two behavior settings named SESSION_WARNING_TIME_SECS and SESSION_WARNING_INTERVAL_SECS. This will give them additional warnings that their session is about to expire.

For users occupying concurrent licenses, the session expiry is controlled by a number of settings, some optional, some mandatory. The two mandatory settings are

Behavior Setting	Purpose
SESSION_EXPIRE_TIME_HOURS	This is the session expiry period measured in hours for a user who occupies a concurrent user license and who remains idle. Note that this is different from the period for users who occupy a named license. Their expiry period is defined in the behavior setting named SESSION_NAME_EXPIRE_TIME_ HOURS. After the period set in USER_EXPIRE_TIME_HOURS and before the value of SESSION_EXPIRE_TIME_HOURS, ExtraView will attempt to restore a user's data when their session has expired. Beyond the time specified in SESSION_EXPIRE_TIME_HOURS

the data will be lost. If the value of this setting is less than one minute or 0, then the session data will be kept permanently until the server is restarted. This is not recommended.

USER_EXPIRE_TIME_HOURS	This is the time in hours for which a user will remain signed in to ExtraView, when his computer is idle. ExtraView recognizes a user as still active when an action causes the user to access the server. When a user's session expires, ExtraView will ask the user to sign in again, and will attempt to restore any data being inserted or updated, and take the user to the point where they were working. Note that the ability to restore data is only enabled for the period specified in the setting named SESSION_EXPIRE_TIME_HOURS. For this reason, SESSION_EXPIRE_TIME_HOURS should always be equal to or greater than USER_EXPIRE_TIME_HOURS. If the value of this setting is less than one minute or 0, then the session will not time out

There are further settings that may be used to determine if a user is really inactive, or whether they have closed their browser, navigated to another page or performed some other action that means that the session information held on the server is no longer required. This is principally used to allow the administrator to fine tune the allocation of concurrent user licenses after the USER_EXPIRE_TIME_HOURS and before the SESSION_EXPIRE_TIME_HOURS is reached, and to ensure that users do not retain a license if there is no need. The basic principal is that if the user requires resource to be retained on the server, they will retain their license during this period, but if they do not require resources on the server, their license will be released. Resources are required on the server if the user is in one of these functions:

- *Add* screen
- *Edit* screen
- Item History screen
- Related Issue Update screen

This functionality comes at a modest performance cost, in that there needs to be communication at regular intervals between the client browser and the server to determine a user's status. These packets of information may have an affect on performance, therefore this feature may be turned on or off, and the administrator may determine how frequently checks are made.

If you are using named user licenses the recommendation is to turn the feature off using the USER_TIMEOUT_SESSION_REMOVAL setting.

Behavior Setting	Purpose
USER_TIMEOUT_SESSION_REMOVAL	This is used to turn this feature on or off. The value of YES implies that sessions are killed after USER_EXPIRE_TIME_HOURS of inactivity, unless the user has sessions that remain active in the *add* or *edit* screen mode. NO implies that these sessions are not killed after this period of inactivity. The user's browser is checked by the server every SESSION_MONITOR_POLL_SECS to see if the ExtraView session is still active and the client updates the server with this information every KEEPALIVE_INTERVAL_SECS seconds.
KEEPALIVE_INTERVAL_SECS	This is the number of seconds between polling operations from the *add* or *edit* screen. This timer is used to send a message from the user's browser to the server, to allow the server to keep track of users with open *add* or *edit* sessions. If the user closes their browser or navigates away from an *add* or *edit* page, then the server will be able to recognize this event
SESSION_MONITOR_POLL_SECS	This is the number of seconds between periodic tests from the server, for session removal due to user expire time activity. The server will test to see if the user still has an open session at this interval. This is only used if USER_TIMEOUT_SESSION_REMOVAL is YES. and if the user's browser has not reported to the server that a session is still active with the KEEPALIVE_INTERVAL_SECS timer.

Disabling User Access

Occasionally, the administrator may wish to lock users out of ExtraView, for maintenance purposes such as creating new screen layouts or business rules or when performing a mass update of many issues. To facilitate this requirement, the feature **Disable and Enable User Access** is available from the **Administration**, **Operational Tasks** menu.

You must be the ADMIN user, or in the role specified by the ADMIN_OVERRIDE_ROLE behavior setting in order to disable end user access to ExtraView.

Disabling and enabling user access

When entering this feature, the administrator will see a list of all users who are currently signed on, giving them and indication of the level of activity. A message can be defined, that is displayed to the user when they sign on. It is suggested the message include an indication of when service is expected to be restored. When the system is disabled, there is a confirmation message to this effect. Until the time that the administrator re-enables access, end users are prevented from activity, and will see the message defined above when they attempt to sign on. Access will continue to be granted to administrators who are members of the group defined in the behavior setting named ADMIN_OVERRIDE_ROLE. A security key controls access to this feature. It is named CF_ENABLE_DISABLE_USER_ACCESS.

Note: It is not recommended that anyone other than administrators who are part of the ADMIN_OVERRIDE_ROLE be given access to this feature.

Disconnecting Users

1. Click **End-User Session Management** link within the **Administration** ==> **Operational Tasks** menu.
2. Click on the checkbox associated with the user(s) whom you want to remove from active sessions within ExtraView.
3. Click the Disconnect Users button. This action will expire the selected user sessions and open up concurrent license seats for access by other users.

Report Manager

The Report Manager utility allows the ExtraView administrator to maintain public reports and user's reports. Both public reports and reports owned by individual users may be accessed, deleted and some report attributes may be updated. Reports may be be moved from one user to another. The update operations act on single reports or different filtered groups of reports. The typical functions of the utility are to perform the following tasks:

- Delete all reports in the system that have not been run in a while, say the last year
- Delete all the reports in the system belonging to an individual user, after they have ceased being an ExtraView user

The utility is extremely flexible in allowing the administrator to filter through very large groups of reports, to quickly decide on which reports the mass operation is to be fulfilled. It is possible to delete many reports with a few keystrokes, so unless you are familiar with the operation of this utility, it is suggested you take a backup of the entire site before beginning. When first entering the utility, the screen will look somwewhat similar to this:
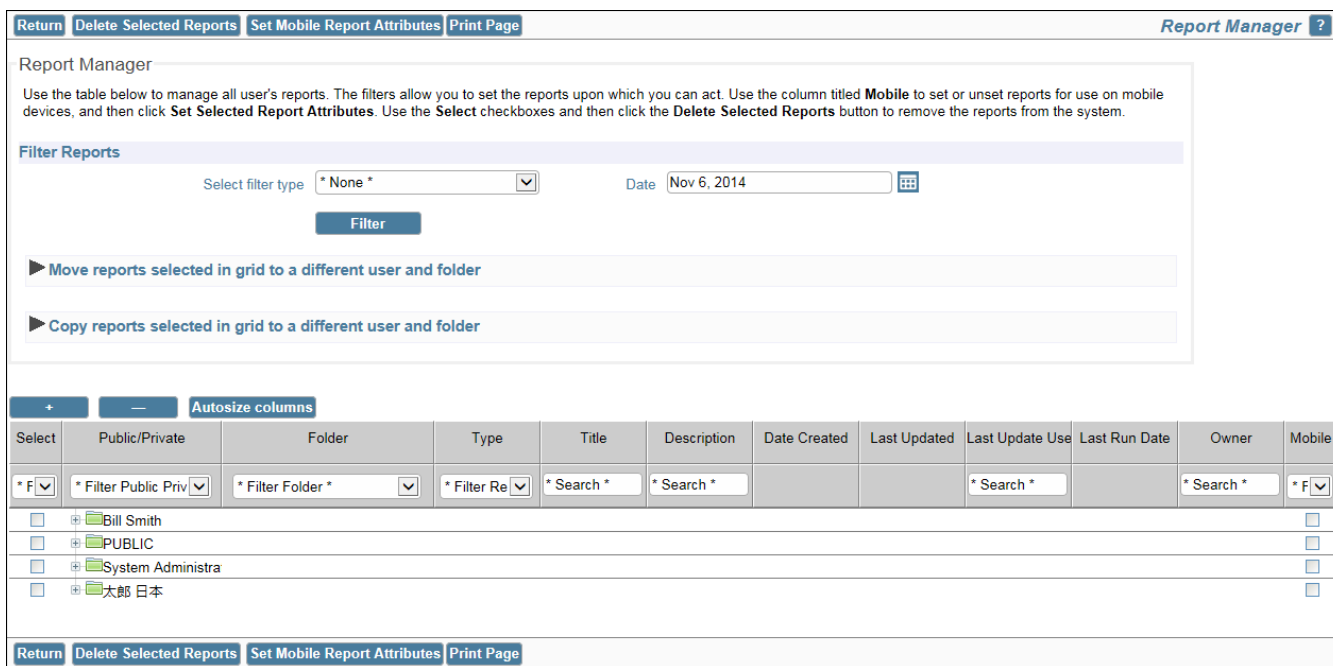
The filters allow you to select the date range to be viewed, based on either the dates the reports were created, the dates they were last updated, or the dates they were last run. If you do not use the filters, you are able to view all the reports in the system.

The overall functionality of the report grid is:

- **Filters** - Based on a date that you select, you can filter the reports that are displayed within the grid with the following filter types:
  - Last run on or before a date
  - Created on or before a date
  - Last updated on or before a date
  - Last run on or after a date
  - Created on or after a date
  - Last updated on or after a date
- **Move Reports** - Click on the fold to expand it to view the following screenshot:
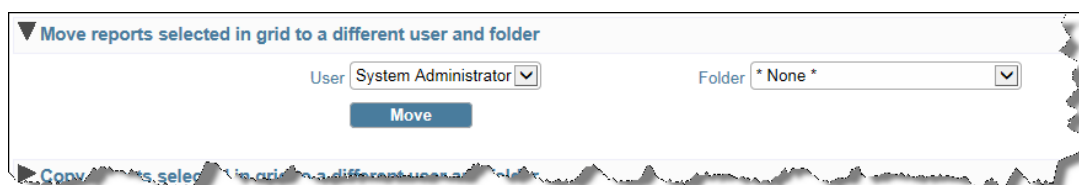
Once you have selected the reports to move, select the user and their folder into which to move the reports. Then click the **Move** button

- **Copy Reports** - Click on the fold to expand it to view the following screenshot:
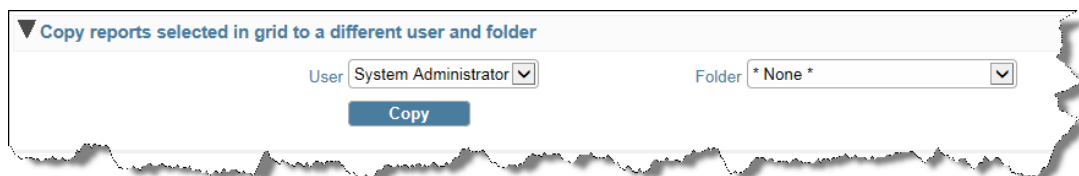
Once you have selected the reports to copy, select the user and their folder into which to copy the reports. Then click the **Copy** button

- Click + to view all the reports that are defined by the filters you have set. Note that if you have selected a large number of reports, this might be a time-consuming task to read the database and to render the grid with the report
- Click — to reduce the grid to the level of public reports and the reports for each user defined within the filters you have set
- **Autosize columns** will set all the columns in the grid to a size where all the contents of all columns will be displayed
- Click on any column title to sort the grid by that column. A second click will reverse the order of the sort

- Drag the edge of column with your mouse to widen or narrow its display
- Click on the **+** or **-** within the report tree to expand or contract that segment of the grid, for example to see all the reports belonging to a specific user
- Use the filter boxes within the title section to select a single category within the column. If the box is a simple text input box, your entry is treated as a wildcard and only the reports that match the wildcard will be displayed.

The columns in the grid have the following functions:

Select	This is the checkbox that allows the selection of a report upon which you want to act. Note that you can check the box within the title to select all the filtered reports and you can check at a node within the tree to select all the reports that reside within that node, such as all the reports that belong to a user or the reports that reside within a specific folder
Public / Private	All *Public* reports are grouped together, within the *Public* node of the tree. Within the grid, the *Public* reports are followed by the reports for each user. If you place your mouse over the folder that contains a user's reports, you wlll see the key data associated with the user's account, such as their user ID, Comany name and whether they are an enabled or disabled user. When you place your mouse over a report, you will see data such as the report ID within the database and whether the report is shared
Folder	This is the folder name, if any, beneath which the report is stored
Type	This is the type of report, such as Column Report, Matrix Report, etc.
Title	The title of the report
Description	The description of the report
Created	The date the report was created. This is one of the filters that may be used when creating the grid
Last updated	The date the report was last update. This is one of the filters that may be used when creating the grid
Last update user	This is the user ID of the person who last update the report
Last run date	The date the report was last run, by any user with permission to the report. This is one of the filters that may be used when creating the grid
Owner	This is the user ID of the person who last update the report. For user's reports this will always be the same person who appears in the Public / Private column

Once you have made the selection of reports within the grid, the following actions can be taken:

Delete Selected Reports - All the reports which are checked in the **Select** column will be deleted permanently from the database. Note that empty folders which result from deleting all the reports they contained are also deleted.

Report Scheduler

The report scheduler allows end users to set up and maintain a schedule for their reports. There is an internal ExtraView task that monitors these reports and at the appointed time, will prepare the report and attach it to an outgoing email. The report schedule maintains a list of recipients for each report. See here for details on setting up and managing the task.

Column reports, Summary reports, Matrix reports, Charts and Container reports may be scheduled and delivered via email.

Security Permission Keys

SR_REPORT_SCHEDULE	This is the overall security permission key that allows access to the report scheduler by user role
SR_REPORT_REPOSITORY_ACCESS	Within the report scheduler, access to write report output to the document repository is controlled by this security permission key
SR_SCHEDULE_ALLOW_ROLE_SELECT	This permission key controls the presence of the list of user roles to which a schedule may be sent, or who may access documents written to the document repository by the report scheduler

Administrative Control

As scheduled reports may have a large impact on system performance, there is an administrative control interface over scheduled reports. Scheduled reports are not necessarily run immediately according to their schedule. They are queued and run sequentially. This is why the scheduler interface says "Start no sooner than". As an administrator, you may still want to adjust, or even disable scheduled reports. Imagine the consequences if the scheduler tried to send out 100 separate reports, to 100 separate users all at 9:00 a.m. on a Monday morning, at the same time these users are all trying to sign on to ExtraView.

Setting the schedule for a report

The administrator can manage the schedule for any user's scheduled report, they may deactivate the schedule, or they may delete the schedule. There is an option to send email to the report owner, when their report schedule is removed. There is also an option that allows the scheduled report to be suppressed, if there are zero results on the output.

Field List Values

For convenience, list values may be maintained from several places:

- Within the **Administration**, **Operational Tasks**, **Manage List Values** menu. For convenience, the list of items on this screen includes Privacy Groups and User Roles. These lists can be viewed and modified both from this administrative section as well as within the Users section. The metadata field lists that appear on this screen include all fields with a data dictionary display type of List, Popup, Checkbox, Radio Button and Tab.
- Within the **Administration**, **Site Configuration**, **Data Dictionary**, where all field display types that support list values have a *List* button to maintain the values
- From within the **Administration**, **Site Configuration**, **Design Center**. When you right-click with your mouse over a field which has list values, an option to modify the list values is present
- For user defined list and popup list display types, there is a layout cell attribute which is added via the **Design Center**, titled *ADD NEW*. This allows the end user to add new values to the list, without being an administrator.

Lists values may be dependent upon values in other lists. This is termed *Allowed Values*, and is covered in more depth in the **Allowed Value** page within the **Site Configuration** section of this guide.

Within all user defined field lists, there is the opportunity to disable a value. Once the value has been disabled, it will not appear in any screen in a selectable manner. If an existing record already has the value set, then the user may leave the value in place or they may select a new, enabled value.

Note that checkbox fields are stored internally as lists that are constrained to just two values, typically Yes and No, On and Off or something similar. You may not disable a value within a checkbox field.

Interest Lists are also maintained for field values within List maintenance. For example, if you want to maintain an interest list for a specific product, or an interest list for Priority 1 issues, then this may be completed through the add and edit functions of each list.

Loading List Values from Files

This feature only works to load values into user defined field lists. It does not work with inbuilt fields such as **STATUS, CATEGORY, RESOLUTION** and user field lists. This is not a real limitation as these inbuilt field lists tend to be small lists. Each user defined field list has a button **Import Field Values**. When this is pressed, the administrator is able to load a file of values into the field.

Importing a file of list values

The format of the file to be uploaded is very simple. It is simply a file with a single value per row. The list of values may be ordered and the order retained when it is imported, or it may be unordered, in which case ExtraView will sort the file alphanumerically. When the administrator presses this button, the following dialog appears:

Specifying the file to upload

When the file is uploaded, you will see a sample of the file:

Sample of uploaded file

Note the prompt that allows you to retain the order of the field values in your import file, or allows the list to be sorted alphanumerically.

List Entries with Allowed Values

Editing lists that are the child in an allowed value relationship is a little different. As seen in the screenshot below, the list of all the potential parents is displayed, each with a checkbox. Also, if Business Areas and Projects are enabled, you must select the Business Area and Project for which you want to set the allowed values (see the next section).

Click on each of the parent values that this child is allowed to be related to. The example shows an allowed value relationship between two fields, where the field named *Building* is the parent and the field named *Floor Number* is the child.

Allowed Value relationships are maintained within their child list

Note that this screen allows you to modify the allowed value children for a single parent, on a single screen. This is ideal for making alterations to existing relationships. If you want to add a series of allowed value relationships for different parents, you may find it more convenient to use the administration screen under Allowed Value Types. Both methods will result in the same changes, but sometimes it is more convenient to take a "top-down" rather than "bottom-up" approach.

Business Areas and Allowed Value Relationships

It is important to note that when Business Areas and Projects are enabled, the allowed values that you set are only for the selected Business Area and Project. This is controlled by additional prompts on the maintenance screen as shown above. You must set up the allowed value relationship for each Business Area and Project.

Note: If the Business Area field is the parent field in the allowed value relationship then you should be aware of how the interaction with the fact that allowed values are dependent upon Business Area works. The implication is that you can only define the child in the relationship to either belong, or not belong to the currently selected Business Area. There is no meaning to allowing the child to be selected with other Business Areas. Therefore, you will only see the currently selected Business Area with a checkbox in the list of possible allowed values.

Note: You can set up a set of default values for a relationship. These will be used, unless overridden by a specific set up within any Business Area and Project.

End-User List Management

End users may add new values to lists under certain circumstances, where the administrator has enabled the feature with security permissions, and has turned on the feature on specific layouts. To an end user, the ability to add a new value to a list appears like this:

Adding a new value to a list

When the end user selects the * **New** * entry in the list, a window pops up, asking for the details of the list. The appearance of this popup varies a little, according to whether the list is an inbuilt list field (such as **product_name**), a child list such as **module_id** or **release_found** or **release_fixed**, or is a user-defined field list. The inbuilt fields require the addition of the field's fixed name, plus its title. Dependent child fields require the parent field to be selected, and the relevant fixed name and/or title. User defined fields only require the title to be input. This is an example of a popup:

Adding a new value from a list field

To configure a field to use this feature, take the following steps. Note that only a small number of inbuilt fields may be used in this way, but any user defined field list may be configured.

Field	Security Permission Key
PRODUCT_NAME	CF_PRODUCT
MODULE_ID	CF_MODULE
RELEASE_FOUND, RELEASE_FIXED	CF_PRODUCT_RELEASE
All UDF's with a display type of List, with the exception of multi-valued lists	CF_UDF_LIST

- For the user role or user roles that you want to be able to add new values to a list, give the security permission key in the above table write access
- The * New * entry is added to the list by adding a layout cell attribute to the field on the appropriate *add* and/or *edit* layout. The layout cell attribute is FIELD ADD NEW and it applies to ADD PROBLEM and EDIT PROBLEM layouts.
- The feature works with UDF list and popup fields on repeating rows as well as on standard list fields
- The feature works with multi-valued list and popup fields on standard list fields only. It does not work with multi-valued list fields on repeating row fields
- The feature does not work with fields being used on reports with the Quickedit mode
- The feature does not work with other types of list fields such as tab and radio button.

User Groups

User Groups are composed of any number of ExtraView users. These users may be added to a named user group in an arbitrary way. They do not need to connected in any way such as belonging to the same user role. Once a user group has been formed there are specific actions that can be taked with the user group. At this time, these actions are:

- Reports can be saved so that they are shared among the members of a user group
- Future release of ExtraView will extend this list.

Managing User Groups

With the use of the SR_REPORT_GROUP security permission key, there is control over access to user groups. Without *write* access to this permission key, end users are not able to save reports to be shared by a user group, and are not able to see the **Manage User Groups** button on the **Report** screen. The permission key that controls administrative access to the **Manage User Groups** function in the **Operational Tasks** tab of administration is named CF_USER_GROUP.

The **Manage User Groups** function looks similar to both end users and to administrators. However, end users may create and manage user groups and then have control over these user groups, but not user groups created by other end users. The **Manage User Groups** function for administrators allows the complete management of all user groups.

Interest Lists

An interest list is placed upon a value or combination of field values in order to notify one or more users when an issue contains the field or fields with the value or values. For example, a product manager may want an interest list on all issues that touch his **product**. The engineering director may want to see automatically, all the issues that are marked with a **severity level** of *critical*.

Both administrators, and to a more limited extent, users may manipulate interest lists. There is one special interest, which affects entire issues, that is not dependent upon the value of any field within an issue. In this case members of the interest list will receive notification upon all changes to the issue.

Interest lists may be global, or they may refer to a single combination of Business Area and Project. Interest lists are not inherited. Like layouts and security permissions.

Interest lists are enabled on each field from within the field's data dictionary definition.

Interest List Definition

Any interest list for any field can be maintained from this screen. Use the **Edit** button by any entry to modify an existing interest list's details, use the **List** button to modify the users who belong to the interest list, and to create a new interest list, click on the **Add** button by the prompt **Add a new interest list**. The following screen appears when you add a new interest list:

Adding a new interest list

If the interest list you create is to be global, create it within the *** Global Interest List ***, in the fields titled Business Area and Project. If you want the interest list to be mandatory for all users, then do not check the box **Users may opt-out from interest list**. If you check this box, users will be able to opt-out of the interest list from their account option screen. You can also disable an interest list, without deleting it from the system with the **Enabled** checkbox.

Note: Only regular users of ExtraView can be added to interest lists. According to the terms of the license agreement, guest and other users who only belong to the behavior setting named LIMITED_USER_ROLE, cannot be added to interest lists.

If you are creating a list value for a field in the list management area and interest lists are enabled for the field, the administrator is allowed to create an interest list for the field at that point, without going to the interest list administration screen. This is purely for convenience.

Also note that if the user has write permission to CF_INTEREST_LIST, then the user will see all the global interest lists and may opt in and opt out of these lists. The user needs write permission to individual business areas and projects to be able to opt in and opt out of any interest lists defined within these.

Personal Administration of Interest Lists

Please see the section on [User Account Maintenance](#) for details on how a user or the administrator can maintain the interest lists for the individual user.

Field-Based Interest Lists

Enabling Interest Lists on Fields

You many enable interest list on fields that have a display type of List, Tab, or Popup. In addition, end users may create an interest list on any issue, provided their role has permission to the security keys PR_ADD_PROBLEM.INTEREST_LIST or PR_RESOLUTION.INTEREST_LIST. Interest list on fields are enabled within the data dictionary.

Option to Enable Interest Lists

- Check the **Yes** radio button on the Email Interest List option to enable the interest list for this data dictionary entry.
- Click the **Update** button at the top or bottom of the screen.

Utilizing Field-based Interest Lists

Interest lists are enabled on each field from within the field's data dictionary definition. Within the **Email Notification** section of administration, there is a menu entry named **Interest Lists**. This administration function is controlled by the security permission key named CF_INTEREST_LIST.

Interest List Access

Note that you may grant non-administrative users permission to this permission key. When you do this, they will see a prompt on the Notification section of their Personal Options screen that allows them to administer the interest lists. Note that this gives them a level

of administrative privilege, including being able to create and delete interest lists, and to add and remove any user to and from an interest list. When you select this menu entry, a screen similar to the following will appear:

Interest List Definition

From this screen you may create new interest lists, modify existing interest lists, or manage the list of users who are subscribed to any interest list. To create a new interest list, click on the **Add** button. The following screen appears:

Adding a new interest list

If the interest list you create is to be global, create it with the Business Area and Project having a value of * **Global interest list** *. If you want the interest list to be mandatory for all users, then do not check the box **Users may opt-out from interest list**. If you check this box, users will be able to opt-out of the interest list from their personal account option screen. You can also disable an interest list, without deleting it from the system with the **Enabled** checkbox. If the field you select for the interest list criteria is a multi-valued list field, then you may only select a single value for the interest list. If you need to select multiple values, you will need to set up multiple interest lists. Interest lists may be built upon field conditions beyond looking for equality with a value. For example, for fields with a display type of LIST, POPUP, and TAB, you may create an interest list condition looking for when a field is **changed to** or **changed from** a specific value. The following screenshot displays how this is achieved.

Setting the operator for an interest list field

After adding an interest list, you manage the users associated with the list using the **List** button by the interest list details. The following screenshot shows the presentation of the interest list members and where they are maintained.

Maintaining the list of users on an interest list

Add new users to the interest list by choosing their name from the select list, then pressing the **Update** button. Delete users from the list by pressing the **Delete** button by their name.

Note: Only regular users of ExtraView can be added to interest lists. According to the terms of the license agreement, guest and other users who only belong to the behavior setting named LIMITED_USER_ROLE, cannot be added to interest lists.

Issue-Based Interest Lists

Creating Interest Lists on Issues

On the *add* screen and from the *edit* screen, there is a checkbox named **Include self on interest list**. When a user checks the box, they add themselves to an interest list for this individual issue. If the user is already subscribed to the interest list then they may uncheck the box to remove themselves from the interest list.

Within a user's personal notification options, they may remove themselves from any issue-based interest list to which they have subscribed, but the only way to add themselves to an issue-based interest list is to update the issue and check the **Include self on interest list** checkbox. The reason for this is that in a large installation, there could be tens of thousands separate issue-based interest lists, and this would be unmanageable on a maintenance screen.

Subscribing and unsubscribing from an issue interest list

Security Permissions & Interest Lists

Due to the potential for change in the valid user roles for any user, and the potential for change in the security permissions of any field, there is no direct connection between the setup of an interest list and the users who may belong to that interest list. For example, it may be that a user belongs to interest list on a field named *any_field*. A specific user may not have permission with their current role to see this field, but they may be on an interest list that contains a reference to *any_field*.

The key point is that although a notification to the user may be generated when the value of *any_field* changes, the user will never see the field or its contents without permission to the security permission key.

Also, remember that there is a behavior setting named EMAIL_NOTIFY_USERS_ALWAYS that can be set to suppress notification to any user, if no field that is visible to them changes upon the update of an issue.

Escalation Rules

Escalation is the process whereby ExtraView will automatically examine a set of issues, based upon time-based criteria and filters that you provide, and send notification warnings based upon any issue exceeding the time-based criteria. For example, you may want to escalate all issues that have remained with a Status value of Open, if the value is not changed within 24 hours.

Escalations in ExtraView are enabled using a **background task**. Escalations will not run unless this task is set up and is running. The parameters of the task include a poll interval which is the number of seconds between each time that ExtraView will run the escalation task to check for issues that should be updated to signify that they have met the rules defined for the escalation to take place. Please see the section on Managing Tasks for more information on setting up the escalation task.

Escalation rules are used to update issues based on the time spent in a given Status. This update will generate notification email, thereby notifying the issue owners.

Escalation rules may be global, i.e. they are defined in the Global Area and Global Project, or they are defined in a specific business area and project. There is no inheritance of escalation rules through the hierarchy of business areas and projects.

Each escalation can use a different **Business Calendar**. Each calendar can define working and non-working days, working and non-working hours. In this way the escalations you define can calculate times accurately according to your business practices. For example, your team may have an SLA that guarantees the resolution of a problem in 3 business days. A correctly defined calendar will skip weekends and holidays that your company respects.

Initial Setup

Two User Defined Fields are defined to support the use of escalation. These are:

- EV_ESCALATED_LAST, a date field
- EV_ESCALATED_COUNT, a number field.

In the data dictionary, you must set both **Allow selection on reports** and **Filter criteria** to have a value of *Yes* on these fields. In addition, you must have granted permission to read and update these fields for the user role of the ADMIN user. These fields do not need to appear on a layout. In a new ExtraView system, these fields are supplied with read and write permissions for the *admin* user role.

Also, the date mask for the ADMIN user account that is running the escalation task must include a time component; it cannot be a simple day format.

Creating and Maintaining Escalation Rules

The initial maintenance screen shows a summary of all the escalation rules in your installation

Maintaining escalation rules

From this screen, you can create new escalation rules, or maintain the list of users attached to each escalation rule. You can also view whether the escalation rules are global or whether they are executed for an individual business area and project.

Also note that if the user has write permission to CF_ESCALATION_RULES, then the user will see all the global escalation rules and may opt in and opt out of these lists. The user needs write permission to individual business areas and projects to be able to opt in and opt out of any escalation rules defined within these.

Maintaining Escalation Rules

When you **Add** or **Edit** an escalation rule, the following screen appears:

Adding and updating escalation rules

Field	Purpose
Escalation rule title	The title of the escalation rule that is presented to the administrator and to users when opting in or out of a rule
Business Area & Project	If the rule being defined is to be global in its scope, then these should be set to * **Global escalation rule** * and * **Global escalation rule** *. A rule will only apply to an individual area and project if you select a specific area and project
Business Calendar	Select the Business Calendar to use in the calculation of the time to escalate the issue. The default calendar provided with ExtraView is 24 hours by 7 days a week by 365 days per year. Create your own calendars to set the workdays and workhours for different escalation rules. Non-workhours and non-workdays will be skipped in the calculation of escalations

Enabled	This allows the rule to be turned off and turned on
Escalation method	Issues may be escalated based upon the amount of time spent in a single status without change of that status, or they may be escalated according to the value of a date or day type field on that is on the edit layout of the issue within it's business area and project **Time in status escalations** The number of hours an issue has been in a given status before it is escalated. Partial hours are supported using decimal points in the number. Note that there is an interaction with the timer that is used to escalate issues, so an issue is not escalated immediately that the elapsed time is reached, but is escalated on the first occasion following the reaching of the elapsed time that the timer fires. Click on the **Escalate using time in status** to use this method of escalation.

Setting the elapsed time for an escalation based on status

Formula-based escalations You may escalate an issue based upon the value of a date field contained on the edit layout of the issue, plus or minus an offset. For example, you may set a reminder 1 day before an item is due, or you may set a reminder when an item reaches an expiry date. The escalation will occur at the time the evescalation script fires following the rule you create. Click on the **Escalate using formula based on a date field** radio button for this type of escalation.

Basing an escalation on a date field in an issue

The offset may be left blank, in which case the escalation happens at the next time the script fires, following the name of the date field you select. You can set the offset to a positive or negative number of hours as required.

Frequency	After the issue is escalated, this is the amount of time that elapses, before the issue is again escalated. You may enter any number of hours or part of an hour as a decimal portion.
Sort Sequence	Providing a sort sequence for escalation rules has two purposes. First, this will affect the order they are displayed in. Rules with no sort sequence are displayed in alphabetic order. Secondly, and importantly, rules are executed in the order of the sort sequence. Therefore, if you have a number of rules which could conflict with each other, the order becomes significant, and you should provide a high-order sort sequence for your most important rules, ensuring they are executed before less important rules.
Action	This is where you place assignments to fields, to update these fields to the new value(s) you provide within the action. Actions are delimited with semi-colons and follow the same syntax rules as described in the section in this guide named Business Rules. Leave this field blank if you simply want to have the escalation rule notify users without changing the values of fields in the issue. You can notify additional users in the **Action** field, either by placing their explicit user ID in the field (followed by a semi-colon) or by using the data dictionary name of a field such as ORIGINATOR, ASSIGNED_TO or OWNER. In the second case, the person whose value is in that field is notified upon the triggering of the escalation.
Escalation rule criteria	These are filters that are used to provide the specific set of values, in addition to any date value you select. For example, you may set a filter to only fire the trigger base on the value of the product, or based on the current priority of the issue.

Note: There is only one field named EV_ESCALATED_LAST. This is shared by all escalation rules. The rule that fires first will update this field, perhaps preventing other rules from being triggered. If you want other rules to fire in addition to the first one

triggered, set their interval to 0; these rules then fire every time the escalation engine runs. This may not be desirable, mainly for performance reasons, but is a way to solve the problem.

Note: If an issue is escalated according to one rule, it may no longer meet the same criteria to be escalated again, but it may be triggered by another rule using the same status. However, the pre-existing value of EV_ESCALATED_LAST may cause premature firing of the new rule. In this case, write a business rule to clear EV_ESCALATED_LAST when the issue is updated to move outside the scope of the first rule.

Example Escalation Rules

Example 1

Escalate all issues with **PRIORITY**=*P 1* and **STATUS**=*Open* if status has not changed for at least 4 hours. Keep escalating issues every two hours thereafter, as long as the **STATUS** and the **PRIORITY** remain unchanged. At every escalation, add a new **COMMENT** field with the specified text to the qualified issues and send the notification to all members of the escalation rule.

Field	Value
Escalation rule title	Escalate all P1 and Open Issues
Business Area	Global Area
Project	Global Project
Elapsed time	4
Frequency	2
Enabled	Checked
Action	COMMENTS = "This issue is escalated because its priority is P1 and its Status is Open"; COMMENTS &= SYSDATE;
Escalation rule criteria	STATUS = Open PRIORITY=P 1

Example 2

Escalate all Bugs issues where the **SEVERITY** is *High*, if the **STATUS** has not changed in at least 2 hours. Keep escalating issues every hour as long as the **SEVERITY** and the **STATUS** remain unchanged. At every escalation send notification email to the users who opt-in to the escalation rule.

Field	Value
Escalation rule title	Escalate all issues which have a High severity level for more than 4 hours
Business Area	Bugs
Project	Bugs Data
Elapsed time	2
Frequency	1
Enabled	Checked
Action	COMMENTS = COMMENTS = "This issue is being escalated because its Priority is P 1 and its Status is Open";COMMENTS &= SYSDATE;
Escalation rule criteria	SEVERITY_LEVEL=1;

Example 3

Escalate all Customer **P 2** issues 24 hours after they were due, based upon the value of a date field named **due_date**, if the status of the issue is **Open**. Set the **Priority** to a new value of **P 1**.

Field	Value
Escalation rule title	Escalate Customer P 2 Issues 1 day after creation, if they are still open
Business Area	Customer Issues
Project	Customer Issues Defaults
Date Field	Due Date
Offset	24
Frequency	24
Enabled	Checked
Action	COMMENT = "This customer issue has been escalated to P 1 as the SLA for a P 2 issue mandates the issue must be handled within 24 hours"; PRIORITY = "P 1";
Escalation rule criteria	Status = Open

Business Calendar

Business calendars allow the definition of work days, holidays and non-work days. The calendar is used in computations of elapsed time between two moments, to only count the business time involved.

For example, you might want to escalate issues that have remained in a specific status for more than 3 business days, but omit Saturdays and Sundays. Similarly, you might want to omit hours outside of 9:00 am through 5:00 pm when you calculate the elapsed time.

The Business Calendar feature allows the setup of any number of calendars for different purposes, so if different groups or different parts of your organization work different hours or days, then this can be taken into account. Each calendar is independent and may, for example, have different company holidays.

ExtraView comes with two default calendars, one named 24_BY_7 and one named WEEKDAY. The 24_BY_7 calendar has no holidays, or non-workdays and therefore counts the absolute time between two dates. The WEEKDAY calendar has Saturday and Sunday set as non-work days, and the work hours are set to 9:00 am to 5:00 pm. These default calendars can be updated, or you can create any number of new calendars.

There is a behavior setting named RULES_DEFAULT_CALENDAR. If a business rule does not specify a calendar in a directive, then this calendar will be used for computations involving dates. If this does not have a valid entry, then the 24_BY_7 calendar is used within the rules.

You can use a business calendar by specifying which one to use within escalation rules or within business rules. When you update a calendar, the screen will look similar to this:

Save	Save As	Return	Print Page	**Business Calendar**

Directions

1 - Provide a name and title for the calendar
2 - Set the workdays, and workhours if required
3 - Click with the mouse left button to alter a day on the calendar to a holiday
4 - Click with the mouse right button to override the work hours for a day

Calendar Name WEEKDAY

Calendar Title Monday to Friday, 9 a.m. to 5 p.m   [Localize]

Time Zone ? (GMT -8:00) America/Los_Angeles

Work Days ?   **Work Hours ?**

Work Day		From		To
☐ Sun				
☑ Mon		9:00 AM	to	5:00 PM
☑ Tue		9:00 AM	to	5:00 PM
☑ Wed		9:00 AM	to	5:00 PM
☑ Thu		9:00 AM	to	5:00 PM
☑ Fri		9:00 AM	to	5:00 PM
☐ Sat				

◀◀ **2011** ▶▶

January

Sun	Mon	Tue	Wed	Thu	Fri	Sat
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

February

Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28					

March

Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

April

Sun	Mon	Tue	Wed	Thu	Fri	Sat
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

May

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

June

Sun	Mon	Tue	Wed	Thu	Fri	Sat
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

July

Sun	Mon	Tue	Wed	Thu	Fri	Sat
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

August

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

September

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

October

Sun	Mon	Tue	Wed	Thu	Fri	Sat
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

November

Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

December

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Setting up a calendar

- Calendar names conform to the standard naming convention
- Calendar titles conform to the standard convention of other titles, and may be changed and may be localized
- The Time Zone is typically the time zone of the owner of the calendar
- You use the Work Days checkboxes to nominate the standard work and non-work days through the calendar
- You use the Work Hours select lists to nominate the standard work and non-work work hours for each work day, through the calendar
- Use the arrows on either side of the year to move forward and backwards, in one-year intervals
- The numbers on the calendar with the grey background are the non-work days; numbers with a white background are work days
- To set a work day to a holiday, click once on the date with your mouse. To change the day to a non-work day, click once again. To restore the date back to a work date, click again. At this point in time, there is no difference between holidays and non-work days. In the future, there may be differentiation between holidays and non-work days
- You can override the work hours for any work day on the calendar by right-clicking with the mouse on the day. When you do this, a popup window allows you to set new work hours for that day only. Once you have overridden work hours for any day, a small clock will appear by the date, to show that the day has different work hours than the base work hours for the day. The following screenshots show how to override the time on a work day:
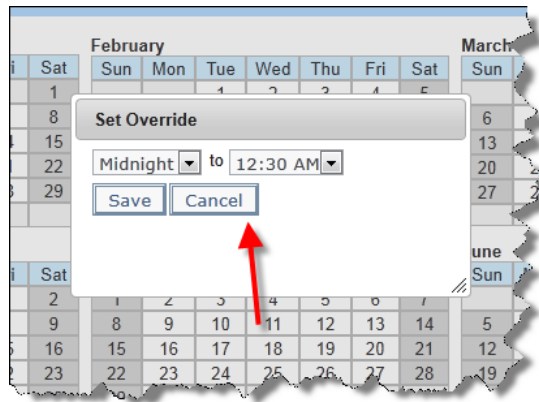
Overriding work hours
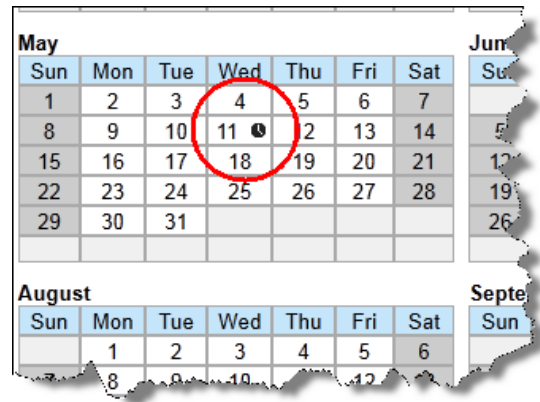
Seeing overriden work hours

Custom Email Templates

This feature allows the administrator to create standard email templates for email notification in situations where a standard response is warranted. For example, if your product team had a number of issues entered by the members of a beta test group, and they wanted to be able to send a standard confirmation to a group member whenever there was a resolution, this feature could be employed for this purpose. Another example is that a customer support representative could choose from one of several standard replies to a customer reporting a problem. These replies could be geared to acknowledging receipt of a problem, informing the customer of progress towards resolution of a problem or notifying the customer that a problem has been resolved. Custom email notifications may be sent from the *edit* screen of an issue, using the **Email** button on the menubar, or they may be sent using an automated process via a business rule. From the administration **Operational Tasks** menu, click the **Email Templates** button. The following screen appears:
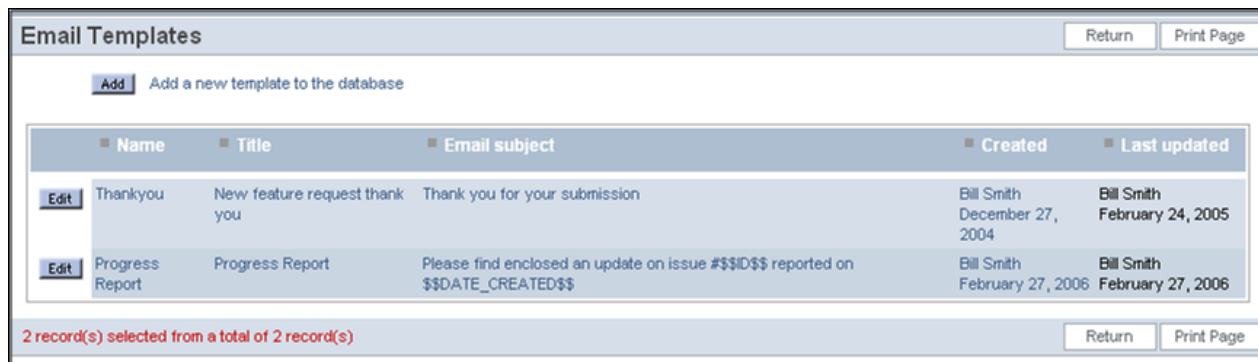
Email Templates screen

To add a new email template, click the **Add** button. The following screen appears:

Add Email Template screen

- Provide a name for the template
- Provide a title for the template
- Provide the email subject line for the outgoing email. This may include tokens that will be replaced by field values when the email is sent, as described below
- If you want to specify the email as originating from a specific user, select the user's name from the *From User* list. This allows you to select a single user whose primary email address will be used as the *From Address*. Note that you can select the current user, in order to allow the sender to be the person who initiates the email. If this option is not selected, the email address in the EMAIL_FROM_USER_ID behavior setting is used
- If you want to specify the *Reply-To User* of the generated email to be set to the primary email address of a specific user, select their name from the **Reply-To User** list. This allows you to select a single user whose primary email address will be used as the reply-to address. Note that you can select the current user, in order to allow the recipient of a reply to be the person who initiates the outgoing email. If this option is not selected, the email address in the EMAIL_FROM_USER_ID behavior setting is used
- Indicate whether the mail is to be sent as plain text or as HTML using the checkbox **Send Email as HTML**
- If you intend for the recipient of the outgoing email to reply to the email when you generate the email from the *edit* screen of an issue, and for EVMail to process the reply, such that it will update the issue with the recipient's reply, then check the box **Delimiter in Outgoing Mail**. This introduces an invisible delimiter into the email so that EVMail will discard all the mail following this delimiter, leaving only the recipient's reply to the mail being added back into the issue. The text for the delimiter is defined in the behavior setting named EVMAIL_DELIMITER_TEXT.
- The outgoing email can be saved back with the issue from which you originated the template. You enable this behavior by checking the **Save Outgoing Emails** option
- If you are saving the outgoing email, then the default is to save it as an attachment to the issue. However, if the outgoing email is text, then you have the option to save the outgoing email into a field with a display type of *log area*. You can choose from the list, to either save as an attachment, ot from the list of available *log area* fields. Make sure the users who use the template have write permission to this field
- When the outgoing email is attached into the issue from which it was generated, it can be viewed in the normal way that you view all other attachments. Note that the header information used to process the emails is not visible with one exception. The exception is that you will be able to see the names of attachments you might have added to the outgoing email. This allows better auditing of the history of email transactions. Just like in email clients, if the user wants to see all the email header information, they can view the source of the downloaded attachment
- Use the text area to compose the email that is to be sent. The text for the email can be plain text, or can be HTML. The appropriate type of text box will be displayed, dependent upon the selection for whether the mail is to be sent as HTML or as plain text.

Within the subject and the body of the mail, you may insert tokens that are replaced when the email is sent. These tokens represent the value associated with the field from the record that is currently displayed when the mail is being sent. For example, if you want to

substitute the issue ID in the mail, you would use $$ID$$. To insert the issue status, you would use $$STATUS$$. Valid tokens are data dictionary field names, data dictionary UDF's as well as the following:

$$SYSDATE$$	The current date, including time
$$SYSDAY$$	The current date
$$SITE_URL$$	The URL of the site
$$EXT_SITE_URL$$	The external site URL. This provides a link within the email which can be used to access ExtraView from the email being generated. The most common purpose for this is to drill down to an issue using the $$ID$$ or $$ALT_ID$$ of the issue.

It is possible to substitute the **From User** and **Reply-To User** values at runtime, by populating their values from fields on the underlying *edit* screen. The fields used to populate the template must have a display type of either TEXT FIELD or USER and must be single-valued. The template will contain $$FIELD_NAME$$ in the **From User** and/or **Reply-To User** fields. The $$FIELD_NAME$$ must exist on the *edit* screen. When the user enters data into the *edit* screen, they may either enter a valid User ID or an explicit email address.

An example email template body:

Dear $$CUSTOMER_NAME$$,

We are in receipt of your issue, reported on $$DATE_CREATED$$, is receiving our prompt attention.

Our records show that you reported the issue with the following description:

$$DESCRIPTION$$

We will contact you as soon as we can provide a solution to your report.

Thanks,

$$OWNER$$

would produce email output similar to:

Dear Brian Jones,

We are in receipt of your issue, reported on 12/11/2002, is receiving our prompt attention.

Our records show that you reported the issue with the following description:

I cannot access the top-level widget within the cabinet of the power supply, unless the power is turned off and the unit is disconnected from the power. I understood that changes like this could be achieved without powering down the equipment.

Can you please provide a solution?

We will contact you as soon as we can provide a solution to your report.

Thanks,

Tony Smith

You may include image fields and document fields as part of a custom email. When you place an image field name into a custom email (e.g. $$MY_IMG_FIELD$$), the image will appear inline within the body of the email. If you place a document field (e.g. $$MY_DOC_FIELD$$), then the name of the document file is placed into the email body in place of the token, and the document is added as an attachment to the outgoing email notification.

As well as fields that you can refer to with the tokens surrounded with the $$ characters, there is a selection of inbuilt variables that you can refer to as tokens. This list is:

- APP_HOME - the relative path to the WEB-INF folder on your application server
- BG_COLOR - one of the two background colors used to draw tables in ExtraView
- BG_ALT_COLOR one of the two background colors used to draw tables in ExtraView
- BROWSER_NAME - the name of the user's browser that generates the email
- COMPANY_LOGO_IMG_HOME - the relative path to the location where the CompanyLogo.gif is stored
- DEFAULT_FONT - the name of the default font used in the installation
- EXT_URL - the absolute path to the ExtraView installation
- FIXED_WIDTH_FONT - the name of the fixed width font used in the installation
- IMG_HOME - the relative path to the location where the ExtraView images are stored
- JAVASCRIPT_HOME - the relative path to the location of the ExtraView JavaScript files
- LABEL_COLOR - the color of labels on ExtraView screens
- STYLESHEETS_HOME - the relative path to the location where the ExtraView stylesheets are stored
- WINDOW_BGCOLOR - the background color of the ExtraView screens

Once you click the **Add** button to save the template, it will appear in the template dropdown box for users when they click the **Email** button from a given issue's Edit screen.

Note: A field named EMAIL_ADDRESS is available as a UDF field with a display type of Text. This field may be placed upon layouts. It serves a special purpose. When a user accesses the custom email function from the edit screen, to send an ad-hoc email, or an email created from a pre-defined template, this field will be used to automatically populate the email address to which the mail is to be sent. This simplifies communication to users who, for example, enter an email address when reporting an issue. The value stored in this field automatically gives a return address.

Managing Tasks

If ExtraView Corporation is hosting your installation, you do not have direct access to the file system of the server to configure, alter or use this feature without contacting ExtraView support.

This administration section allows the control of tasks that are connected to ExtraView. These tasks control important features, and run in the background of the server. Not all tasks should be configured for all installations. Many tasks handle the interaction between ExtraView and external software, such as your mail server. The information that provides the link between ExtraView and the external software is managed within the properties of each task. The **Add** / **Edit** buttons on the Task Manager screen provide access to configure the connection between ExtraView and the external software.

Often, the connection information to external software will contain sensitive connection information such as addresses, account information and passwords. It is strongly suggested that you only provide permission for appropriately credentialed users to access the Task Manager utility. ExtraView Corporation strongly endorses the use of strong passwords which contain combinations of letters, numbers and special characters. However, it is not recommended that you include the = character within password strings. It has been noted that some ExtraView environments do not handle the = character within passwords correctly when validating the setup of a task.. If the ExtraView application server log indicates that the password is **null**, then the most likely reason is the presence of an = character.

When you enter the default ExtraView setup, you will see a screen similar to this:

	Task name ▲	Title	Node ID	Current Status	Start Option	Poll Interval	Thread and Log Information	Created	Last updated
Edit	ADD_UPDATE	Add and Update Background	WS_A(Current Node)	STARTED	START_NOW	30	Priority: 5,Thread State: running, Interval: 30, secs since execution:4 [Download Log]	System Administrator 04/24/2013	System Administrator 04/24/2013
Edit	FULL_TEXT_SYNCHRONIZE	Quickfind Synchronize Task	WS_A(Current Node)	STARTED	START_NOW	60	Priority: 5,Thread State: running, Interval: 60, secs since execution:7 [Download Log]	Bill Smith 04/09/2012	Bill Smith 07/06/2013
Edit	METADATA_EXPORT	Metadata Exporter	WS_A(Current Node)		AUTO-START	30	2,213 hours 28 minutes 20 seconds since last execution (approx.). [Download Log]	System Administrator 10/06/2011	Bill Smith 04/06/2013
Edit	METADATA_IMPORT	Metadata Importer	WS_A(Current Node)		AUTO-START	30	8,016 hours 10 minutes 52 seconds since last execution (approx.). [Download Log]	System Administrator 10/28/2011	Bill Smith 08/07/2012
Edit	SCHEDULED_REPORT_TASK	Report Scheduler Task	WS_A(Current Node)	STARTED	START_NOW	120	Priority: 5,Thread State: running, Interval: 120, secs since execution:67 [Download Log]	ExtraView 07/12/2011	Bill Smith 07/07/2013
Edit	SESSION_MONITOR	Session Monitor Task	WS_A(Current Node)	STARTED	START_NOW	300	Priority: 5,Thread State: running, Interval: 300, secs since execution:64 [Download Log]	ExtraView 07/12/2011	System Administrator 07/12/2011
Edit	TASK_CONTROL_TASK	Task Control Task	WS_A(Current Node)	STARTED	START_NOW	30	Priority: 5,Thread State: running, Interval: 30, secs since execution:7 [Download Log]	ExtraView 07/12/2011	System Administrator 07/12/2011

7 record(s) selected from a total of 7 record(s)

Managing tasks

Note the **Show non-local nodes** prompt on the screen. If you are running in a clustered system, clicking this will show you all the tasks running on all nodes. The task management utility runs as an interactive administrative utility controlled by the CF_MANAGE_TASKS security permission key. This utility displays a list of tasks currently defined. New tasks (adding an existing task object to a node, for example) can be added, and existing tasks may be edited. The task management utility allows the administrator to create/modify and delete background tasks that may run in any node clustered on the ExtraView database. The standard tasks are automatically cloned and started as necessary when a new node joins the application server cluster. This document makes no attempt to describe how to create your own tasks that can be managed by ExtraView. At some future time this interface will be documented and opened for customer use. The tasks managed by ExtraView are detailed in the following sections:

Session Monitor Task

This is an internal task which should not be altered by the administrator. It must be running for ExtraView to function correctly. This task manages the internal ExtraView sessions that are created by users as they perform their work inside ExtraView. For example, sessions are created when users add or update issues or when they run reports. This task is responsible for cleaning up sessions which are left behind when users do not "clean up" after performing a task. This improved memory utilization on the server. For example a user may simply close their browser without updating an issue on the screen, and after some time the session monitor task will recognize that this has happened and will remove the session objects associated with the browser window.

Task Control Task

This is an internal task which should not be altered by the administrator. It is the overall control for the task manager. It is part of the integral working of ExtraView and should not be altered by the administrator. The task control task on any node ID is special, as this task is used to control all other tasks that run under task management. It must be running for ExtraView to function correctly. Therefore, if the task control task is not running on a node, there can be no changes to other task states on that node. There must be a task control task configured on each node where background tasks run and can be modified. There is no reason for the administrator to alter the default properties of this task. Doing so may result in other tasks not executing correctly, or not executing at all.

Metadata Export Task

This is an internal task which should not be altered by the administrator. This task is responsible for ensuring the correct handling of metadata exports performed by an administrator. There is no reason to alter the default properties of this task.

Metadata Import Task

This is an internal task which should not be altered by the administrator. This task is responsible for ensuring the correct handling of metadata exports performed by an administrator. There is no reason to alter the default properties of this task.

Batchmail Task (Outgoing email)

This task is responsible for handing off outgoing email notifications created within ExtraView, to your company's email server. It keeps a log of all activity and removes the email text from the server once it has been successfully sent. If there was a problem sending email to the email server, then the email text is left on the server. In many instances the email will be sent once the problem is resolved.

Quickfind Synchonization Task

This is the task that routinely scans newly entered issue text to be indexed for use by the Quickfind search engine. This task is not required if Quickfind is not turned on within ExtraView.

ExtraView Peer Integration Task

This task is only configured if you require ExtraView to be synchronized with one or more remote ExtraView installations.

Email Escalation Task

This task initiates the escalation mechanism which automatically updates issues and notifies users according to criteria that is defined by the administrator. When installing ExtraView you should add this task to be started automatically by ExtraView. The default poll interval is 2 minutes (120 seconds). This task does not require further configuration. Note that the task runs using the ADMIN user account. When configuring the escalation task, you should also ensure that the date format selected for the ADMIN user account includes a time component. If this is not done, the escalation task will not run correctly.

EVMail Task (Incoming Email)

This task controls the functioning of the EVMail utility that interfaces and imports incoming email to Extraview. This enables the creating of new issues or the updating of existing issues within your ExtraView system, from incoming email to an email address that you use for this purpose. First you create the task, and then you configure the task. To create a new EVMail task on the current node, add the task and select a poll interval of 30 seconds or so, with START_NOW as the start option. This sets the frequency with which EVMail checks for new incoming emails.

Report Scheduler Task

The Report Scheduler Task must be running in order to deliver scheduled reports that users create for email delivery. You should create this task so that it starts at boot time. The default is that this task will execute every two minutes, but this time can be adjusted to accommodate your specific requirements.

LDAP Background Task

This task should only be configured when you are connecting your installation to an LDAP server. It is optional to run this task. When it is running, it will synchronize the LDAP server data that is required within ExtraView on a periodic basis. This avoids ExtraView reaching out to the LDAP server for user information as it is required, and uses the information previously read into ExtraView. This task should be configured in preference to the *LDAP Synchronize Task*.

LDAP Synchronize Task

This task should only be configured when you are connecting your installation to an LDAP server. It is optional to run this task. When it is running, it will synchronize all the LDAP server data to ExtraView on a periodic basis. It is preferable to use the *LDAP Background Task*.

Template Loader Task

This task only needs to be configured and started if you are pre-caching layouts at the time of starting the system server. This is useful on systems with complex or very large layouts. See the page here for more details. Note that you should configure the poll time to zero so the task only runs once at system startup, and does not run continually.

Managing Tasks in a Load-balanced Environment

If you have a cluster of application servers supporting your ExtraView environment, there are some important points to be aware. Principally, some tasks should be run on all nodes of the cluster, and some should not be run on all nodes of the cluster. We suggest you configure a clustered environment in the following way:

- Configure the multiple nodes of the application server, whether you are using WebLogic or multiple load-balanced Tomcat servers
- Select a single node and configure the tasks on that node only. We suggest you use the first node in the cluster for this purpose
- The Task Control and the Session Monitor tasks must be configured on all nodes in a load balanced environment
- Only one EVMail task should be configured on one node, per installation - that is one per mailbox checked
- Only one Escalation task should be configured per installation - for multiple node installations, create a single Escalation task on a single node only
- Only one Report Scheduler task should be configured per installation
- Configure only one Quickfind task per installation on a single node. However, the Quickfind index location must reside on a file system location accessible to all nodes in the cluster
- With the Batchmail outgoing email task, there are options:
    - If all nodes are able to write to a common email directory, only configure Batchmail on a single node
    - If the nodes are on separate application servers, each node must be able to write to a filesystem location defined by the path in the behavior setting named EMAIL_DIRECTORY. You configure all nodes to have the same path configured and a Batchmail task must be configured per separate email directory folder

  Overall, only a single Batchmail task should be polling the email directory folder. If you have multiple separate physical locations for this folder, you need multiple Batchmail tasks, one for each location

- Once the tasks are configured, restart ExtraView on all nodes of the cluster

In the case where the load-balancing algorithm does not send any users to the node where your managed tasks are running, the EVMail and escalation tasks will not operate. Simply starting the application server does not initialize the ExtraView application, and thus if nobody signs onto your installation, no tasks will start up. This is not simply a load-balancing issue - if your system is rebooted and restarted, and nobody signs on, the application is not initialized and the tasks are not started up. For correct operation, all nodes of the cluster should be started. One way to make sure that your site is initialized when the application server is restarted is to configuring a monitoring tool such as Nagios, and to use the ExtraView get_heartbeat API call against each individual node on a scheduled basis - this will ensure that if an application server is restarted for any reason, then the API call will "touch" each node, and start all the copies of ExtraView in the load-balanced installation.

Error Conditions

Tasks are often reliant on programs running external to ExtraView. When an external program fails, the ExtraView task can fail. The task may be able to recover when the external program is restarted. When a task is seen to be in an error state, email is sent to the ExtraView administrator. If ExtraView is able to recover from the error, then another email is sent to inform the administrator of the recovery.

User Custom Tasks

These are used to create tasks for specific purposes within an installation. Knowledge of user custom coding is required to set up and use these tasks. Please contact ExtraView Support if you believe you have a requirement for a customized task.

Task Options

Creating a Task

Each task has the following properties:

- **Task Name** – this specifies the functional operation that the task performs
- **Title** – this is a localizable title that is visible in the utility to identify the task
- **Node ID** –: this is a pull-down list that is populated by the names of the nodes that are started on the cluster. The node ID is specified in the Configuration.properties file as the WEB_SERVER_NAME. The name of each node ID must be unique in the cluster
- **Start Option** – This may be one of the following:
    - **None**: no action to be taken for/by the task
    - **START_ON_BOOT**: start up the task when the application server node is started
    - **START_NOW**: start up the task immediately and also when the node is started
    - **STOP_NOW**: stop the task immediately and discard its thread
    - **AUTO_START**: the task will be started by ExtraView when it is required. For example, the Metadata Export and Metadata Import tasks are only started when they are required
    - **STOPPED**: the task is currently stopped, but if required again, it will be started automatcially. For example, the Metadata Export and Metadata Import tasks are stopped when they have completed their functionality, but can be started again if required

- **Poll Interval** – This is the number of seconds between each triggering of the task. Once the task has finished, it stops executing until the poll interval is reached
- **Class Name** – This is the name of the Java class that contains the code for the task to be executed.

Modifying a Task

After viewing the list of tasks, the administrator may modify some aspects of any one of the tasks:

1. Title
2. Node Id
3. Start Option
4. Start Option
5. Poll Interval
6. Class name

Generally, this function is used to modify the start option, either to start or to stop a task from executing or to alter the frequency with which the task executes. No matter what the prior value is for start option, when the user updates the task, the specified start option is carried out by the task control task. START_NOW and STOP_NOW have immediate consequences on the target node, as long as the task control task (TASK_CONTROL_TASK) is running.

Deleting a Task

One of the options in the task management utility EDIT screen is to delete a task. Note that this only removes the task from the task management utility and has no effect on the background processing of the task. If it is already running, it continues to run. If the task is stopped or in error it remains in that state.

Configuring EVMail

EVMail is configured as a task. If you are setting up EVMail for the first time, you will need to create the task. Use the button labeled **Add** to add the task from the **Task Manager** screen, or use the **Edit** button to modify the attributes that control the EVMail task. Select the **EVMail Task** task name and observe a screen similar to:

EVMail configuration screen

Basic Configuration

To configure EVMail, you will need a properly configured POP3 mailbox and you will need its username and password. You will also need to know the port to use for communication with the mailbox. It is strongly recommended that you use the POP3S protocol for a higher level of security than is achieved with POP3.

Next, you will need to identify an ExtraView user account, and a user role that EVMail will utilize. These, along with a nominated Business Area and Project, determine the *add* and *edit* screen layouts that will be used by EVMail to create new issues and to update existing issues. When selecting these values, take into account the standard licensing condition of ExtraView, i.e. non-licensed guest role users cannot update issues. Therefore, if you select a guest user for the runtime account, only new tickets will be created and you will not be able to update issues.

Note: The MAILBOX_USER value should never be the email address of a user within the ExtraView system. If you do this, there is a chance for an infinite loop to be set up where incoming emails generate notification to this user which then reply to the same address, thereby generating a new event which repeats itself.

Note: The MAILBOX_USER value should always be set to the same value as the behavior setting named EMAIL_FROM_ADDRESS. This ensures that updates to issues made by users and resubmitted are sent back to the same email address that ExtraView monitors.

You will then need to set up the fields that the EVMail task will populate - there are some field options that map to portions of the incoming email message according to whether you are inserting a new issue or updating an existing issue.

EVMAIL_BODY_UDF and/or EVMAIL_BODY_HTML_UDF	For a new issue being created, use either or both of these fields to store the body of the email message. This allows you to store the body as text, HTML or both. Typically this will be mapped to the DESCRIPTION field or an HTML AREA field
EVMAIL_BODY_UPDATE_UDF and/or EVMAIL_BODY_HTML_UPDATE_UDF	For issue updates, use either or both of these fields to store the body of the email message. This allows you to store the body as text, HTML or both. Typically this will be mapped to the COMMENTS field or an HTML AREA field, so that successive updates will create a new entry in this log area field. Note that the COMMENTS field may use an option in the data dictionary to use it directly with HTML.
EVMAIL_SUBJECT	This ExtraView field will be where the subject of the email message is placed. Typically this will be the SHORT_DESCR field

You can select other fields and set values, allowing you to set list values and user fields when creating new issues. The syntax for these fields is EVMAIL_*FIELDNAME*, where *FIELDNAME* is the field name of the field being mapped. For example, these are valid expressions:

EVMAIL_AREA = 10	You must use the AREA_ID for the AREA field
EVMAIL_PROJECT = 19	You must use the PROJECT_ID for the PROJECT field
EVMAIL_STATUS = NEW	You must use the name for inbuilt fields
EVMAIL_ORIGINATOR = BSMITH	You must use the User ID for user fields
EVMAIL_COMPONENT = 12435	You must use the UDF_LIST_ID for user defined list fields
EVMAIL_INSTRUCTIONS = Here are instructions	You can popuplate text, text area and log area fields with text

Note: Updating existing issues will only result in the update of the EVMAIL_BODY_UPDATE_UDF field.

The values for these fields are database values or the internal ID value of the field (i.e. the UDF_LIST_ID value), not the title of the field. You can find the internal ID value by placing your mouse cursor over the list value in its maintenance screen, and looking at the status bar of your browser.

Finding the UDF_LIST_ID internal ID value

You can use a similar technique to find the AREA_ID for the AREA field and the PROJECT_ID for the PROJECT field. Look for the p_aid parameter for the AREA and the p_pid parameter for the PROJECT field.

All fields mapped in the Configure EVMail screen must be on the *add* layout, for the ExtraView user ID, their role, their default Business Area and their Project.

The default behavior of the EVMail task is to take the email address in the **From**: section of the email message, and to search for a matching user in ExtraView who has that same email address in either their primary or secondary email address field in their account. If a user is found, their User ID is used to populate the ORIGINATOR field. If no user is found, the user specified by EVMAIL_ORIGINATOR will be set into the ORIGINATOR field.

Full List of EVMail Settings

EVMAIL_*FIELDNAME*	See the description above on how to map different fields from EVMail to ExtraView
EVMAIL_ADDL_DELIMS_TEXT	This setting is used in conjunction with the behavior setting named EVMAIL_DELIMITER_TEXT and the EVMail configuration setting EVMAIL_ADDL_DELIMS_HTML. Together they provide additional controls on incoming email to the EVMail feature, to parse the email looing for the new content to include and to inhibit the repeated addition of older content of the emails. This is typically seen when users reply to emails add simply append their new message to the thread of existing messages. It is often desirable to only include the latest mail as an update to the ExtraView issue, and not to repeat existing, older comments. These settings are only used when updating existing issues within the ExtraView database, and not used when new issues are being inserted. The EVMAIL_ADDL_DELIMS_TEXT setting is used when the incoming email has a contentType of **text/plain**. Any number of delimiters may be specified, delimited within the settings with ||. **Example:** `EVMAIL_ADDL_DELIMS_TEXT = delimiter string 1 || delimiter string 2 || delimiter string 3` Three delimiters are specified in the example, and the incoming mail is parsed looking for all occurences of the three strings
EVMAIL_ADDL_DELIMS_HTML	See EVMAIL_ADDL_DELIMS_TEXT above. The principal difference is that the delimiter being parsed within the incoming email has a contentType of **text/html**. Different email clients generate different delimiters and some experience with examining different email headers is

required in order to be able to successfully determine the range of delimiters you expect.

Example:

EVMAIL_ADDL_DELIMS_HTML = <div style="border:none;border-top:solid #B5C4DF 1.0pt;

This delimiter string is often seen within Microsoft Outlook emails.

EVMAIL_ALLOW_HTML_MSG	Valid values are YES, NO and PARSE. If this is not specified, the default value is PARSE
EVMAIL_ALLOW_HTML_MSG_FAIL_TEMPLATE	This is used only if the EVMAIL_ALLOW_HTML_MSG parameter is set to is NO. If it does not exist, then SMTP_MSG_FAIL_TEMPLATE is used, if it exists
EVMAIL_ATTACH_EML	This parameter allows you to attach a copy of the received email in EML format to the created or edited issue when it is set to YES. A setting of NO or the absence of this entry means the attachment will not sent with the mail
EVMAIL_BODY_UDF	For a new issue being created, use this field to store the body of the email message as text. Typically this will be mapped to the DESCRIPTION field
EVMAIL_BODY_HTML_UDF	When you want to retain the HTML within the body of an incoming email, use this in place of, or as a complement to EVMAIL_BODY_UDF. This field should be an HTML AREA field
EVMAIL_BODY_UPDATE_UDF	For issue updates, use this field to store the body of the email message. This allows you to store the body as text. Typically this will be mapped to the COMMENTS field, so that successive updates will create a new entry in this log area field.
EVMAIL_BODY_HTML_UPDATE_UDF	When you want to retain the HTML within the body of an incoming email that is updating existing issues, use this in place of, or as a complement to EVMAIL_BODY_UDF. This field should be an HTML AREA field or a LOG AREA field configured to use HTML
EVMAIL_CC_UDF	This parameter takes the entries in the CC field of an incoming email being processed by EVMail. You may set the value of this parameter to a field with a display type of *text* or *text area*. The contents of the EVMAIL_CC_UDF field are not cumulative, but the issue history will show the successive values from each issue update. The value stored in the field will simply be the CC addresses of the current incoming email message. The email address within the setting named MAILBOX_USER is not used as a recipient.
EVMAIL_FROM_UDF	You may set the value of this parameter to a field with a display type of *text* or *text area*. If an incoming email is processed by EVMail and ExtraView fails to find a valid user who has an email address or alternate email address that matches the incoming message FROM address, Extraview will place the incoming FROM email address in the EVMAIL_FROM_UDF parameter. If ExtraView finds a matching user, it will not populate the new field, and will simply set the Originator field to the identified user account. This action works only when creating new issues - the EVMAIL_FROM_UDF field is not modified when an existing issue is being updated. This setting only works with ExtraView Enterprise versions.

EVMAIL_ID_REGEX	This is a regular expression used to parse the subject line and search for an ExtraView ID value. If an ID is found, EVMail attempts to update a matching issue ID in ExtraView. If no matching ID is found, a new issue is created. If you enter an invalid regular expression, ExtraView will report this information to the application server log.
EVMAIL_REMOVE_CONSECUTIVE_BLANKLINE	Some email client software such as Microsoft Outlook turn each occurence of a newline/linefeed sequence of characters into 2 linefeed characters. This can have the effect when the email is handled via EV Mail that the 2 linefeed characters introduce additional, unwanted space, betweeen paragraphs. This option will alter 2 consecutive linefeed characters into a single linefeed character, thereby preserving the intent of the author or the email.

EVMAIL_SENDER	When set, this option will take the email address of the sender of the incoming email and store it within a field within the issue that is being inserted or updated. The usage is:
	
	`EVMAIL_SENDER = SOME_TEXT_FIELD`
	or
	`EVMAIL_SENDER = SOME_USER_FIELD`
	
	This allows the user to view the field and determine who provided the last update to the issue via email, as opposed to the user who last updated the issue with the user interface of ExtraView.

EVMAIL_SUBJECT	This ExtraView field will be where the subject of the email message is placed. Typically this will be the SHORT_DESCR field
EVMAIL_SUBJECT_LOOKUP	The EVMAIL_SUBJECT_LOOKUP setting works in addition to the EVMAIL_ID_REGEX setting by searching existing records for matching subject line. When no ExtraView ID is detected in the subject line and EVMAIL_SUBJECT_LOOKUP is set to YES, then existing records are searched by the field specified in the EVMAIL_SUBJECT setting and matched against the incoming email subject line. If a record matches, then that record is updated. Otherwise, when no match is found, then a new record is inserted. Note that subject line prefixes such as **Re:** and **Fwd:** are removed from the incoming email subject line for the purposes of performing the search and match. Also, the field specified in the EVMAIL_SUBJECT setting must be configured in Data Dictionary to have their **Filter Criteria** attribute set to *Yes*, so that a query can be executed with that field as a filter. If multiple records match the subject line, then a new record is inserted as opposed to updating multiple existing records.
EVMAIL_SUPPRESS_HEADER	This configuration option allows ther administrator to remove specific lines from the header of the incoming mail, before the mail is inserted into a field within ExtraView. The remaining text is inserted into the destination field defined with EVMAIL_BODY_UDF. The header lines removed begin with the following strings and they must appear within the incoming mail before the occurrence of the EVMAIL_DELIMITER_TEXT contained within the email body: **From:** **To:** **Send:** **Subject:** **Date:** Only header lines that begin with these strings are removed. There must not be blank lines between the header lines and they must all be followed within the mail body by the EVMAIL_DELIMITER_TEXT. If the mail does not contain a the expected value for EVMAIL_DELIMITER_TEXT, then this configuration option is ignored. This implies that the header

information is not removed when creating a new issue within ExtraView, and is only operable when updating issues.

EVMAIL_TO_UDF

You may set the value of this parameter to a field with a display type of *text* or *text area*.

EVMail parses the TO header within the incoming email and extracts the email address(es) in this list. If the EVMail address is in the list, it is removed. This may result in an empty list. If the list is not empty, EVMail creates a semi-colon separated list of email address and uses this as the value within the EVMAIL_TO_UDF field. This processing happens for both emails that create new issues within ExtraView as well as for emails that update issues within ExtraView.

The contents of the EVMAIL_TO_UDF field are not cumulative, but the issue history will show the successive values from each issue update. The value stored in the field will simply be the TO addresses of the current incoming email message.

The email address within the setting named MAILBOX_USER is not used as a recipient.

Great care should be taken in the use of this parameter, as it is possible to generate an endless loop whereby an email generates an email that is processed by EVMail, generating an email notification that is again seen by EVMail and updates the same issue, ad infinitum.

IGNORE_SUBJECT.xx

The optional IGNORE_SUBJECT setting can be used to specify one or more strings to detect within the subject line of the incoming email, to determine whether to ignore and skip the entire email message from being processed. The string text configured in this setting is case-sensitive and be matched exactly anywhere in the subject line of the incoming email message.

Note: For a single text, use simply IGNORE_SUBJECT; for multiple texts, use IGNORE_SUBJECT with a unique postfix for each entry, such as a number.

For example to configure multiple entries to be ignored:

```
IGNORE_SUBJECT.1 = Mail Delivery Failure
IGNORE_SUBJECT.2 = Automatic reply:
IGNORE_SUBJECT.3 = Free offer
```

PARSE_EMAIL
PARSE_FIELD.STATUS
PARSE_FIELD.COMMENTS
PARSE_FIELD.DUE_DATE

This collection of properties allow parsing of incoming mails to ExtraView.

You may configure the parsing of the incoming email message subject line and the body looking for specific tags to specify the field and value to assign to the ExtraView record being created or updated.

The PARSE_EMAIL setting must be configured to one of three values:

- INSERT enables the feature for new records only
- UPDATE enables the feature for updates to existing record
- BOTH enables both inserts and updates

If the setting is commented out or set to any other value, then the feature is not configured correctly and is not used.

The PARSE_FIELD setting specifies the field name to map to the tag expected in the incoming email message, as follows:

```
PARSE_FIELD.<ddname> = <tag>
```

where `<ddname>` is the field name in Data Dictionary and `<tag>` is any text pattern that appears in the subject or body of the incoming email.

Note: Do not use the equals character within `<tag>`.

The value to set the field to must follow the tag in the email message, and must be on the same line as the tag in the email body. For example, to specify value of *John Doe* for the `[Reporter]` tag:

```
[Reporter] John Doe
```

In the subject line, the value must be enclosed in double-quotes, separate from the rest of the subject line text. For example, to specify value of *High* for the `[Severity]` tag:

```
Subject: [Severity] "High" - The roof is leaking
```

Multi-line text or multi-list values must be ended with the same tag name within the email body.

For example, to specify multi-line text for the [Comment] tag (i.e. a log area display type field):

```
[Comment] This is a multi-line
comment that spans
three lines
[Comment]
```

When this setting is used, but the field configured with PARSE_FIELD does not exist on the *Add* or *Edit* screen layout, or the EVMail user does not have write access to the field, or the value specified in the email is not valid, the expected behavior for the field is to not be assigned a value by EVMail. A warning message is written to the ExtraView application log.

If you are writing to a multi-valued list field, then values are additive, i.e. the new values are added to existing values in the list.

If you mistakenly utilize the same tag more than once within the email message, only the first occurrence is processed and the others are ignored.

Examples

```
PARSE_FIELD.TEST_CURRENCY = $$Currency Test$$
PARSE_FIELD.DUE_DATE = [Due by]
PARSE_FIELD.TEST_DATE2 = [Date test]
PARSE_FIELD.CUST_RMA_NUM = "number test"
PARSE_FIELD.TEST_DECIMAL = (Decimal Test)
PARSE_FIELD.CUST_LIST = /Customer Name/
PARSE_FIELD.CUST_LIST_MULTI = #Multi-Customers#
PARSE_FIELD.TEST_POPUP_SINGLE = 'popup'
PARSE_FIELD.TEST_POPUP_MULTI = __multi-popup__
PARSE_FIELD.TEST_HTML_AREA = <HTML>
PARSE_FIELD.TEST_PRINT_TEXT = --PRINT--
PARSE_FIELD.TEST_RADIO_H = hRadio
PARSE_FIELD.TEST_RADIO_V = vRadio
PARSE_FIELD.CUST_SELECT_TABS = {tabs}
PARSE_FIELD.CUST_ADDRESS = |Address|
PARSE_FIELD.CUST_EMAIL = @EMAIL@
PARSE_FIELD.IT_REQUESTOR = *User*
PARSE_FIELD.TEST_USER_MULTI = ++USERS++
```

SMTP_CC	Other email addresses to copy when sending auto-reply or failure messages back to the sender of the email
SMTP_FAIL_MSG_TEMPLATE	The database name of an Ad Hoc email template to use when sending out failure messages. The template supports the use of the special variable, $$BODY$$. If you include $$BODY$$ in the template, the body of the original incoming email will be included.

SMTP_FROM	The email address used when sending auto-reply or failure messages back to the sender of the email
SMTP_NAME	The friendly name used when sending auto-reply or failure messages back to the sender of the email
SMTP_REPLY_TEMPLATE	The database name of an Ad Hoc email template to use when sending out auto-reply messages. The template supports the use of two special variables, $$ID$$ and $$BODY$$. If you include $$ID$$ in the template, then the ID of the newly created issue is retrieved and included in the template email. If you include $$BODY$$ in the template, the body of the original incoming email will be included.

The behavior of the last two parameters controls how HTML-based incoming emails are handled.

1. Accept HTML mail as is, and place it into an HTML area field, trying to preserve the formatting - the assumption is that you are mainly receiving short emails, and you want to attempt to preserve HTML constructions such as tables and links
2. Strip the incoming HTML-formatted mail of HTML tags, and place the contents into a text area or comments field. ExtraView will parse the text, removing as much HTML as possible. This is not a perfect solution, as some tags may not be recognized and any poor HTML may not be rejected.
3. Reject the incoming HTML-only mail and send back to the submitter a rejection notice.

Configuring RUNTIME_ Parameters

- For updates to occur, choose a RUNTIME_USER_ID that occupies a real license as opposed to a GUEST user license. If you pick a GUEST user as the RUNTIME_USER_ID, then issues will only ever be created, never updated
- The RUNTIME_USER_ROLE needs to be a role that has access to the fields and layouts for the Business Area and Project of issues that you want to create and/or update. Again, if you select the GUEST role, issues will only ever be created, and never updated
- RUNTIME_AREA_NAME and RUNTIME_PROJECT_NAME - these parameters determine the Business Area and Project of issues created by EVMail.

Sample EVMail configuration

```
RUNTIME_USER_ID = BSMITH
RUNTIME_AREA_NAME = CUST_ISSUES
RUNTIME_PROJECT_NAME = CUST_ISSUES_DATA
RUNTIME_USER_ROLE = SUPPORT

MAILBOX_TYPE = POP3
MAILBOX_PORT = 110
MAILBOX_SECURE = NO
MAILBOX_SERVER = mail.mycompany.com
MAILBOX_USER = support
MAILBOX_PASSWORD = myPassword

SMTP_FROM = support@mycompany.com
SMTP_NAME = "The Support Team"
SMTP_CC = leeann@mycompany.com
SMTP_REPLY_TEMPLATE = SUCCESS_EMAIL

EVMAIL_ID_REGEX = [ExtraView-(d+)]
EVMAIL_BODY_UDF = DESCRIPTION
EVMAIL_BODY_UPDATE_UDF = CUST_DESC
EVMAIL_SUBJECT = SHORT_DESCR
EVMAIL_CATEGORY = EMAIL
EVMAIL_STATUS = NEW
EVMAIL_ORIGINATOR = DEV
```

Handling Issue Updates from EVMail

When a user sends an email to ExtraView, EVMail examines the incoming email to determine whether this is a reply to an existing issue which should update the existing issue, or whether to generate a new issue within ExtraView. This determination is made by parsing the subject line of the incoming email. If the EVMail utility is able to parse a valid, existing issue ID out of the subject line, then it is assumed that the mail is being used to update an issue; else it is assumed that the mail is meant to generate a new issue.

One consequence of this is that when a user replies to an email notification which was generated by ExtraView automatically, the original notification will be included in the reply, simply adding to the update with information that already exists. To eliminate this,

the value within a behavior setting named EVMAIL_DELIMITER_TEXT is used. This value is embedded with invisible tags within HTML emails from ExtraView to surround the entire contents of the outgoing email. When EVMail is updating the issue it looks for the value of the behavior setting and removes all text between the tags. Thus, only the new comment made by the user is added to the issue.

When issues are updated via EVMail, the **Last Change User** field is set to the value of the RUNTIME_USER_ID, and not to the person who sent the email.

Note that if EVMail is running with a user who only has a GUEST role, updates to issues are not permitted.

Which Issue is Updated Via EVMail?

The logic to locate which issue to update with a received mail is a multi-step process, as follows.

1. First, the EMAIL_SUBJECT_REGEX parameter is used to attempt to identify a unique ExtraView issue ID in its subject line
2. Second, the MESSAGE_ID within the message header is examined and used to try and find a matching ExtraView issue

   This is accomplished if the configuration parameter EVMAIL_MESSAGE_ID points to a TEXTAREA field to store the MESSAGE_ID. This field must be present on both the ADD_PROBLEM and EDIT_PROBLEM layouts and must be writable, although you can hide the field.

3. If more than one matching issue is found, ExtraView next looks at the date of the incoming mail. This logic is configured with the definition of the EVMAIL_RECORD_MATCHING_DATE parameter within the task configuration. The field identified with this parameter must have a display type of DATE or DAY. For example, you can use the inbuilt TIMESTAMP or DATE_CREATED fields
4. DATE_RANGE checking logic is then triggered with the configuration value of EVMAIL_RECORD_MATCHING_DATE_RANGE.

   Supported formats and their meanings are:

* **SYSDATE-2** means between SYSDATE and 2 hours or 2 days earlier.
* **0-2** has the same meaning as **SYSDATE-2**. Zero is interpreted as SYSDATE
* **1-3** has the meaning of between 1 hour earlier than the SYSDATE and 3 hours earlier than the starting point, or a DAY earlier than the SYSDATE and 3 DAYS earlier than the starting point.

   Examples:

   * Both the DATE field and DATE RANGE field are defined:
     * The one with latest DATE value withing the DATE_RANGE is chosen
     * No issue is found if there are no results within the DATE RANGE.
   * The DATE field is defined but the DATE RANGE field is not:
     * The issue with the latest DATE value is chosen
   * Both the DATE and DATE RANGE fields are not defined oronly the DATE RANGE field is defined:
     * No issue is selected

1. ExtraView uses SUBJECT line matching if none of the above steps work, and if EVMAIL_SUBJECT_LOOKUP = YES is defined within the configuration.

EVMail and the ExtraView ALT_ID Configuration

If your ExtraView configuration is configured to use ALT_ID as the issue ID, you will have set the behavior setting named ITEM_ID_DISPLAY to a value of ALT_ID. Once this is set, EVMail will use ALT_ID as the issue ID when matching incoming email updates.

Testing EVMail

EVMail follows all the status change rules and the ALLOW_EDIT_CLOSED behavior setting. These may be based on the user, their role, their current business area and project as configured in your installation.

If you want to bypass all of those constraints for testing purposes, and make sure that tickets are updated regardless of any rules in place, use the ADMIN user account. This account (which should not be used for ordinary purposes) bypasses all status change rules.

It is worth ensuring that your workflow does not put any issue into a status value where EVMail will not be able to update it -- you may get failures, and/or the update will be captured as a new issue within the system as opposed to an update to an existing issue.

Using a Secure Mail Server

This section covers instructions on how to connect ExtraView to a POP3 mail server configured with SSL, i.e. POP3S. It is highly recommended that you configure POP3S rather than POP3.

The instructions here are for a Microsoft Windows environment, but are similar for other operating systems. The basic steps to configure SSL, are to install the SSL certificate on your mail server into the Java keystore, as a trusted root certificate.

- Use a browser to go to the web management interface for your mail server, or to some other URL provided by your email client, to access your rootca certificate
- You should now see a dialog box warning you about the certificate. Click on the **View Certificate** and install the certificate. You can ignore any warning messages
- Now that the server certificate is installed on your computer, your browser will no longer warn you when you visit the same site again. However your Java runtime JRE does not yet know about this certificate's existence until you add the certificate to its keystore. Usually you will use the Java keytool utility to manage certificates. keytool is a command-line utility with numerous arguments that allows you to create and manage keystores that house digital certificates. For the complete documentation of keytool, visit http://java.sun.com/j2se/1.3/docs/tooldocs/win32/keytool.html
- You can list the current certificates contained within the keystore using the keytool -list command. The initial password for the cacerts keystore is *changeit*. For example, you may see the following in a Windows environment:

    C:\ExtraView\java\jre\bin>keytool -list -keystore ..libsecuritycacerts Enter default keystore password: changeit

    You will then see the something like this:

    Keystore type: jks
    Keystore provider: SUN

    Your keystore contains 11 entries:

    engweb, Wed Apr 11 16:22:49 EDT 2001, trustedCertEntry?,
    Certificate fingerprint (MD5): 8C:24:DA:52:7A:4A:16:4B:8E:FB:67:44:C9:D2:E4:16 thawtepersonalfreemailca, Fri Feb 12 15:12:16 EST 1999, trustedCertEntry?,
    Certificate fingerprint (MD5): 1E:74:C3:86:3C:0C:35:C5:3E:C2:7F:EF:3C:AA:3C:D9 thawtepersonalbasicca, Fri Feb 12 15:11:01 EST 1999, trustedCertEntry?,
    Certificate fingerprint (MD5): E6:0B:D2:C9:CA:2D:88:DB:1A:71:0E:4B:78:EB:02:41 verisignclass3ca, Mon Jun 29 13:05:51 EDT 1998, trustedCertEntry?,
    Certificate fingerprint (MD5): 78:2A:02:DF:DB:2E:14:D5:A7:5F:0A:DF:B6:8E:9C:5D thawteserverca, Fri Feb 12 15:14:33 EST 1999, trustedCertEntry?,
    Certificate fingerprint (MD5): C5:70:C4:A2:ED:53:78:0C:C8:10:53:81:64:CB:D0:1D thawtepersonalpremiumca, Fri Feb 12 15:13:21 EST 1999, trustedCertEntry?,
    Certificate fingerprint (MD5): 3A:B2:DE:22:9A:20:93:49:F9:ED:C8:D2:8A:E7:68:0D verisignclass4ca, Mon Jun 29 13:06:57 EDT 1998, trustedCertEntry?,
    Certificate fingerprint (MD5): 1B:D1:AD:17:8B:7F:22:13:24:F5:26:E2:5D:4E:B9:10 verisignclass1ca, Mon Jun 29 13:06:17 EDT 1998, trustedCertEntry?,
    Certificate fingerprint (MD5): 51:86:E8:1F:BC:B1:C3:71:B5:18:10:DB:5F:DC:F6:20 verisignserverca, Mon Jun 29 13:07:34 EDT 1998, trustedCertEntry?,
    Certificate fingerprint (MD5): 74:7B:82:03:43:F0:00:9E:6B:B3:EC:47:BF:85:A5:93 thawtepremiumserverca, Fri Feb 12 15:15:26 EST 1999, trustedCertEntry?,
    Certificate fingerprint (MD5): 06:9F:69:79:16:66:90:02:1B:8C:8C:A2:C3:07:6F:3A verisignclass2ca, Mon Jun 29 13:06:39 EDT 1998, trustedCertEntry?,
    Certificate fingerprint (MD5): EC:40:7D:2B:76:52:67:05:2C:EA:F2:3A:4F:65:F0:D8

- Now you have to add the certificate you previously installed to this keystore. Begin by exporting your CA Root certificate from your browser as a **DER-encoded binary file** and save it as C:root.cer. Within Internet Explorer, you can view the installed certificates under **Tools** --> **Internet Options** --> **Content** --> **Certificates**. Once you open the certificates, locate the one you just installed under **Trusted Root Certification Authorities**. Select and click on **Export**. You can now save it as a **DER encoded binary** under your C: drive
- Use the keytool -import command to import this file into your cacerts keystore. For example:

    C:\ExtraView\java\bin\keytool -alias myprivateroot -keystore ..libsecuritycacerts -file c:root.cer

    Enter default keystore password: changeit

    Owner: CN=Division name, OU=Department, O=Your Company, L=Anytown,

ST=NC, C=US, EmailAddress?=you@company.com
Issuer: CN=Division name, OU=Department, O=Your Company, L=Anytown,
ST=NC, C=US, EmailAddress?=you@company.com
Serial number: 79805d77eecfadb147e84f8cc2a22106
Valid from: Wed Sep 19 14:15:10 EDT 2001 until: Mon Sep 19 14:23:20 EDT 2101
Certificate fingerprints: MD5: B6:30:03:DC:6D:73:57:9B:F4:EE:13:16:C7:68:85:09
SHA1: B5:C3:BB:CA:34:DF:54:85:2A:E9:B2:05:E0:F7:84:1E:6E:E3:E7:68

Trust this certificate? [no]: yes

Certificate was added to keystore

- Run keytool -list again to verify that your private root certificate was added:

C:\ExtraView\java\bin\keytool -list -keystore ..libsecuritycacerts

You will now see a list of all the certificates including the one you just added. This confirms that your private root certificate has been added to the ExtraView Java cacerts keystore as a trusted certificate authority

- Stop and restart your application server to pick up the new certificate.

Configuring EVMail with Microsoft Exchange as a Server

Microsoft Exchange Server 2007 and newer is recommended. Configure these settings to achieve this connection:

MAILBOX_TYPE	Set this to a value of EXCHANGE to use Microsoft Exchange Server
MAILBOX_USER	Set this to the user name of the connection to Exchange
MAILBOX_PASSWORD	Set this to the password of the user with which you are connecting to Exchange
EXCHANGE_URL	This value should be the URL of the Exchange server

EVMail can connect to the older Microsoft Exchange 2003 Server, via OWA (Outlook Web Access). To select Exchange 2003 mailbox in EVMail use the following configuration settings.

MAILBOX_TYPE = OWA
MAILBOX_PORT = 110
MAILBOX_SECURE = YES
MAILBOX_SERVER = owa.exchange.ms
MAILBOX_USER = evmailuser
MAILBOX_PASSWORD = password

#These parameters are used when MAILBOX_TYPE = OWA
# for connecting to Exchange 2003 using OWA
OWA_PATH = Exchange
OWA_DESTINATION = https://owa.exchange.ms/Exchange/evmailuser@yourdomain.com
OWA_ISA = true
OWA_DOMAIN = companydomain
OWA_FBA_PATH = /exchweb/bin/auth/owaauth.dll

Notes:

1. Do not configure these parameters for Microsoft Exchange Server if you are using a POP3 or POP3S server
2. For OWA the mailbox_port is not used, but it is a required parameter; leave its value as 110
3. MAILBOX_SECURE - set to YES if your OWA url is https, set to NO if your OWA url is http
4. MAILBOX_SERVER - your Exchange Server
5. MAILBOX_USER and MAILBOX_PASSWORD - the username and password of your Exchange 2003 mailbox

Specific settings for OWA Exchange 2003 connections only - we provide sample settings to help you identify the equivalent settings in your Exchange Server.

MAILBOX_SERVER = Name of your MS Exchange Server

OWA_PATH - Exchange path for MS Exchange OutLook Web Access

The URL used to access your Outlook Web Access page is made up of the values for MAILBOX_SERVER and the OWA_PATH, e.g.

http(s)://MAILBOX_SERVER/OWA_PATH/ - should take you to your Outlook Web Access page

OWA_DOMAIN - the Exchange/Windows domain for the MAILBOX_USER account

OWA_FTA_PATH - path to Form based authentication - this can be found if you look at the HTML source of the OWA authentication/login page - search for "destination".

OWA_DESTINATION - on the OWA login page, search for the hidden field "destination" OWA_ISA - needed for use with OWA_DESTINATION

How EVMail Handles Errors

When EVMail encounters a problem, it may be either a "transient" error, i.e. one that might fix itself if the operation is tried again at a later time, or it might be a "permanent" error, i.e. one that needs a change in the EVMail configuration in order to be resolved. Examples of permanent errors include:

1. Bad username/password for the MAIL_USER or MAIL_PASSWORD parameters
2. Invalid values for RUNTIME_USER_ID, RUNTIME_AREA_NAME, RUNTIME_PROJECT_NAME, RUNTIME_USER_ROLE

An example of a transient errors would be the inability to connect to the mail server because of a network error. In a test situation, you can create a transient error by setting MAILBOX_PORT or MAILBOX_SERVER to an invalid value temporarily. When EVMail encounters a permanent error, it does the following:

1. Changes the task status to ERROR
2. Sends an email to all users in the ADMIN role indicating that EVMail is in an ERROR state **Fatal Task Configuration Error**
3. Stops the task so that it no longer runs

Once the task is in an ERROR state, the following must be done to repair the error

1. Edit the task and fix the configuration problem that caused the error, e.g. fix the MAILBOX_PORT if this is in error
2. Change the status of the task from START_NOW to STOP_NOW
3. Save the task
4. Wait until the **Task Management Task** has run, and the EVMail task status has changed to STOPPED
5. Edit the EVMail task and change the status from STOP_NOW to START_NOW
6. Save the task
7. Wait until the **Task Management Task** has run, and the EVMail task status has changed to STARTED

When EVMail is running and encounters a transient error, it will do the following:

1. Generate a system_log warning message indicating that it has encountered a transient error
2. Check the values for the new EVMail configuration parameters RUNTIME_WARN_NOTIFICATION_INTERVAL and RUNTIME_WARN_NOTIFICATION_COUNT
3. Check the system log table and see how many warning messages have been created within the past RUNTIME_WARN_NOTIFICATION_INTERVAL minutes. If there are more than RUNTIME_WARN_NOTIFICATION_COUNT of them, it will create a new notification entry in the system log
4. Check the RUNTIME_ERROR_NOTIFICATION_INTERVAL. If this is the first notification entry in the system log for this warning, in the last RUNTIME_ERROR_NOTIFICATION_INTERVAL minutes, send the email out to the ADMIN role users
5. Note that EVMail will continue to run

Example:

- RUNTIME_ERROR_NOTIFICATION_INTERVAL = 15 minutes
- RUNTIME_WARN_NOTIFICATION_INTERVAL = 3 minutes
- RUNTIME_WARN_NOTIFICATION_COUNT = 3
- EVMail is set to run every minute

At 10:00 a.m. EVMail runs, and encounters a transient error as it cannot connect to the POP3 mailbox. A message is placed in the system_log table similar to the following:

<LOG_MESSAGE>EVMail Task had a WARN condition</LOG_MESSAGE>
<TASK_ID>700983</TASK_ID>

```
<TASK_TITLE>EVMail Task (Incoming email)</TASK_TITLE>
</LOG_MESSAGE>
```

ExtraView checks to see how many messages of this type, for this task_id exist in the system log table in the past 3 minutes - there is 1. 1 is less than 3 so ExtraView does not send any notification to the administrators.

At 10:01 a.m. EVMail runs again, and encounters the same error - a message is placed in the system_log table:

```
<LOG_MESSAGE>EVMail Task had a WARN condition</LOG_MESSAGE>
<TASK_ID>700983</TASK_ID>
<TASK_TITLE>EVMail Task (Incoming email)</TASK_TITLE>
</LOG_MESSAGE>
```

ExtraView checks to see how many messages of this type, for this task_id exist in the system log table in the past 3 minutes - there are 2. 2 is less than 3 so ExtraView does not send any notification to the administrators.

At 10:02 a.m. EVMail runs again, and encounters the same error - a message is placed in the system_log table:

```
<LOG_MESSAGE>EVMail Task had a WARN condition</LOG_MESSAGE>
<TASK_ID>700983</TASK_ID>
<TASK_TITLE>EVMail Task (Incoming email)</TASK_TITLE>
</LOG_MESSAGE>
```

ExtraView checks to see how many messages of this type, for this task_id exist in the system log table in the past 3 minutes - there are now 3. 3 is not greater than 3 so ExtraView does not send any notification to the administrators.

At 10:03 a.m. EVMail runs again, and encounters the same error - a message is placed in the system_log table:

```
<LOG_MESSAGE>EVMail Task had a WARN condition</LOG_MESSAGE>
<TASK_ID>700983</TASK_ID>
<TASK_TITLE>EVMail Task (Incoming email)</TASK_TITLE>
</LOG_MESSAGE>
```

ExtraView checks to see how many messages of this type, for this task_id exist in the system log table in the past 3 minutes - there are now 4. 4 is greater than 3, so we add a new message to the system log table:

```
<LOG_MESSAGE>EVMail Task had a transient error that persisted more than 3 times in 10 minutes.
<TASK_ID>700983</TASK_ID>
<TASK_TITLE>EVMail Task (Incoming email)</TASK_TITLE>
</LOG_MESSAGE>
```

ExtraView checks to see how many messages of this type for this task_id, exist in the system log table in the past 10 minutes - we have 1, so we need to send an email to all ADMIN users.

At 10:04 a.m. EVMail runs, and encounters the same error - a message is placed in the system_log table:

```
<LOG_MESSAGE>EVMail Task had a WARN condition</LOG_MESSAGE>
<TASK_ID>700983</TASK_ID>
<TASK_TITLE>EVMail Task (Incoming email)</TASK_TITLE>
</LOG_MESSAGE>
```

ExtraView checks to see how many messages of this type, for this task_id exist in the system log table in the past 3 minutes - there are now 4. 4 is greater than 3, so a new message is added to the system log table:

```
<LOG_MESSAGE>EVMail Task had a transient error that persisted more than 3 times in 10 minutes.
<TASK_ID>700983</TASK_ID>
<TASK_TITLE>EVMail Task (Incoming email)</TASK_TITLE>
</LOG_MESSAGE>
```

ExtraView checks to see how many messages of this type for this task_id, exist in the system log table in the past 10 minutes - there is 1, so ExtraView will send an email to all ADMIN users. Even if the transient error persists ExtraView only sends out warning emails every 10 minutes. If your EVJ.log shows errors similar to:

```
2012-10-30 13:45:54 [ ERROR] EVMT: EVMail Task (Incoming email) Mailbox Authentication failed - check your
MAILBOX_USER and MAILBOX_PASSWORD
2012-10-30 13:45:54 [ ERROR] Exception = javax.mail.AuthenticationFailedException: Command is not valid in this state.
javax.mail.AuthenticationFailedException: Command is not valid in this state.
```

Check to see if the POP3 account is on an Exchange2010 server. If so, the problem is likely due to the fact that the Exchange POP is expecting encrypted credentials. To resolve your email administrators will need to run the following command in an Exchange shell which allows Exchange POP to use plain text credentials:

Set-PopSettings -LoginType PlainTextLogin

Then restart the POP3 service and the problem should be resolved.

Multiple EVMail Tasks

It is possible to configure multiple EVMail tasks, each with their own configuration. If you configure this on a single application server, then all the mail connections must either be secure (using POP3S) or not secure (using POP3). If there is a requirement to have mixed secure and non-secure servers, then you should set up a cluster of application servers, and then different servers may be configured to be secure or non-secure.

Technical Note on Incoming Emails with Attachments

There are three possible ways in which attachments to incoming emails can be handled. The first type is handling attachment *parts* or *non-body* parts of the incoming email. In general, anything that is attached to the incoming email should be uploaded to the ExtraView issue as a file attachment. The second type is when the body of the email message is not copied to a field on the ExtraView issue, but is instead uploaded as a file attachment to the ticket. This happens automatically if the body of the message itself is greater than 512kb in size. The third type is when the incoming email message itself is attached to the ExtraView issue. This will occur, through parameter settings, or encountering a message that cannot be processed for some reason.

1. The part disposition is **Part.ATTACHMENT**

   Incoming emails processed by EVMail that have their own attachments - those files are always attached to the ticket.

   contentType.toLowerCase() = image/gif; name="ali's.gif" (the BODY is empty)

- The part disposition is **Part.INLINE**

   Incoming emails processed by EVMail sometimes have inline attachments. These are often images in the user's signature, inline images within the email body or something similar to "*this email has been checked by Virus Scanner*".

   contentType.toLowerCase() = image/jpeg; name="businesscard.jpg" (the BODY is empty)

   These are only attached to the ticket if the EVMail behavior setting named EVMAIL_INCLUDE_INLINE_ATTACHMENTS is set to YES and a filename and content type are present. Otherwise, if content type is text, the text is appended to the email body. This setting defaults to YES, to not lose anything, but if your preference is not to have what are often just logos attached to issues, you can change this behavior setting to a value of NO.

1. The part disposition is **null**

   If the incoming body of the email message exceeds a size of 512kb, ExtraView will NOT put the message body in the description/comments field, but will attach the email message to the issue and insert a line of text in the issue, indicating "*see attachment <attachment filename> for body of email.*"

   contentType.toLowerCase() = text/plain; charset=utf-8; format=flowed (this has a BODY)
   contentType.toLowerCase() = text/html; charset=utf-8 (the BODY is empty)

2. In all cases, and in addition to any other attachments added to the issue, there is a behavior setting named EVMAIL_ATTACH_EML. When this is set to a value of YES, the source text of the incoming email is also attached to the issue, as an .eml type file
3. If EVMail receives a badly formed email message for some reason, and it cannot be processed, it is attached to the issue that is created, as opposed to throwing it away.

Configuring BatchMail

This task controls the sending of email notifications by ExtraView. In previous versions this was a separate process to ExtraView. From version 6.1, the BatchMail task can be started and stopped within ExtraView, and the log can be examined. It is recommended

that this task be set to START_ON_BOOT, so that it initializes automatically when ExtraView starts. The parameters to set up and utilize BatchMail are as follows:

Parameter	Explanation
ALLOW_ENCRYPTION	This flags whether email encryption is to be set or not. The default is NO
BatchMail.properties	This configuration file is used for standalone BatchMail operation. If you are running BatchMail from within this menu, this file is not needed. It is provided for backwards compatibility with previous versions of ExtraView, which did not run BatchMail as a task within ExtraView. See the Installation Guide for details of the settings within this file.
CHECK_CONNECTION_PAUSE	This defines the number of seconds to wait when contacting your mail server, before assuming there is no connection. The default is 30 seconds
ENCRYPTION_DIR	The directory where the encryption program is stored. This may be an absolute pathname or a relative pathname. Relative paths are relative to the WEB-INF directory on the server
ENCRYPTION_SCRIPT	The script to be used to generate the encrypted result (either file or written to standard out). To test this execute the following from the scripts folder: ./encrypt.sh where is the placeholder for the recipient. It will be retreived by the BatchMail program and should match the name on the keyring. is the placeholder for the filename. Its value will be retreived by the BatchMail program and will be based on the filename generated by ExtraView. For example, this might be: ENCRYPTION_SCRIPT=scripts/encrypt.sh
ENCRYPTED_TO_STANDARD_OUT	If this has a value of "yes" then BatchMail reads the encrypted standard out, otherwise if there is a value of "no" it reads the encrypted file
FORCE_DIRECTORY_RELOAD	This may be YES or NO. It is used principally for NFS mounted directories which are used to store the outgoing emails. Occasionally, the directory entries may be cached and become stale, leading to ExtraView seeing out-of-date information. When this setting is YES, ExtraView forces a re-read of the directory for all operations.
LOG_FILE_MAX_RETAINED	This is the maximum number of "rolled over" log files that are retained on the server before they are deleted. The default is 5
LOG_FILE_MAX_SIZE	This is the maximum size to which the log file is allowed to grow, before it is "rolled over" and a new log file created. The old file is retained on the file system. The default is 20,000 kiloBytes
LOG_FILE_PATH_NAME	This is the path to the BatchMail log file, held within the properties file.
LOG_FILE_PATH_NAME_ABSOLUTE	This is used for standalone installations on Microsoft Windows, or when running in a war file
LOG_LEVEL	The level of logging written to the log file. The standard operational value is 6, and a number up to 12 can be used to produce increasing amounts of debug information in the log, for debugging purposes
MAIL_DIR	This is the absolute or relative path to the mail directory where outgoing emails are stored temporarily, defined in the Configuration.properties file. This must be set. Relative paths are relative to the WEB-INF directory on the server
MAIL_PASSWORD	When the outbound SMTP server requires a username and password to authenticate before sending the email, this parameter provides the password
MAIL_PORT	The MAIL_PORT is typically 25 or 465, depending on whether you are using a secure connection
MAIL_SERVER	The domain name to your mail server. This must be set
MAIL_USER	When the outbound SMTP server requires a username and password to authenticate before sending the email, this parameter provides the username
NOTIFICATION	This setting can be either NOTIFY_NEVER (by itself), or any combination of the following with each setting separated by semicolon(;) : • NOTIFY_DELAY • NOTIFY_FAILURE

- NOTIFY_SUCCESS

For example, the setting might be
NOTIFICATION=NOTIFY_SUCCESS;NOTIFY_FAILURE

PAUSE_AFTER_NUM_OF_EMAIL	This setting pauses the EVMail task after sending the number of mails in the value of the setting. This pauses the EVMail task to stop it commanding all the resources on the server for long periods of time, such as might happen with a mass update of many issues sending out thousands of notifications in a very short period of time. An alternative method is to use the PAUSE_BETWEEN_SEND setting, It is recommended that you use only one of the two settings.
PAUSE_BETWEEN_SEND	This is the number of milliseconds to wait between the sending of each mail notifications, to allow the server to process other tasks. Similar to the PAUSE_AFTER_NUM_OF_EMAIL setting, it reduces the load on the server. It is recommended that you use only one of the two settings. The default is 200 milliseonds
POLL_RATE	The number of seconds to pause between checking for emails to be sent. The default is 5 seconds
RETURN_OPTION	This defines the action to take with returned mail. You can elect to either return the full email or just the headers. The value can be either RETURN_FULL or RETURN_HDRS
SECURE_PORT	This is a dedicated port number for this application. The default is 879. The number must always be less than 1024
SEND_TO_ALL_ONCE_ADDR	This is used in conjunction with SEND_TO_ALL_ONCE setting to determine whether recipients of email notifications will see the other addressees who received the email notification. The default for this optional setting is **BCC**.
SEND_TO_ALL_ONCE	The default for this optional setting is **false**. It may be set to a value of **true**, in which case you can set the SEND_TO_ALL_ONCE_ADDR to a value of **TO** or **BCC**. If set to **TO**, the mail addresses of the recipients will appear in the **TO** header of the outgoing email, as opposed to the **BCC**. It should be understood that this implies that all recipients will receive the email as if they are in the user role of the sender, as it is a single email that is generated, as opposed to an email being generated for each user, customized to their user role.
SERVER_FAILED_PAUSE	This is the number of seconds to wait after a failed connection before retrying. The default is 60 seconds
SIGNING_SCRIPT	The script to be used to generate the signed result (either file or written to standard out). To test this, execute the following from the scripts folder: ./sign.sh The return value will be retrieved by the BatchMail program and will be based on the filename generated by ExtraView. For example, use this: SIGNING_SCRIPT=scripts/sign.sh
TRANSPORT	Use either *smtp* or *smtps* for a secure connection if this is configured on your mail server
XML_LOG_FLAG	The default for this is FALSE. If you set it to true, the log will be created in an XML format

Session Monitor

This is a cleanup task which periodically monitors user sessions and removes expired sessions from the database, thus ensuring that old, expired sessions do not accumulate and cause an unnecessary load on the system. Again, the recommendation is that this is set to START_ON_BOOT and that the default properties of the task are not altered by the administrator.

Quickfind Synchronization

This task automatically indexes newly added text to all issues entered or updated by users in order to speed up keyword searches. Once configured and started, this task will keep the indexes up-to-date as users add or update any text within issues.

This task defines exactly what text will be indexed, and where the indexes will be stored on the file system of your server. The Quickfind indexing mechanism is based upon the Apache Foundation Lucene technology.

Note that files larger than 16MB in size are not indexed. For full end-user information on the use of Quickfind, see the page titled [Keyword Searching](#).

Preparation

Before setting up Quickfind, you should carefully consider where to store the indexe files that are created. The following pointers should be taken into account:

- You should have sufficient space for the index storage. This is dependent upon the size of database and the amount of text that is contained within it. Indexes can vary in size from tens of Megabytes of space to several Gigabytes of space in a very large system
- You must have write permission from the ExtraView WEB-INF folder to this location
- The path to the index files may be relative to WEB-INF, or be an absolute path. It is recommended that you choose an absolute path. The reason is that upon a system upgrade, you will probably rename and replace the old ExtraView installation, and the relative path that contains the indexes will no longer be in the correct location.

First-time Setup

The basic setup and configuration of Quickfind is accomplished on the **Admin** ==> **Task Manager** screen.

- First, *Add a new task*, and select the *Quickfind Synchronization Task*
- Select a Node ID on which to run the task. You must not run the task on more than one node within a clustered system. If you do so, you risk corrupting the indexes created with the utility
- Set the Start Option to *STOP_NOW*. More configuration is required before starting the task
- You can set the frequency with which the task runs with the Poll Interval. It is recommended that this is set between 60 and 600 seconds. The default is 300 seconds
- Do not alter the class name of the task
- If you enter an absolute path to the index files in the **Quickfind Index Location**, and the path already exists, the behavior setting named QUICKFIND_INDEX is updated to reflect that path. If the path does not exist when you create the task, the task is still created, but you will encounter an error, and you are taken to the edit mode, where you can correct the path. Suggestions for index names are:
  - **C:\ExtraView\\<instance name>\\<index name>** on Microsoft Windows platforms, e.g. C:\ExtraView\QuickfindIndex
  - **/usr/local/extraview/<instance name>/<index name>** on Linux and Unix platforms, e.g. /usr/local/ExtraView/QuickfindIndex
  - On an NFS mount, include this fact in the index name by using the convention nfs:pathname
- The behavior setting is updated with the index name
- Click **Add** to create the task
- Notice that the **Task Manager** screen now shows the FULL_TEXT_SYNCHRONIZATION Quickfind Synchronize task
- Click on the *Edit* button of the Quickfind Synchronize task. You will now see a screen that looks similar to the following:

The Quickfind task management utility

- Notice the new configuration options that are now displayed
- Set *Enable Quickfind* to a value of *Yes*
- Setup the path to the location where the index files will be stored on your file system. If you will use the default path as explained below, you can skip this step. All application servers in your installation must have read and write access to the index file location. The index location is also the value stored in the behavior setting named QUICKFIND_INDEX_LOCATION. Further, you must make sure you have sufficient disk space to store the indexes as they grow. The amount of disk space is highly dependent on the quantity of text and number of fields being stored
- The prompt **Allow Quickfind on user defined text fields** determines whether the text within all the user defined fields should be indexed. The recommendation is that you set this to *Yes*
- Observe the dates on which files were last indexed, and the count of files still to be indexed
- If this is new site with no data, then you may now set the value of the *Start Option* to START_NOW, and **Update**. This will start the Quickfind synchronization task and begin to index data as it is entered by users
- If this is a site that has existing data, run the external utility named FullTextIndexSetup. This is described below. The utility creates the Quickfind indexes in one fell swoop, in a fast manner. After completing this indexing operation, you can return to this screen, and set the value of the *Start Option* to START_NOW, and **Update**
- If you need to re-index all or part of the issue data within your database, the bottom section of the screen will provide an aid. You can re-index just the issue data, just the attachments or just the data within document field types, or all of these, by entering a date within each category. To re-index the complete database, put a date in each of the fields, earlier than the date at which data was first entered into the site, and then press **Update**. On a large site it can take a considerable amount of time to index the data, but your users may continue accessing the site in the meantime. Obviously, some searches may not return complete results. You can use this screen to see the progress and how many items remain to be indexed.

Quickfind Properties

The following properties may be set to aid in the running of Quickfind or the investigation of problems:

QUICKFIND_ERROR_NOTIFICATION_INTERVAL	The minimum number of minutes between subsequent error email notifications; this prevents multiple emails to the administrator in the case there is a persistent warning condition

QUICKFIND_WARN_NOTIFICATION_COUNT	The number of warnings in the warning notification interval before a notification email is sent. It is typically acceptable to have a number of warnings before a warning is sent
QUICKFIND_WARN_NOTIFICATION_INTERVAL	The number of minutes in which the warning count specified in QUICKFIND_WARN_NOTIFICATION_COUNT must occur before a notification email is sent
USE_SYSTEM_LOG	If this has a value of YES, then the ExtraView application server log file will be used, as opposed to using the log file specified in either LOG_FILE_PATH_NAME or LOG_FILE_PATH_NAME_ABSOLUTE. If this is set to YES, do not use LOG_FILE_PATH_NAME or LOG_FILE_PATH_NAME_ABSOLUTE
LOG_FILE_PATH_NAME	The relative pathname to the log file. If this is used, set USE_SYSTEM_LOG to a value of NO and do not use LOG_FILE_PATH_NAME_ABSOLUTE
LOG_FILE_PATH_NAME_ABSOLUTE	The absolute pathname to the log file. If this is used, set USE_SYSTEM_LOG to a value of NO and do not use LOG_FILE_PATH_NAME
PSP_LOG	Set this to a value of YES if you need to see the SQL statements generated by the task in the log
XML_LOG_FLAG	Set this to TRUE if you want the log statements to be generated in XML format
DEFAULT_LOG_LEVEL	The default logging level. This is 6
LOG_LEVEL	The log level to use for debugging. In normal operation this should be set to 6
LOG_CHARSET	The character set of the log output. In most cases this should be left as UTF-8
LOG_INCLUDE_THREAD_NAME	This is set to YES to include the threadname of the Quickfind task in the log file

FullTextIndexSetup

If you are enabling Quickfind on an existing installation, you should index all the existing information by using the external program named FullTextIndexSetup. This should be accomplished with the task named Quickfind Synchronization Task (FULL_TEXT_SYNCHRONIZE) turned off, or the ExtraView application stopped. Some upgrades may also require the indexes to be rebuilt, using the same FullTextIndexSetup program.

The FullTextIndexSetup utility is extremely quick on a new database, but may take some time on a very large existing database. It is difficult to predict exactly how long this will take on a large database, as there are many factors such as the processing speed, memory, amount of text, and the amount of attachments that all have an impact. However, it is unlikely that the process will take more than a few hours. Your users can continue working during this period, and the search results will improve as the process continues. Our recommendation is to start the process after the majority of users leave work for the day.

The utility is found in the directory named WEB-INF/data. The syntax to run this utility is:

For Windows platforms -

FullTextIndexSetup.bat JAVA_HOME TOMCAT_HOME EV_BASE

   where JAVA_HOME is the path to your Java, TOMCAT_HOME is the path to your application server and EV_BASE is the path to the ExtraView installation

or for Linux platforms -

sh FullTextIndexSetup.sh evj

where evj is the path to the ExtraView installation. Alter the script to set environment variables set for JAVA_HOME and TOMCAT_HOME in the shell script file if needed.

What is Indexed for Quickfind Keyword Searches

The following are indexed as part of Quickfind's operations.

- Inbuilt Fields
  - the SHORT_DESCR field
  - If the behavior setting named QUICKFIND_INDEX_USERS is YES, the following user fields are indexed:
    - OWNER
    - ASSIGNED_TO
    - LAST_UPDATED_BY_USER
    - ORIGINATOR
    - CONTACT
    - All MODULEs: ASSIGNED_TO field
    - User Defined Field USER type fields
  - All User Defined Fields with a display type of TEXTFIELD
  - All large User Defined Fields with a display type of TEXTAREA, LOGAREA, PRINT_TEXT
  - The User Defined Fields with a display type of HTMLAREA, after removing the HTML tags
- DOCUMENT fields
  - Document description
  - File name
  - Document content according to MIME type and content size:
    - Size must be less than 16MB
    - MIME type must not be any kind of video, audio, or image
    - The MIME type is mapped to an extractor – see MIME type extractors below
- IMAGE fields
  - Image description
  - File name
- Attachments - these are only searched when the **Search Attachment** checkbox is checked
  - Attachment description
  - Attachment content according to MIME type and content size:
    - Size must be less than 16MB
    - MIME type must not be any kind of video, audio, or image
    - MIME type is mapped to an extractor – see MIME type extractors below

MIME Type Extractors

There are two text extractors:

1. PDF Text Extractor: uses iText PdfReader object to tokenize the PDF strings. Document content will NOT be indexed if it is marked as being "encrypted"
2. Office Text Extractor: uses one of the POI extractors appropriate to the type (Word, Excel, Powerpoint, Outlook, Publisher, or Visio). Note: if Excel text extractor fails, it tries to extract text assuming it is a comma-separated or tab-separated text file with an Excel MIME type
3. Other MIME type documents are indexed as text

Notes

As stated above, Quickfind utilizes the Apache Lucene software to provide the indexing mechanism. One support issue is that if your Java Virtual Machine, or your application server (Apache Tomcat or whatever) crashes for any reason, then the indexes may be left in a locked state on the server. The lock files are kept in the directory specified by the **org.apache.lucene.lockdir** system property if it is set, or by default in the directory specified by the **java.io.tmpdir** system property (on Unix boxes this is usually **/var/tmp** or **/tmp**). If for some reason **java.io.tmpdir** is not set, then the directory path you specified to create your index is used. Lock files have names that start with *lucene-* followed by an MD5 hash of the index directory path. If you are certain that a lock file is not in use, you can delete it manually.

Report Schedule Task

When ExtraView is first installed this task is not set to start and no reports may be scheduled for delivery at specific times by users. In order to allow reports to be scheduled for email delivery by users, the **Report Schedule** task must be configured. The **BatchMail** task must also be configured as this is used to deliver the mail once the report is generated by the schedule task.

Note that the default polling interval for the task is 120 seconds. If you are unlikely to have users schedule many reports, and it is not critical for recipients to receive these reports on time, consider making this interval longer.

Task manager for scheduled reports

The Report Schedule task should be present in a new installation, but will be in a STOPPED status. Set the START_NOW start option to start the task. It will be started automatically in the future when you restart the application server.

Add / Update Task

The Add / Update task is responsible for the following actions:

- It actions the NOWAIT option with Business Rules. The option works with the ADD, COPY and UPDATE action rules. If this task is not configured, the updates from ADD, COPY and UPDATE rules are queued, but never applied.
- The task handles issue import from the administrative File Import Utility. This allows the import of issues to happen in the background, thereby speeding the utility. Generally, Add / Update tasks that are configured to run on nodes other than the node where the file import originated can participate in the parallelism of inserts and updates, thereby speeding the overall process. However, if there are images, documents or attachments that are part of the imported issue, then the insert/update must be completed by the originating node. There is nothing for the user to do to ensure this, but the parallelism will be affected because those update/inserts are dedicated to a single node.

The task is configured on the following screen:

Task manager for adding / updating tasks

The task should be run on a single node a clustered environment. A suitable poll time is 30 seconds as the task takes little resource on the server. The options within the properties manage the log file options as follows:

Property	Function
LOG_FILE_PATH_NAME	Specifies the path name to the log file, relative to the WEB-INF directory
USE_SYSTEM_LOG	This is an alternative to using a log file solely for the Add / Update log, and will interleave the log messages with the standard ExtraView application server log file
LOG_FILE_PATH_NAME_ABSOLUTE	This is used as an alternative to the previous two options, and is used for stand-alone installation on the Microsoft Windows platform, or when running ExtraView from within a WAR file deployment
PSP_LOG	This option can be used to debug the feature by writing all the SQL statements executed to the log file
XML_LOG_FLAG	Outputs the log file in XML as opposed to text, if the option is set to TRUE
LOG_FILE_MAX_SIZE	The size of the log file in bytes. Once this size is reached, the log file is archived and a new file started
DEFAULT_LOG_LEVEL	This is the default level for log messages. This value should be 6
LOG_CHARSET	The character set for messages sent to the log file. This should be UTF-8
LOG_INCLUDE_THREAD_NAME	Includes or excludes the thread name in the log output
LOG_LEVEL	The level of log messages, from 1 to 12. The default level is 6. Increasing the number increases the number of log messages

Extraview - ExtraView Integration Task

The ExtraView - ExtraView peer daemon synchronizes items and specific metadata between two ExtraView instances, with specific mappings and filters configured in a configuration file.

This configuration file is part of an ExtraView task.

The synchronization supports all field types stored in the database with item data, including file attachments. Item relationships are also synchronized. The daemon synchronizes specific metadata objects such as list values and allowed values combinations. However, metadata such as field definitions, layouts, rules and other metadata is not synchronized – these objects are the domain of the XML export / import feature.

The feature is fully described in the Extraview - ExtraView Instance Integration section of these guides.

Add the task, then use the above link to configure the integration of two or more ExtraView instances.

LDAP Background Task

This task should only be configured when you are connecting your installation to an LDAP server. It is optional to run this task. When it is running, it will synchronize the LDAP server data that is required within ExtraView, on a periodic basis. This avoids ExtraView reaching out to the LDAP server for user information on each occasion it is required, and uses the information previously read into ExtraView. Simply adding, or editing may often require reading many users' details, and this task makes this operation much more efficient. This task should be configured in preference to the *LDAP Synchronization Task*. You should not configure both tasks.

Without this task being started, ExtraView makes a call to the LDAP server for each and every requirement for any user's information that is required. This ensures that ExtraView always receives up-to-date information. Unless the LDAP server is configured to offer high speed delivery of all responses, this can lead to minor delays in the processing within ExtraView and a resultant slow response to end users.

However, user data within an LDAP server is typically "slow-moving", and changes infrequently. This task will refresh the data for users whose information is stored within ExtraView on a timed basis. There is a possibility that the user's data may have changed in

the LDAP server, but as long as the poll interval is kept to a modest time, the chances are that the cached data within ExtraView will be correct.

This task should only be run on a single node within a clustered environment.

LDAP Synchronization Task

This task should only be configured when you are connecting your installation to an LDAP server. It is optional to run this task. When it is running, it will synchronize the all LDAP server data on a periodic basis that is defined in the task configuration. This avoids ExtraView reaching out to the LDAP server for user information as it is required, and uses the information previously read into ExtraView. Unless you require the synchronization of all the data in your LDAP server, you should configure the *LDAP Background Task* as opposed to this task. You should not configure both this task and the *LDAP Background Task*.

Without this task, ExtraView makes a call to the LDAP server for each and every requirement for any user's information that is required. This ensures that ExtraView always receives up-to-date information. Unless the LDAP server is configured to offer high speed delivery of all responses, this can lead to minor delays in the processing within ExtraView and a resultant slow response to end users.

However, user data within an LDAP server is typically "slow-moving", and changes infrequently. This task will refresh the data for users whose information is stored within ExtraView on a timed basis. There is a possibility that the user's data may have changed in the LDAP server, but as long as the poll interval is kept to a modest time, the chances are that the cached data within ExtraView will be correct.

The *LDAP Background Task* is typically more efficient and should normally be used in preference to this task.

This task should only be run on a single node within a clustered environment. The task configuration looks similar to this:

The purpose of the properties in the configuration are as follows:

Property	Purpose
DEFAULT_LOG_LEVEL	The default is 6 for normal operations, but you may set this to 8 or 12 for an increasing number of log messages. Only use 8 or 12 when debugging

DISABLED_USER_LDAP_SEARCH_FILTER	This property specifies a search filter that selects users that have been disabled or removed in the LDAP server. This property overrides the behavior setting named LDAP_SEARCH_FILTER
EARLIEST_SYNCHRONIZATION_TIMESTAMP	This setting specifies a timestamp that will be used, instead of the timestamp taken from the database table named TABLE_TIMESTAMP. This table tracks the synchronization tasks' run times. This timestamp is only used if, and only if, the timestamp in the table is earlier than the specified timestamp. The timestamp within the property must be in the format: YMMddHHmmss'Z
ENABLED_USER_LDAP_SEARCH_FILTER	This property specifies a search filter that selects only users that are enabled in the LDAP server. This overrides the behavior setting named LDAP_SEARCH_FILTER
FREQUENCY_INTERVAL	Once the task runs, this is the number of minutes before the task will run again. In the example of the previous property, you could set this to 1440, in order to run the task once each day at 3:00 AM. 1440 is the default value for this property, if it is not set

GT_TIME_ZONE_FORMAT	This defaults to a value of Z. It may have one of the following enumerated values:
	
	• DIFF_HOUR
	  ==>g-differential format, using hour only
	• DIFF_HOUR_MINUTE
	  ==>g-differential format, using hour and minute
	• Z
	  ==>g-time-zone (Zulu) format

GT_FRACTION_DELIMITER	This defaults to a value of DOT. It may have one of these values: • DOT • COMMA
GT_FRACTIONAL_DIGITS	This setting is a positive integer that indicates the number of digits in the fractional part of the generalized time. The default value is 3
GENERALIZED_TIME_FORMAT	This setting has a default value of YEAR_MONTH_DAY_HOUR_MIN_FRACTION. The complete set of values that may be set are: • YEAR_MONTH_DAY_HOUR • YEAR_MONTH_DAY_HOUR_FRACTION • YEAR_MONTH_DAY_HOUR_MIN • YEAR_MONTH_DAY_HOUR_MIN_FRACTION • YEAR_MONTH_DAY_HOUR_MIN_SEC • YEAR_MONTH_DAY_HOUR_MIN_SEC_FRACTION
LAST_MODIFIED_FIELD_NAME	When synchronizing with the LDAP server, you use this property to identify the field within the LDAP that can be used to ensure you only download data that has changed since the last occasion on which the synchronization task ran. For example, if you have 50,000 names in your LDAP, and only a handful change each day, then you only want to retrieve and synchronize the changed names. Specification of this property generates an LDAP filter similar to: <att name> >= <last synchronization run time> where <last synchronization run time> is the Generalized Time expression for the date/time (in GMT time zone) of the last synchronization completed by this task

LOG_CHARSET	The character set of the log file. Normally this is UTF-8
LOG_FILE_MAX_SIZE	The maximum size of the log file in kB, before it is archived and a new log file started
LOG_FILE_PATH_NAME	The path and name to the log file for the task. This path name is relative to the WEB-INF folder
LOG_FILE_PATH_NAME_ABSOLUTE	This provides an absolute path name to the log file as an alternative to LOG_FILE_PATH_NAME. This is useful if you are running on Microsoft Windows, or running ExtraView within a WAR file
LOG_INCLUDE_THREAD_NAME	When set to YES, the thread name of the task running on the server is included in the task log
PSP_LOG	This is useful for debugging purposes only and should be commented out, or set to NO. If it is set to YES, then ExtraView logs all the SQL commands generated by the task
RUNTIME_ERROR_NOTIFICATION_INTERVAL	The number of minutes between sending emails when an error is detected and the administrator is notified
RUNTIME_WARN_NOTIFICATION_COUNT	The minimum number of warning occurrences in the notification interval time period before an administrator notification email is sent
RUNTIME_WARN_NOTIFICATION_INTERVAL	The number of minutes during which warnings are counted before a notification email is sent to the administrator
START_TIME_OF_DAY	The time of day to start the task running. This is useful if you want to run the task at a specific time of day, for example at 3:00 AM each day. Use this in conjunction with the FREQUENCY_INTERVAL. This time is entered as GMT
USE_SYSTEM_LOG	If you set this to a value of Y, then the task will use the same log file as the ExtraView application server, and not use the log file defined by LOG_FILE_PATH_NAME
XML_LOG_FLAG	Normally set to FALSE, but if set to TRUE then the log file will be formatted using XML

ExtraView System Logs

If ExtraView Corporation is hosting your installation, you do not have direct access to the file system of the server to configure, alter or use this feature without contacting ExtraView support. There are four primary logs used within ExtraView to record information, activities and errors.

- **Sign On Log**. This records all sign on and sign off activities, and any unauthorized access attempts. Please consult the User Administration section of this guide for information on this log and how to use it.
- **System Log**. This records all metadata transactions. Please consult the System Controls section of this guide for information on this log and how to use it.
- **Application Server Log**. This records each entry and departure of each access to the ExtraView servlet. This log maintains a record of who performed what accesses, and how long each access took to perform.
- **BatchMail Log**. This records the flow of email notifications from ExtraView.

Application Server Log

The application server log is not resident within ExtraView. The administrator normally accesses this log from the file system of the server. As an alternative, the administrator may view the log from the System Log feature. Directly accessing the log from the file system allows the log to be viewed with an editor. Viewing the application server log from the System log screen only allows the entire log to be viewed.

The path to the log file is specified within the Configuration.properties file for ExtraView. This file is typically, and by default, stored within the ExtraView WEB-INF directory. The path is defined within the within the entry named LOG_FILE_PATH_NAME. For

example, in an Apache Tomcat installation of ExtraView with a home directory of /evj, the path to the log file may be similar to /evj/tomcat/webapps/golden411/WEB-INF/logs.

Format of the Application Server Log File

A typical section of the log file in plain text may look like this:

```
2003-07-17 17:45:46 [  info   ] ExtraView., 6544, Thread-31, >>>Entering
service, security.LoginDisplay.getMenu, sc, 0, tmem, 126, fmem, 75, , , , , uid, SHOYDIC, nid, http://avalon.extr
aview.net/med ON WS_A:
2003-07-17 17:45:46 [  info   ] ExtraView., 6544, Thread-31, >>>Leaving
service, security.LoginDisplay.getMenu
, sc, 0, tmem, 126, fmem, 75, cc, 22, time, 20, uid, SHOYDIC, nid, http://avalon.extraview.net/med ON WS_A:
2003-07-17 17:45:46 [  info   ] ExtraView., 6544, Thread-21, >>>Entering
service, security.LoginDisplay.getMain, sc, 0, tmem, 126, fmem, 75, , , , , uid, SHOYDIC, nid, http://avalon.extr
aview.net/med ON WS_A:
2003-07-17 17:45:46 [  info   ] ExtraView., 6544, Thread-21, >>>Leaving
service, security.LoginDisplay.getMain
, sc, 0, tmem, 126, fmem, 75, cc, 22, time, 10, uid, SHOYDIC, nid, http://avalon.extraview.net/med ON WS_A:
2003-07-17 17:45:46 [  info   ] ExtraView., 6544, Thread-31, >>>Entering
service, Display.doEditDisplay, sc, 0, tmem, 126, fmem, 75, , , , , uid, SHOYDIC, nid, http://avalon.extraview.ne
t/med ON WS_A:
2003-07-17 17:45:46 [  info   ] ExtraView., 6547, Thread-29, >>>Leaving service.Display.evSignon
action , sc, 1, tmem, 126, fmem, 72, cc, 24, time, 54, uid, SHOYDIC, nid, http://avalon.extraview.net/med ON
WS_A:
2003-07-17 17:45:47 [  info   ] ExtraView., 6547, Thread-15, >>>Entering
service, security.LoginDisplay.getMenu, sc, 1, tmem, 126, fmem, 82, , , , , uid, SHOYDIC, nid, http://avalon.extr
aview.net/med ON WS_A:
2003-07-17 17:45:47 [  info   ] ExtraView., 6547, Thread-29, >>>Entering
service, security.LoginDisplay.getMain, sc, 2, tmem, 126, fmem, 82, , , , , uid, SHOYDIC, nid, http://avalon.extr
aview.net/med ON WS_A:
2003-07-17 17:45:47 [  info   ] ExtraView., 6547, Thread-29, >>>Leaving
service, security.LoginDisplay.getMain
, sc, 2, tmem, 126, fmem, 82, cc, 24, time, 10, uid, SHOYDIC, nid, http://avalon.extraview.net/med ON WS_A:
2003-07-17 17:45:47 [  info   ] ExtraView., 6547, Thread-15, >>>Leaving
service, security.LoginDisplay.getMenu
, sc, 1, tmem, 126, fmem, 82, cc, 24, time, 27, uid, SHOYDIC, nid, http://avalon.extraview.net/med ON WS_A:
2003-07-17 17:45:47 [  info   ] ExtraView., 6547, Thread-15, >>>Entering
service, Display.doEditDisplay, sc, 1, tmem, 126, fmem, 76, , , , , uid, SHOYDIC, nid, http://avalon.extraview.ne
t/med ON WS_A:
2003-07-17 17:45:48 [  info   ] ExtraView., 6545, Thread-31, >>>Leaving service.Display.doEditDisplay
, sc, 0, tmem, 126, fmem, 74, cc, 25, time, 1842, uid, SHOYDIC, nid, http://avalon.extraview.net/med ON WS_A:
2003-07-17 17:45:48 [  info   ] ExtraView., 6548, Thread-15, >>>Leaving service.Display.doEditDisplay
, sc, 1, tmem, 126, fmem, 82, cc, 25, time, 1322, uid, SHOYDIC, nid, http://avalon.extraview.net/med ON WS_A:
2003-07-17 17:45:48 [  info   ] ExtraView., 6544, Thread-21, >>>Entering
service, security.LoginDisplay.doMenuLoadDone, sc, 0, tmem, 126, fmem, 81, , , , , uid, SHOYDIC, nid, http://aval
on.extraview.net/med ON WS_A:
2003-07-17 17:45:48 [  info   ] ExtraView., 6544, Thread-21, >>>Leaving
service, security.LoginDisplay.doMenuLoadDone
, sc, 0, tmem, 126, fmem, 81, cc, 25, time, 10, uid, SHOYDIC, nid, http://avalon.extraview.net/med ON WS_A:
2003-07-17 17:45:49 [  info   ] ExtraView., 6547, Thread-29, >>>Entering
service, security.LoginDisplay.doMenuLoadDone, sc, 0, tmem, 126, fmem, 81, , , , , uid, SHOYDIC, nid, http://aval
on.extraview.net/med ON WS_A:
2003-07-17 17:45:49 [  info   ] ExtraView., 6547, Thread-29, >>>Leaving
service, security.LoginDisplay.doMenuLoadDone
, sc, 0, tmem, 126, fmem, 81, cc, 25, time, 10, uid, SHOYDIC, nid, http://avalon.extraview.net/med ON WS_A:
2003-07-17 17:46:08 [  info   ] ExtraView., 6537, Thread-24, >>>Entering
service, search.SearchDisplay.doDisplay, sc, 0, tmem, 126, fmem, 81, , , , , uid, SHOYDIC, nid, http://avalon.ext
raview.net/med ON WS_A:
2003-07-17 17:46:08 [  info   ] ExtraView., 6537, Thread-24, >>>Leaving
service.search.SearchDisplay.doDisplay
, sc, 0, tmem, 126, fmem, 77, cc, 25, time, 400, uid, SHOYDIC, nid, http://avalon.extraview.net/med ON WS_A:
```

- Each entry begins with a timestamp
- Each ">>> Entering Service" message has a corresponding ">>>Leaving Service" log entry. This allows a trace of each request to ExtraView, from the moment the request is made, to the moment the request has been serviced, and ExtraView has completed its task.
- The ">>> Leaving service" entries show the time taken on the server to execute the request.
- The file is comma delimited. If you wish to analyze the log file, sorting and extracting the information, it is simple to import the log file into a tool such as Microsoft Excel.

The format of the file is: timestamp, log entry type, servlet name, session ID, thread #, Entering / Leaving, class and method entered, service count, service count value, total memory available, total memory available value, [free memory available, free memory available value, cache count, cache count value, time, time value], user ID, user ID value, network ID, network ID value

Parameter	Explanation
timestamp	The date and time of the entry
log entry type	Normal entries will be shown as INFO. You may also see entries such as ERROR and WARNING
Servlet name	Should always be ExtraView
session ID	A unique identifier for the session
thread #	The thread number that this task is using for execution
Entering / Leaving	Shows whether the task is beginning or ending
class and method entered	This is the class path and the method within the class that is being executed
Service count *sc*	The number of processes waiting for service at the instant this request is received. The lower this number, the higher the performance of the system. For short periods of time, this number may exceed 5, but this indicates that users are waiting upon machine resources to execute their tasks. Setting up multiple application servers that each take some of the load is a good solution if this number is consistently high
total memory available *tmem*	The total memory allocated to this application server. Usually set by the administrator within the start up script for the application server
free memory available *fmem*	The amount of unused memory. If this drops to a very low number, out of memory errors may occur. If this happens, allocate more total memory to this application server
cache count *cc*	The total number of user sessions in cache at the time this request is executed.
time	The time, in milliseconds, that the request took to execute
"morgue" count *mc*	The number of sessions currently in the "morgue". ExtraView allocates sessions to each user to perform virtually every task such as add or update issues, or to run reports. The server stores information about these sessions and retains the information for as long as is needed. Given browser windows may be closed unexpectedly by users, or may be left open for an indeterminate amount of time, the session information is held on the server until the user signs off or their session expires. So that the sign off does not take excessive time, the sessions are not killed at that time, but are flagged as being in the "morgue". The ExtraView background task named *session monitor* comes along at a later time and permanently removes the sessions in the morgue, thus killing them permanently.
user ID *uid*	The user ID of the person who made the request
network ID *nid*	The network ID and node name within an application server cluster that made the request

Errors in the Application Server Log

Errors in the log are generated by ExtraView when it encounters an unexpected event or a programming error. Not all errors are fatal; indeed, it is possible to see an error in the log that has absolutely no consequences to the user of ExtraView who caused the error. However, ExtraView endeavors to minimize all programming errors and exceptions in the log. An exception in the log may look like the following error. This error occurred due to a programming error in a pre-release copy of ExtraView.

```
2003-07-07 12:22:07 [   info   ] ExtraView.,5388,Thread-15,>>>Entering
service,admin.LayoutDisplay.doEdit,sc,0,tmem,126,fmem,106,,,,,uid,ADMIN,nid,http://avalon.extravie
w.net/med ON WS_A:
2003-07-07 12:22:09 [  ERROR] PSP: Invocation Target is:java.sql.SQLException: ORA-00001: unique
constraint (MED.PK_TITLE_MAP) violated
2003-07-07 12:22:09 [  ERROR] PSP: SQL was:  insert into title_map (title_map_key, language,
region, variant, text , date_created, last_date_updated, last_updated_by_user, created_by_user)
values (?, ?, ?, ?, ?,  SYSDATE ,  SYSDATE , ?, ?)
2003-07-07 12:22:09 [  ERROR] PSP: Bind Values are: '6436','ja','JP',' ','null','ADMIN','ADMIN'
2003-07-07 12:22:09 [  ERROR] PSP: SQL State is:23000
2003-07-07 12:22:09 [  ERROR] Exception = java.sql.SQLException: ORA-00001: unique constraint
(MED.PK_TITLE_MAP) violated
java.sql.SQLException: ORA-00001: unique constraint (MED.PK_TITLE_MAP) violated

        at oracle.jdbc.dbaccess.DBError.throwSqlException(DBError.java:180)
        at oracle.jdbc.ttc7.TTIoer.processError(TTIoer.java:208)
        at oracle.jdbc.ttc7.Oall7.receive(Oall7.java:543)
        at oracle.jdbc.ttc7.TTC7Protocol.doOall7(TTC7Protocol.java:1451)
        at oracle.jdbc.ttc7.TTC7Protocol.parseExecuteFetch(TTC7Protocol.java:862)
        at oracle.jdbc.driver.OracleStatement.executeNonQuery(OracleStatement.java:1839)
        at oracle.jdbc.driver.OracleStatement.doExecuteOther(OracleStatement.java:1764)
        at oracle.jdbc.driver.OracleStatement.doExecuteWithTimeout(OracleStatement.java:2354)
        at oracle.jdbc.driver.OraclePreparedStatement.executeUpdate
                (OraclePreparedStatement.java:421)
        at sun.reflect.GeneratedMethodAccessor13.invoke(Unknown Source)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
        at java.lang.reflect.Method.invoke(Method.java:324)
        at com.extraview.util.PreparedStatementProxy.invoke(PreparedStatementProxy.java:91)
        at oracle.jdbc.driver.$Proxy1.executeUpdate(Unknown Source)
        at com.extraview.util.Message.insertOneTitle(Message.java:611)
        at com.extraview.util.Message.insertTitle(Message.java:580)
        at com.extraview.applogic.layout.LayoutElementAttribute.executeTransaction
                (LayoutElementAttribute.java:244)
        at com.extraview.applogic.layout.LayoutElement.insertAttributes(LayoutElement.java:495)
        at com.extraview.applogic.layout.LayoutElement.executeTransaction(LayoutElement.java:344)
        at com.extraview.applogic.layout.Layout.insertElements(Layout.java:1388)
        at com.extraview.applogic.layout.Layout.executeTransaction(Layout.java:1131)
        at com.extraview.applogic.fltool.LayoutDbMaster.executeSave(LayoutDbMaster.java:390)
        at com.extraview.applogic.fltool.LayoutDbMaster.saveLayout(LayoutDbMaster.java:303)
        at com.extraview.presentation.fltool.FlexibleLayout.service(FlexibleLayout.java:494)
        at com.extraview.presentation.admin.LayoutDisplay.doEdit(LayoutDisplay.java:362)
        at sun.reflect.GeneratedMethodAccessor24.invoke(Unknown Source)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
        at java.lang.reflect.Method.invoke(Method.java:324)
        at ExtraView.service(ExtraView.java:250)
        at javax.servlet.http.HttpServlet.service(HttpServlet.java)
        at org.apache.tomcat.facade.ServletHandler.doService(Unknown Source)
        at org.apache.tomcat.core.Handler.invoke(Unknown Source)
        at org.apache.tomcat.core.Handler.service(Unknown Source)
        at org.apache.tomcat.facade.ServletHandler.service(Unknown Source)
        at org.apache.tomcat.core.ContextManager.internalService(Unknown Source)
        at org.apache.tomcat.core.ContextManager.service(Unknown Source)
        at org.apache.tomcat.modules.server.Ajp13Interceptor.processConnection(Unknown Source)
        at org.apache.tomcat.util.net.TcpWorkerThread.runIt(Unknown Source)
        at org.apache.tomcat.util.threads.ThreadPool$ControlRunnable.run(Unknown Source)
        at java.lang.Thread.run(Thread.java:536)

2003-07-07 12:22:10 [   info   ] ExtraView.,5388,Thread-15,>>>Leaving
service,admin.LayoutDisplay.doEdit
,sc,0,tmem,126,fmem,109,cc,1,time,2630,uid,ADMIN,nid,http://avalon.extraview.net/med ON WS_A:
```

When you encounter an error condition in ExtraView, the ExtraView support team may ask you to examine the application server log, and to send this to them. We can determine from an error entry where it occurred, and typically use this information to debug and correct programming errors.

Debugging ExtraView Problems with the Application Server Log

Operational problems may occur for a variety of reasons, from a programming bug in ExtraView, to a database problem, to a network problem, to a hardware problem. The list of potential problem areas with Internet technology is vast. Your best friend in debugging problems is the application server log, although other logs and monitoring tools may be useful, even essential, in pinning down the source of a problem.

Significant problems can be understood by correctly interpreting the log.

As part of debugging problems you may want to change Configuration.properties values such as DEBUG_LEVEL and PSP_LOG. There is a URL that allows you to make an API call to ExtraView to change the values of the parameters in the CONFIGURATION.PROPERTIES file, without needing to restart the application server. This is of the form:

```
user_id=username
&password=password
&statevar= SET_CONFIG_VALUE
&name=parameter_name
&value=YES
```

An example may be:

http://www.myserver.com/evj/ExtraView/ev_api.action?user_id= bsmith&password=bill&statevar=SET_CONFIG_VALUE&name=PSP_LOG& value=YES

BatchMail Log

BatchMail is a separate Java program that is started by ExtraView when it starts, or when your server is booted. This contains the code that processes email notifications when they are output from ExtraView. ExtraView does not send email notifications directly to your SMTP mail server. This is because if the server goes down for any reason, ExtraView may end up waiting (and waiting) for a response.

Instead, ExtraView writes each email notification sent to a temporary directory. The path to the mail directory where the BatchMail program resides is specified within the Configuration.properties file for ExtraView. This file is stored within the ExtraView WEB-INF directory. The path is defined within the within the entry named MAILBOX. For example, in an Apache Tomcat installation of ExtraView with a home directory of /evj, the path to the log file may be similar to /evj/tomcat/webapps/golden411/WEB-INF/BatchMail/logs.

An excerpt from a BatchMail log file follows.

```
2003-07-18 15:47:52 [  ALERT] mail.send() executed ExtraView Notification [10000]:  Unassigned-
Newly reported issue concerning XY module
  to--->ed@baumanns.org in 0 seconds
2003-07-18 15:51:18 [  ALERT] mail.send() executed ExtraView Notification [10001]:  Unassigned-
bug created by sweng1, assigned to qaeng1
  to--->ed@baumanns.org in 0 seconds
2003-07-18 15:56:15 [  ALERT] mail.send() executed ExtraView Notification [10001]:  Unassigned-
bug created by sweng1, assigned to qaeng1
  to--->ed@baumanns.org in 0 seconds
2003-07-18 18:06:07 [  ALERT] mail.send() executed Open[19397]-[P 3]-[Admin Admin] overriding
value for title
  to--->leeann@extraview.com in 0 seconds
2003-07-18 18:06:09 [  ERROR] Error Occured ... javax.mail.SendFailedException: Sending failed;
  nested exception is:
        javax.mail.SendFailedException: Invalid Addresses;
  nested exception is:
        javax.mail.SendFailedException: 550 unknown user

2003-07-18 18:06:09 [  ERROR] Error Occured processing file 19397HtmlQA1058576763923.html for
dpuiia@extraview.com
2003-07-18 18:06:09 [  ERROR] Renaming
/usr/local/extraview/BatchMail/mailbox/19397HtmlQA1058576763923.html to
:/usr/local/extraview/BatchMail/mailbox/__19397HtmlQA1058576763923.html
2003-07-18 18:07:15 [  ALERT] mail.send() executed Open[19398]-[P 3]-[Admin Admin] overriding
value for title
  to--->lpultz@extraview.com in 0 seconds
2003-07-18 18:07:15 [  ALERT] mail.send() executed Open[19398]-[P 3]-[Admin Admin] overriding
value for title
  to--->robbie@extraview.com in 0 seconds
2003-07-18 18:07:15 [  ALERT] mail.send() executed Open[19398]-[P 3]-[Admin Admin] overriding
value for title
  to--->support@extraview.com in 0 seconds
```

Errors in the BatchMail Log

In the case of an error, the log will show the specific error. In the above case, the error shown is an unknown user. When an error is encountered, the email notification file is renamed and remains on the server. The file is renamed with the characters "__" placed in front of the original mail.

Site Configuration Menu

The Site Configuration menu contains the majority of functions used to configure and set up your installation.

Design Center

The Design Center is the principal area where all screen forms and built in user reports, along with the formatting and logic behind them, is configured. The screen forms and built-in reports are designed using **layouts**. You may also create and design your own layouts, embedding them within the built-in layouts. In this way you can build reuseable layouts that can be embedded within many other layouts.

The design center is where you bring together fields, embedded layouts, along with attributes to build the finished layouts of the *Add Issue*, *Edit Issue*, *Query Filters* and other screen layouts. In addition, built in reports such as the *Quicklist* and the *Detailed Report* also use layouts defined within the Design Center. Layouts can be defined for different user roles within your system, offering a tremendous amount of flexibility. See the section on Inheritance for additional details. Each layout is built on top of a **Layout Type**. You may create and define fields within the Design Center, or you may use the Data Dictionary for this purpose.

Layouts work in conjunction with security permissions for each field. Therefore, simply placing a field on a screen does not automatically give all users the ability to read or write to the field. You can alter the permissions to each field within the Design Center, or you may use the Security Permissions option to define which fields are visible and updateable to each group of users. The security privilege for the field overrides the fact that a field may be placed on a screen or report.

A layout may be embedded within another layout. Indeed, you may specify alternate layouts that appear, dependent upon the value of a specific field. For example, you may have a category field that has the values of Software, Hardware and Documentation. Depending on which value is chosen, an embedded layout can be displayed that contains the fields pertinent to gathering the information needed about each of these categories. These embedded layouts may only be embedded within an *Add* or *Edit Issue* layout.

Fields within each layout may have one or more attributes defined. These layout cell attributes affect the display of the field or the way in which it is processed. For example, a layout cell attribute may provide the field with an alternate title, just for the one layout, or the attribute may define that the field is only visible if another field is of a specific value. There are two special type of layouts which may only be embedded within other layouts. These allow the definition of **Related Issue Layouts** and **Repeating Row Layouts.**

Inheritance

First, please read the Key Concepts section on Business Areas to understand the terminology used on this page. The principle is that a specific business area or project will use a layout that is defined specifically for it. If none exists, it will *inherit* a layout in a defined manner as follows:

- If a layout exists for the given business area and project, this is used
- If this does not exist, the layout that exists for this business area's master project will be inherited and used
- If this does not exist, the layout that for the global level will be inherited and used. Every inbuilt layout always exists at the global level, and therefore is always available to be inherited if a junior layout does not exist to override its use

Inheritance is an important principle to understand, as it governs how layouts are selected for each purpose throughout ExtraView. Further, if you decide to create a layout for a specific purpose such as the **Add** screen within a newly created business area, this does not imply that you must create layouts for all the other layout types, such as search filters and email layouts. Each layout is inherited independently, giving tremendous flexibility and reducing the amount of setup required.

Inheritance and User Role layouts

Layouts can be defined for each user role you have created within your system. This is described more fully below. However, note that you may combine the definition of layouts for user roles, business areas within these user roles, and projects within these business areas. For example, if you have a business area named *Customer Calls*, which may or may not be broken down into individual projects, you may define different layouts for different user roles such as *Technical Support* and *Managers*.
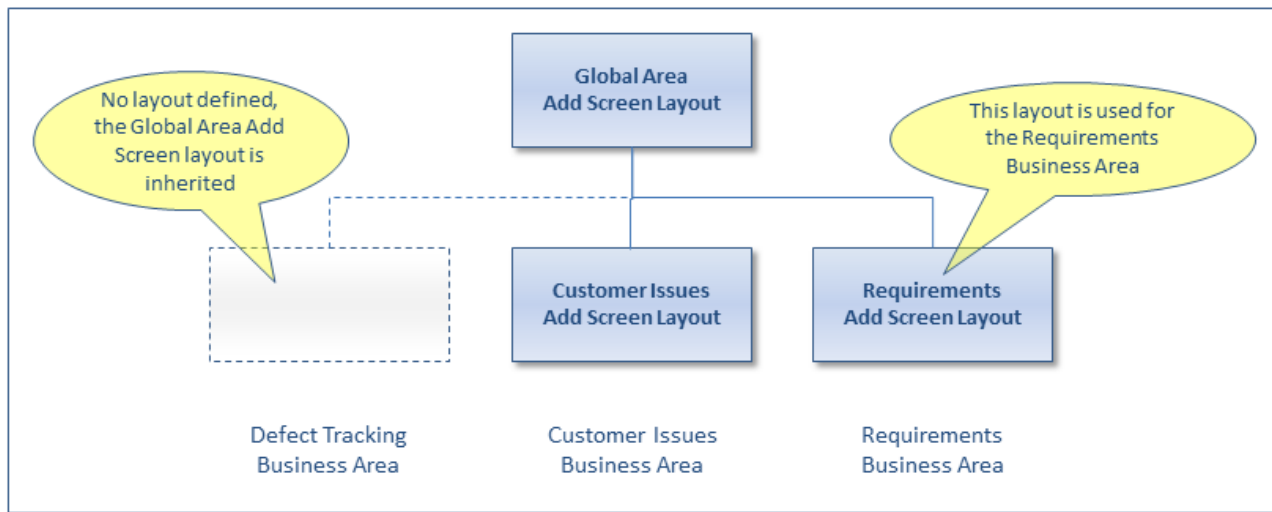
Inheritance of layouts

Note: You should set a default value in the data dictionary, for both the Business Area and the Project fields if you have their display type set to **Tab** and you intend to place the AREA and PROJECT fields on the *Add* screen layouts. However, the overriding setting is Remember Last Value. When this is set against Business Area and / or Project in the data dictionary, then the tab displayed upon entering the Add Issue screen, will be the one last used, not the default value.

Using rules, security permissions and layout cell attributes, you can make decisions on whether field-level data entered on one tab is either displayed on a second tab or is both displayed on the second tab and the values are stored with the record when it is inserted or updated. Consider an application where you insert and update customer information on a tab named "Customers". You may have a second tab named "Customer Issues" where you record issues the customers report. The approximate configuration may be something like this:

```
<== onchange ==>
# rules to set the customer fields in the Customer Issues area
if (CUST_LIST={CHANGED}) {
  CUST_CONTACT_NAME  = (customerLink).CUST_CONTACT_NAME;
  CUST_PHONE_NUMBER  = (customerLink).CUST_PHONE_NUMBER;
  CUST_ADDRESS       = (customerLink).CUST_ADDRESS;
  CUST_EMAIL         = (customerLink).CUST_EMAIL;
  CUST_PHONE_CELL    = (customerLink).CUST_PHONE_CELL;
  CUST_CONTRACT_NUM  = (customerLink).CUST_CONTRACT_NUM;
}
##
## Link types - link records from different business areas
##
<== link customerLink ==> AREA='Customers',CUST_LIST=CUST_LIST
```

With the addition of each customer issue, the Customer details are saved in the Customer Issues Business Area as the Customer is selected. To not allow the user to edit the customer details, you should set the permission of each field to read/write and then use a layout cell attribute on each filed in the Customer Issue layout to have a READ ONLY IF value. For example, set the CUST_ADDRESS field to be read only if the ID is not null. This will always be true, and therefore the field will display as read-only to the user, but the details will be saved in each record. If you only give read access to the user, the Customer details will not be saved in the record.

How Layouts are Selected

With inheritance, and the ability to define or override an individual layout for any combination of Display Target, User role, Business Area and Project, there is the potential for a large number of layouts in your system. Inheritance strictly controls the method by which an individual layout is either selected or inherited for display by the user.

When there is a choice of layouts, the layout is chosen based on:

- User role
- then Business Area
- then Project
- then Display Target

Optional Layouts

For efficiency, and to minimize the configuration effort, a few layouts are optional and are ignored if they are not present. When this happens, ExtraView may then default to another layout, and use this other layout instead.

Layout	Defaults To	Comments
ADD_CONFIRMATION	ADD_PROBLEM	When an issue is inserted, the behavior is to use the ADD_PROBLEM layout from the inheritance path to display a verification of the issue just entered. For most purposes this works in an obvious way and displays all the fields that the user just entered. Occasionally, you as the administrator may want to display a different set of fields. To accomplish this, create a layout of type ADD_CONFIRMATION in the business area and project of your site, or in the inheritance path. The fields that you place onto this layout will be those that are presented to the user when an issue is saved, as opposed to those on the ADD_PROBLEM layout. If the ADD_CONFIRMATION layout is not present, the ADD_PROBLEM layout is used as the verification screen
POST_EDIT_UPDATE	No Default	If this layout exists in the inheritance path, and the issue being updated is a member of a relationship group, then this layout is rendered to give the user the opportunity to apply the same updates to the related issue. If the layout does not exist, nothing is rendered, and only the current issue is updated This layout is deprecated, and only included in the documentation for fullness. Existing implementations may continue to use this, but new installations should not configure this layout type

Layouts

Layouts are a basic building block in putting together all the screen forms used within ExtraView. Layouts are all defined with a **layout type**. There are many inbuilt layout types, one for each of the basic screen forms, such as the **add**, **edit** and **detailed report**. Once a layout type has been defined, it can be reused in different business areas and projects, each with different fields as content, or they may be inherited.

You may define your own layout types. These are always used as *embedded* layouts within one of the inbuilt layout types. For example, you might create a layout type titled **Customer Details** and embed this within the *Add Issue Screen* of the **Customer** Business Area. The layout types you define may be shared and embedded across many inbuilt layouts, within any Business Area, Project, Role and Display target.

Layouts are inherited at any specific Business Area from the Global Area. A new layout may be defined to override the Global Area layout within any Business Area. Within a Business Area, the default layout resides within the Master Project. Further projects that are created either inherit the layout from the Master Project, or have a specific layout defined, overriding the layout within the Master Project. At any level in the hierarchy you can also define different layouts for different user roles. There is a default stored and used for all roles, but this can be overridden by creating a layout for any individual user role.

Some inbuilt layout types, such as the **Related Issue Display** and the **Repeating Row Layout** can only be embedded within other layout types. You can create your own layout types. These are used by embedding them within an inbuilt layout. Not all layout types can be embedded within all inbuilt layouts though. The documentation provides details.

Much of the way that layout types are used is within the Design Center; please consult this section of the guide for additional information. ExtraView has a number of default system layouts that always will exist within an installation. These are as shown below. However, the administrator can create additional layout types, and create layouts using these that are embedded within the default system layouts. For example, if you have a block of fields that you always want to display in the same order on several *Add Issue* or *Edit Issue* screens, you can build these into an embedded layout, and simply include this layout within the default forms.

Another use for embedded layouts is to create blocks of fields that are conditionally included in a form, dependent upon the value of a specific field. For example, in a defect tracking system, you may want to include a different block of fields to record information, dependent upon whether the user selects a **Category** of *Software, Hardware* or *Documentation*. At this time, embedded layouts are not supported within *Search Filter* and *Report* layouts. Screen layouts can be created in a specific hierarchy, allowing many different layouts to be displayed according to your system design.

- There is always a default layout for the following screens and reports. These are provided with the default installation of ExtraView and can be altered to suit your needs, although they should never be deleted:
  - Add Issue screen

- o Attachment Section
- o Attachment History
- o Edit Issue screen
- o Brief Email Notification layout
- o Filter layout for knowledgebase on the Home Page
- o Full Email Notification layout
- o Notification Section
- o Relationship Group Email Filter
- o Relationship Group Filter for managing relationship groups
- o Related Issue Display
- o Embedded Repeating Record layout
- o Detailed Report layout
- o History Report layout
- o Quicklist Report layout
- o Full Search filter screen
- o Quick Search filter screen
- o Chart Filter screen
- o Search Aging Report
- Each of these layouts is described within ExtraView with a Layout Type. A different Layout Type exists for each of the standard layouts, but the same Layout Type is used if a layout is created for a different Business Area, Project or User Role
- You can create new Layout Types. For example, if you want a new embedded layout, you can create a specific Layout Type for this, and then reuse this as often as needed, either as part of different Business Areas, Projects, User Roles or as different layouts that are selected dependent on the value of a field
- For each user role in your installation, you can define alternative layouts for each of the above. Once again, if you do not define an alternative, the default layout will be inherited and used
- If your installation uses Areas and Projects, then you can specify a different layout for each of the above, at both the Area and Project level. If you do not do this, then the default layout will be inherited and used.
- If your installation uses Areas and Projects, you may still define alternative layouts for each user role. This allows you to define an alternative layout for each and every user role and each Project and each Area
- Security privileges set for all fields will take precedence over the placement of a field on a layout. Thus, you can use the same layout for many user roles, but control the visibility and use of all fields simply, with security
- It is important to always include the ID field on the add and the edit layouts. This is because this is the key identifier for any issue you create or identify. However, if you do not want the field to be visible, then you can use a layout cell attribute to hide the field.

Note: The initial installation of ExtraView provides a standard layout for all inbuilt screens and reports. These can be altered for your organization.

Creating & Updating Layouts

Layout types can be created from the administration utility named **Create and Maintain Layout Types** or can be created while editing a layout within the Design Center. Once the layout type exists, you can choose to use it within a combination of Business Area, Project, Role and Display Target, from the prompt **Add a new layout for the entire system.** The following screenshot shows the initial presentation of the Design Center.

Initial Design Center Screen

The initial layout is designed for simple access to the functionality of the Design Center. You choose the Business Area within which you want to work, and can then use the display to navigate to the layout you want to work with, or you can add a new layout type into the current Business Area. The assumption is that you are always going to work within the default Project for the Business Area, and to use the Desktop browser as the display target for the layout you are working with. The default also assumes the layouts you are wortking with will all use the same user role.

If you want to see the full navigation to all the potential layouts, click on the ⚙ icon to the right of the **Layout Selection** title. The screen then offers additional options to navigate to the layout you want to work with:

Complete Design Center Screen Options

Note: The rows in the table with a white background show layouts that are defined explicitly within the current selection of Business Area, Project, Desktop Target and User Role. The layouts with the grey shaded background show inherited layouts.. To see the complete layout details, just hold your mouse over the layout type in the table.

1	Use this icon to change between the initial and complete views of the layouts within the Design Center
2	This select list allows you to navigate to the appropriate Business Area
3	This select list allows you to navigate to the appropriate Project
4	This select list allows you to select the Display Target you want to work with. The choices are Desktop, Tablet and Phone
5	This select list allows you to navigate to the user role you want to work with. The default is that the layout you work with will work across all roles
6	Use this select list to add a layout type into the current combination set with the Business Area, Project, Display Target and User Role. The layout type must already exist
7	The select lists within the table can be used as filters when working with large complex sites that contain many layouts
8	The layouts for the current selection are split into sections as follows:

- **Principal Layouts that Add and Update Issues**
  Virtually all Business Areas will have an *Add* and a *Edit* layout that is defined here. It is unusual to see *Add* and *Edit* layouts that are inherited from the Global Business Area
- **Report Layouts**
  These typically define the Detailed Report, History layout and the Quicklist
- **Query Filter Layouts**
  These define the search filter layouts

- **Email Notification Layouts**
  These are the layouts for the full and brief notifications
- **Embedded Layouts and Independent Layouts**
  This section contains additional layouts such as that used for mass updates and related issue displays that are not embedded within *Add, Edit*, and report layouts

9   Again, layouts that explicitly exist within the current selection are displayed with a white background

10   Again, layouts that are inherited from elsewhere are displayed with a grey background.

Note: The edit function allows you to delete layouts, whether they were created by yourself as an administrator, or created by ExtraView. **You should not delete an ExtraView created layout.** ExtraView will not function correctly if you do so.

Display Target-Based Layouts

The default is that a layout is defined for use as a Desktop layout and used within a supported browser. You should always define these within your system, as the inheritance will look for a Desktop layout to present to the user if a Tablet or Phone layout does not exist. If the user accesses ExtraView from a Phone, and no Phone layout exists, the Tablet layout will be inherited. If no Tablet layout exists, then the Desktop layout is used. Most users tend to use their phones and their tablets in portrait mode, as opposed to landscape mode. The recommendation is to create phone layouts that have a single column, and tablet layouts that use two columns. Many desktop layouts use four columns.

Role-Based Layouts

Layouts are typically defined first for general use across all the roles within a selection. Field-level permissions can be used to control much of the different behavior needed for different user roles. However, it is sometimes extremely useful to define a completely different layout for a user role. For example, you might have a set of layouts used within your company to add and to update issues, but want to present a much more concise and simpler layouts to your customers so they can add and update issues. You accomplish this by creating role-based layouts.

Editing Operations Within a Layout

- If you are not selecting a layout that will be used as the default across all user roles, select the user role to which the layout will apply from the select list at the top of the screen. Note that you can clone a layout for a different user role by editing an existing layout, altering the user role to which it belongs and saving the layout
- To add a layout where an existing layout of the same type already exists, click the **Edit** button beside the existing layout type example you would like to use. If no layout of an available type currently exists, select the desired layout type from the **Add new layout drop down** select list. You can then perform a "save as" operation
- When you click the **Edit** button, a screen similar to the following appears:

ExtraView Design Center

Save Layout	Delete Layout	Clear Layout	Preview Layout	Return	Print Page
1	2	3	4	5	6

Title IT Request Add / Edit **9**   Description IT Request Add / Edit **10**   Localize **11** **+ 12**

Fields & Layouts 13   **14** ↕ **—** **15**

	Col 1	Col 2	Col 3	Col 4
	← ✕ →	← ✕ →	← ✕ →	← ✕ →
		26 27 28		

Fields ▾ **16** Find a field... **17**   **18** New **—** **19**

ALT_ID
ATTACH_CREATED_BY_USER
ATTACH_LAST_UPDATED_BY_USER
BUTTON_COPY_TEST_CASE
CONTACT
CONTACT_LINK_FIELD
CUSTOMER_CONTACT_TITLE
CUSTOMER_SEARCH   **20**
CUST_ADDRESS
CUST_COMMENTS
CUST_CONTACT_NAME
CUST_CONTRACT_NUM
CUST_CONTRACT_RENEWAL_DATE
CUST_CONTRACT_SIGN_DATE
CUST_CONT_ACT_REN_DATE
CUST_EMAIL
CUST_FAX

Main grid:

Row	Col 1	Col 2	Col 3	Col 4
Row 1 ↑✕↓		AREA — Business Area		
Row 2 ↑✕↓		SHADE_PRE_1 **7**		
Row 3 ↑✕↓	ID — ID #	DATE_CREATED — Date Created	LAST_CHANGE_USER — Last Changed By	TIMESTAMP — Last Modified
Row 4 ↑✕↓	CATEGORY — Category	STATUS — Status	PRIORITY — Priority	SEVERITY_LEVEL — Severity
Row 5 ↑✕↓ **29 30 31**		SHORT_DESCR — Title		
Row 6 ↑✕↓	IT_DATE_REQUESTED — Request Date	IT_REQUESTOR — Requestor	IT_CONTACT_PHONE — Contact Phone	ORIGINATOR — Originator
Row 7 ↑✕↓	IT_APPROVED_BY — Approved by	ASSIGNED_TO — Assigned To		
Row 8 ↑✕↓		IT_SELECT_TABS — Helpdesk Tabs		
Row 9 ↑✕↓		LAYOUT.IT_SELECT_TABS — Helpdesk Tabs		
Row 10 ↑✕↓		DESCRIPTION — Description **8**		
Row 11 ↑✕↓		COMMENTS — Comments		
Row 12 ↑✕↓		SHADE_POST_1		

Cell Attributes 32   **33** New **—** **34**

✕ ✎ FIELD HTML MODIFIER
      onkeyup=autoSize(this);   **35**

✕ ✎ FIELD STYLE
      width:99%

Layouts ▾ Find a layout... **22**   **23** New **—** **24**

21
ADD_CONFIRMATION
ATTACHMENT_HISTORY
BUGS_SEARCH
CONTACT_SEARCH
CONT_SEARCH
CUSTOMER_SEARCH
CUST_SEARCH   **25**
CUST_SELECT_TABS
DEV_SELECT_TABS
EMAIL_BRIEF
EMAIL_FULL
HISTORY

Layout Attributes 36   **37** **—**

Layout ID: 1011
Advance by ? None ▾   **38**
Tab order by ? Column ▾

Design Center Screen

1. Save Button - Saves the current layout, leaving the administrator within the layout
2. Delete Button - Deletes the current layout. You must confirm the operation before the layout is deleted
3. Clear Button - Clears the current layout of all fields
4. Preview Button - This has been deprecated
5. Return Button - Return to the initial Design Center screen with the list of layouts
6. Print Button - Print the contents of the current window
7. The main grid where selected fields are shown in the position where they will be rendered on the layout. Note the scroll bars around the main grid which allow you to reposition the layout
8. A selected field on the main grid. The selected field is highlighted. Note that a field may span more than one row and / or column
9. The title for the layout
10. The description of the layout
11. Localize button - this only appears if localization is turned on for the installation, and allows you to localize the Description for the layout
12. Layout Options toggle - when this is clicked, a section similar to this opens:
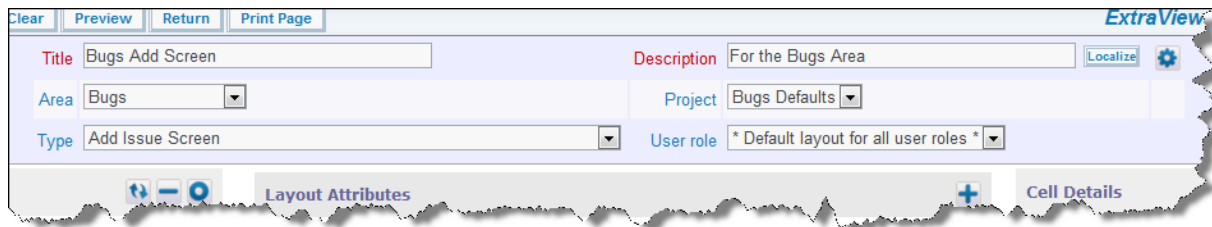
Use the select lists to alter the role, layout type, business area or project where you want to store the layout, as a "save as" operation

13. Fields and Layouts - This section is fixed to the left-hand edge of the screen and provides all the controls to create and place both fields and layouts onto the grid. In addition, you can create and maintain layout cell attributes to control the behavior of the fields, and layout attributes to control the behavior of embedded layouts

14. Invert field list button - this button toggles the field list so that field titles are displayed in order, as opposed to displaying their names

15. Minimize field/layout lists - this button toggles the field and layout section, reducig itsappearance to a single button on the screen

16. Field filter list - selecting this will display allow the filtering of the field list to a single field display type
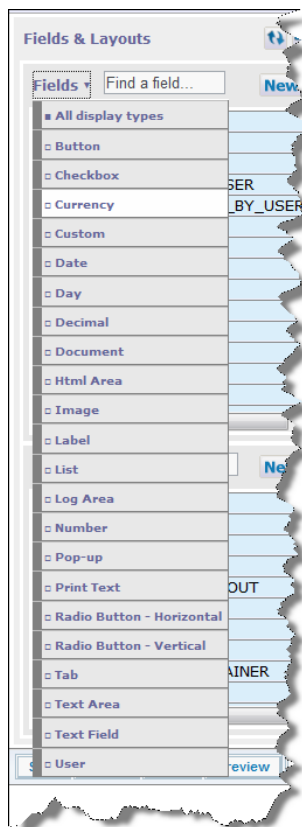
17. Find a field - start typing into this field and all fields in the list that match the pattern will be displayed, and all other fields will be dropped from the list

18. New field button - this brings up the data dictionary screen to add a new field. When you first click the button, you are asked if you are creating a new Field, Expression, Label or Screen field. If you have an ExtraView Standard or Professional license, as opposed to the Enterprise license, you are restricted in how many user defined fields you may add to your installation. For these license types, the screen will display how many fields you have created, and will inhibit you from adding additional fields when you reach the limits. For Standard licenses you may add 25 user defined fields, for Professional licenses, the limit is 150 fields. Enterprise licenses may add an unlimited number of fields

19. Minimize field list - this button toggles the field list, to reduce its appearance to a single row on the screen

20. Field list - this scrollable list shows all the available fields that you can drag and then drop on the main grid. Fields may be dropped on any empty location in the main grid. If you right-click on a field name, you will see details of the field. Note the **Edit** button. When you click this, a window will open showing the field in the data dictionary. You can make any modification you require to the field and its attributes at this point. If the field has a display type of *list*, *popup*, *tab*, or *radio button*, there is also the ability to **Edit field list values**. Clicking this pops up a window where you can modify the list values. An option is also available on this list to modify the permissions of the field. This pops up a window within which you can modify the permissions, role-by-role
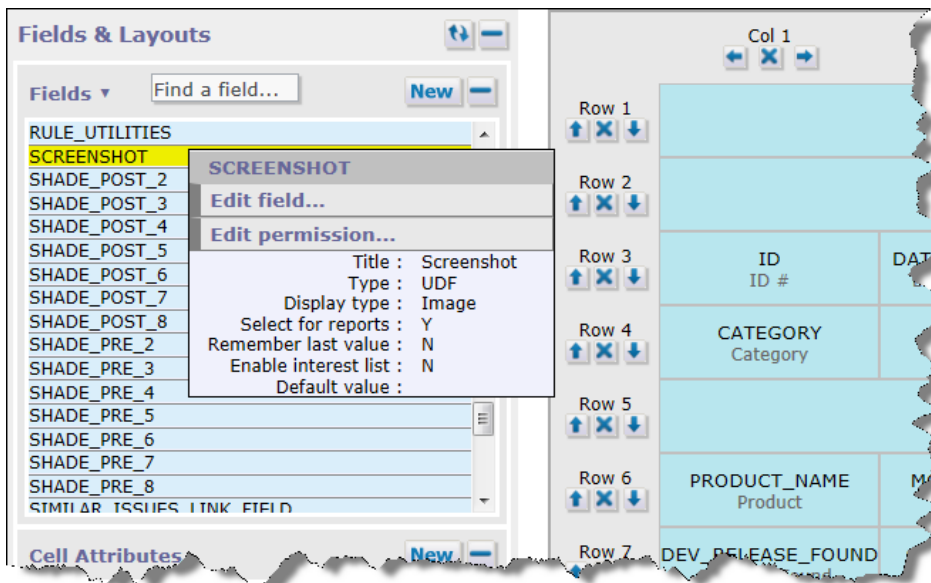
21. Layout section - this allows you to manage your layouts to embed within the main grid, including creating new ones
22. Find a layout - start typing into this field and all layouts in the list that match the pattern will be displayed, and all other layouts will be dropped from the list
23. New layout type button - this brings up the layout type screen to add a new layout type
24. Minimize layout list - this button toggles the layout list, to reduce its appearance to a single row on the screen
25. Layout list - this scrollable list shows all the available layouts that you can drag and then drop on the main grid. Layouts may be dropped on any empty location in the main grid. Use the **Layouts** button to filter the layouts by type
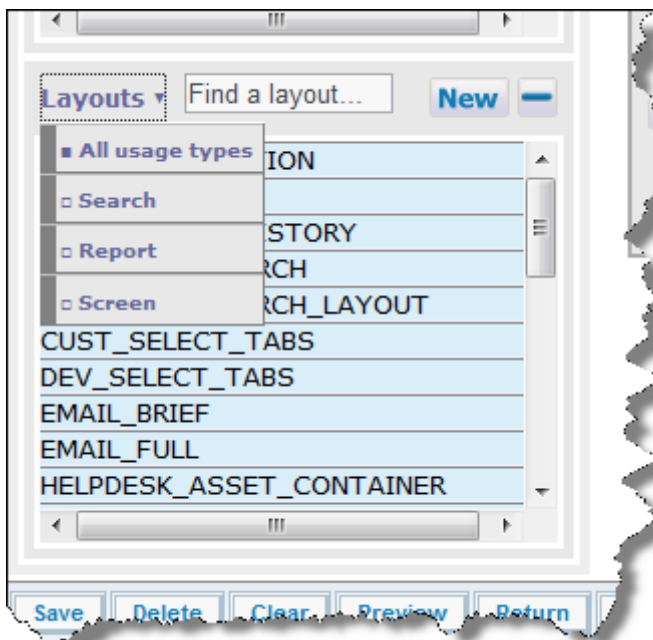
If you right-click on a layout within the list, you will see its details. Note that layouts you have created will have a **Permission** button. When you click this, a window will open allowing you to modify the permissions to the layout. You can make any modification you require to the field and its attributes at this point.
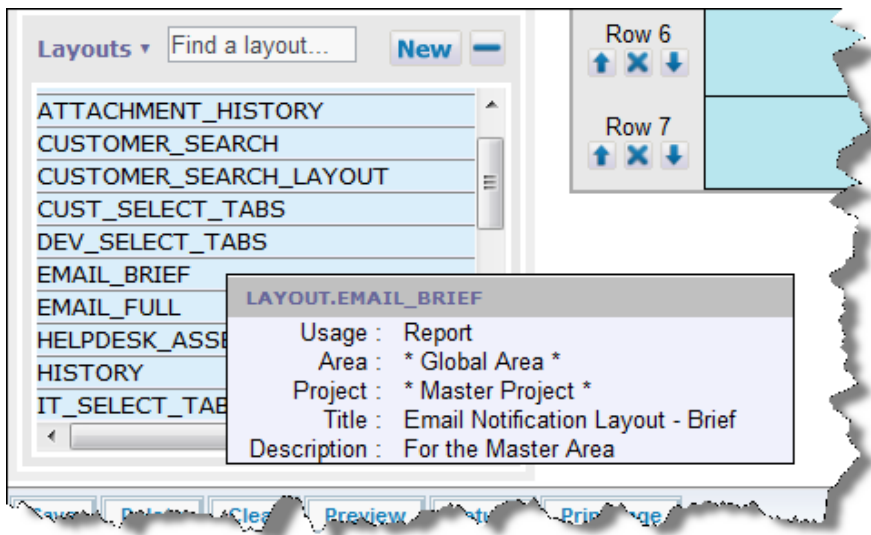
26. Insert column to left button - Insert a new column to the left of the button
27. Delete column button - Delete the column
28. Insert column to right button - Insert a new column to the right of the button
29. Insert row above button - Insert a new row above the button
30. Delete row button - Delete the row
31. Insert row below button - Insert a new row below the button
32. Cell attributes area - This is the area where all the layout cell attributes are managed. Click on a cell in the main grid to select it as shown in the screenshot below, and the cell details panel will display all the information about the cell
33. New layout cell attribute button - Click this button to add a new layout cell attribute. See the section on layout cell attributes for information on managing these
34. Minimize layout cell attributes button - this button toggles the layout cell attributes information, to reduce its appearance to a single row on the screen
35. The layout cell attributes that are configured for the selected field in the main grid. Note the buttons to delete and modify the layout cell attributes
36. Layout Attributes area - this area will display information on any selected embedded layout within the main grid
37. Minimize Layout Attributes - this button allows you to open and close the layout attributes section
38. The Advance By option is a layout attribute that allows you to set a direction to advance the placement of a field as you place these on the layout, to column or row. The Tab Order By layout attribute allows you to set the order for tab control by end users entering data into the form. This can be by column or by row

Field Manipulation on the Main Grid

- Selecting a field - click on the background to the cell
- Moving a cell - click on a cell and drag it to a new, empty location on the main grid. Note that any layout cell attributes to the field within the cell will be moved along with the field itself
- Removing a cell from the main grid - click on the background to a cell and drag it back to the field list.
- Layouts - perform similar actions as those above to manipulate layouts
- Extending a cell over multiple columns - click on the right-hand edge of the cell and drag the edge to the right over empty cells. Release the mouse over the rightmost cell to which you want to extend the cell
- Extending a cell over multiple rows - click on the bottom edge of the cell and drag the edge downwards to the bottom row to contain the cell. Release the mouse
- Selecting the text in a field to copy to the clipboard -
  - Click on the cell to highlight the cell
  - Either Ctrl-Click or Double-Click on the cell and the text is highlighted
  - Ctrl-C to copy the text within the cell to the clipboard
  - You may then paste the text anywhere else you require.

Duplicate Fields on Layouts

It is not permitted to have the same field included on the same layout or different layouts that are grouped together by embedding one layout within another. ExtraView catches a field duplicated within a single layout within the design center, but the design center cannot catch the situation when a field is duplicated by virtue of being contained in an embedded layout. The reason for this is that when you create an embedded layout, it may be included in any one of a number of other layouts, and each of these may or may not have the field duplicated. The error of duplicated fields can therefore only be caught at runtime. When testing your installation, you may see an alert message similar to the following screenshot:
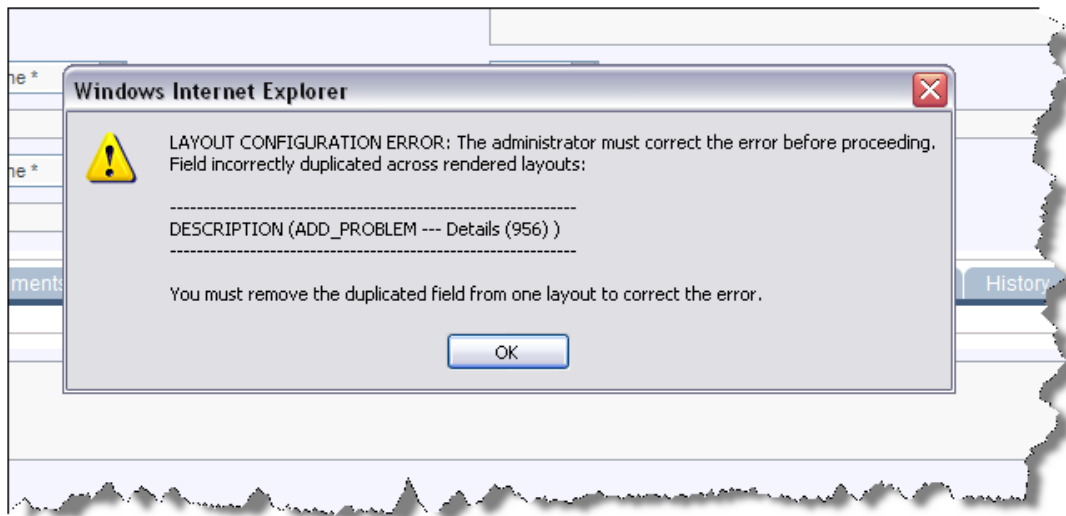
The above example shows that the field named **DESCRIPTION** is duplicated. First it is included on the **ADD_PROBLEM** layout, and then on the embedded **Details** layout. Further, the ID of the **Details** layout is displayed. You must resolve the duplication by removing one of the fields from one of the layouts before your installation will function correctly.

Using the Layout Description as the Title to *Add* and *Edit* Screens

The **Description** field on *Add* and *Edit* layouts has a special property. It may be used as the title to these screens, allowing the administrator to provide different titles for each *Add* and *Edit* screen. For example, you might give the *Add* and *Edit* screens different titles in different Business Areas. The standard behavior is that the title of the data dictionary field for the screen is used as the screen title. If you give a title of **$$LAYOUT.TITLE$$** to any of the following fields, then the layout description will be used as the title to the screen as it is rendered.

* ADD_PROBLEM - The *Add* Issue screen
* ADD_PROBLEM_SUMMARY - The *Add* Issue Verification screen
* EDIT_ISSUE_SUMMARY - The *Edit* Issue screen

Layout Rendering and Ajax

The layouts are rendered by the user's browser when they navigate to the appropriate function for the layout. There are many occasions when ExtraView uses Ajax calls to the server from the browser to handle the updating and refreshing of information on the layout. For example, the layout cell attributes **Visible If** and **Read-Only If** use Ajax calls to refresh only the part of the screen that is affected with the logic.

The biggest upside of this is vastly improved performance, compared to refreshing the whole screen, but there are a few consequences that should be understood.

* If ExtraView sees that an entire row of cells on your layout is empty, i.e. there are no values with write permission, and there are no values to render with read-only permissions, then the entire row is suppressed and is not rendered. This is most often the desired effect as there is not "empty" space on the screen
* The fields that are suppressed in this way do not exist within the browser's memory (what is known as the DOM). This means that the fields are not available to be processed using JavaScript. You might therefore try to manipulate a field within a hidden row, and find that your logic does not work as expected. The solution is to always have a visible field on the row, even if that field is just a placeholder. Alternatively, use a layout cell attribute of STYLE, and place a CSS style on the field of **display:none**. This places the field into the browser's DOM, but it is made invisble with the CSS style.

Built In Layout Types

Layout types are used to name and label each different type of report or layout. These are largely used by ExtraView to describe built in functions such as the Add Issue screen, the search filters. A layout type allows many individual layouts of the same type to be created. For example, an Add Issue screen can be created for each user role, and for each area, and for each project in the entire system. As an administrator, the most likely reason you will have to create additional layout types will be if you want to create new embedded layouts. Embedded layouts may be contained within one or more layouts. A typical use of this feature is to have an embedded layout that is selected according to the value of a field such as Category. For example, if Category = DOCUMENTATION, then a set of fields on an embedded layout related to documentation issues can be displayed. If Category = HARDWARE, then a set

of fields related to hardware issues can be displayed. The following built in layout types exist in a new ExtraView installation. You typically need to create additional new layout types if you want to create layouts that you will embed within other standard layouts.

- ADD_PROBLEM - the add issue layout
- ADD_CONFIRMATION - the confirmation/verification screen displayed after a new issue is added
- ATTACHMENT_HISTORY - the layout for the history of attachments
- ATTACHMENT_SECTION - the layout for file attachment data
- EDIT_PROBLEM - the edit issue layout
- EMAIL_BRIEF - the layout for brief email mode
- EMAIL_FULL - the layout for standard email notifications
- HISTORY - the layout for history audit trails
- MASS_UPDATE_REPORT - this is a layout used to control the content of mass update screens
- NOTIFICATION_SECTION - this layout contains all the fields related to notification of issues when they are update
- POST_EDIT_UPDATE - the layout that is triggered following the update of an issue with related issues
- RELATED_ISSUE_DISPLAY - the layout for displaying related issues
- RELEASE - the layout for repeating rows
- SEARCH_AGING_REPORT - an internally used layout used to create aging reports
- SEARCH_CHART_REPORT - an internally used layout used to create charts
- SEARCH_DETAILED - the layout for the detailed report
- SEARCH_EMAIL - the layout used as a filter when creating ad hoc email notifications
- SEARCH_EXPANDED - the layout used to created filter screens within reports
- SEARCH_QUICK - the layout used on the Query screen
- SEARCH_QUICKLIST - the layout used to produce Quicklists
- SEARCH_RGFILTER - the layout used as a filter for relationship groups
- SIMPLE_COL_SELECT - an internally used layout for the preparation of column reports
- SUMMARY_REPORT - an internally used layout for the preparation of summary reports

Creating New Layout Types

To create a new layout type, click the administrative tab named Create and Maintain Layout Types. You can also access this screen from within the Design Center, by clicking the **New** button within the **Layouts** Panel

The following screen appears:

	Name	Title	Usage	Created	Last updated
Edit	ADD_PROBLEM	Add Issue Screen	Screen	1/1/05 12:00 AM ExtraView	1/1/05 12:00 AM ExtraView
Edit	EDIT_PROBLEM	Edit Issue Screen	Screen	1/1/05 12:00 AM ExtraView	1/1/05 12:00 AM ExtraView
Edit	SEARCH_QUICKLIST	Report Layout - Quicklist Report	Report	1/1/05 12:00 AM ExtraView	1/1/05 12:00 AM ExtraView
Edit	SEARCH_DETAILED	Report Layout - Detailed Report	Report	1/1/05 12:00 AM ExtraView	1/1/05 12:00 AM ExtraView
Edit	HISTORY	Report Layout - History Report from Edit Screen	Report	1/1/05 12:00 AM ExtraView	1/1/05 12:00 AM ExtraView
Edit	SEARCH_EXPANDED	Search Filter Layout - Full Search	Search	1/1/05 12:00 AM ExtraView	1/1/05 12:00 AM ExtraView
Edit	SEARCH_QUICK	Search Filter Layout - Quick Search	Search	1/1/05 12:00 AM ExtraView	1/1/05 12:00 AM ExtraView
Edit	SUMMARY_REPORT	Report Layout - Summary Report	Report	1/1/05 12:00 AM ExtraView	1/1/05 12:00 AM ExtraView
Edit	RELEASE	Repeating Row Record	Screen	1/1/05 12:00 AM ExtraView	1/1/05 12:00 AM ExtraView
Edit	SIMPLE_COL_SELECT	Report Layout Template for Column Reports	Report	1/1/05 12:00 AM ExtraView	1/1/05 12:00 AM ExtraView
Edit	EMAIL_FULL	Report Layout - Full Email Notification Layout	Report	1/1/05 12:00 AM ExtraView	1/1/05 12:00 AM ExtraView
Edit	EMAIL_BRIEF	Report Layout - Email Brief Notification Layout	Report	1/1/05 12:00 AM ExtraView	1/1/05 12:00 AM ExtraView
Edit	SEARCH_EMAIL	Search Filter Layout - Email Filter for Ad Hoc Email from Edit screen	Search	1/1/05 12:00 AM ExtraView	1/1/05 12:00 AM ExtraView
Edit	SEARCH_RGFILTER	Relationship Group Filter	Screen	1/1/05 12:00 AM ExtraView	1/1/05 12:00 AM ExtraView
Edit	SEARCH_CHART_REPORT	Search Filter Layout - Charts	Search	1/1/05 12:00 AM ExtraView	1/1/05 12:00 AM ExtraView
Edit	DEV_SELECT_TABS	Select tabs for bugs area	Screen	1/1/05 12:00 AM	1/1/05 12:00 AM

Maintaining Layout Types

To add a new layout type, press the Add button and use the following screen:

Maintaining Layout Types

Adding a New Layout Type from the Design Center

Layout types must have a name, a title and a usage. The following usage values are valid: Screen, Report, Search, User Report.

When you edit a layout type, you will see a list of all the places where the layout type is used as a layout within your installation, and you can click the **Edit** button by each usage, to take you to the Design Center where that layout is used.

If, and only if, you are creating a new layout type to be used as a repeating row, then complete the bottom section of the form. If you are creating a new repeating row layout type, then you also give this a name and a title. When you press the radio button to **Create a new repeating row type** these fields are shown. You can create a new layout type using an existing repeating row type, in which case you choose the option to **Select an existing repeating row type**. In this case you select the repeating row type from the list.

Note: The edit function allows you to delete layout types, whether they were created by yourself as an administrator, or created by ExtraView. **You should not delete an ExtraView created layout type**. ExtraView will not function correctly if you do so.

Modifying Layout Types

You may modify the title of a layout, and its usage from the Edit screen of layout types.

In addition, the edit screen allows you to see all the places where a layout is used, and to use the Design Center to modify these layouts.

Modifying Layout Types

If you click on the Edit button with the utility, you are taken directly to the Design Center where you can modify the layout.

Clearing and Deleting Layouts
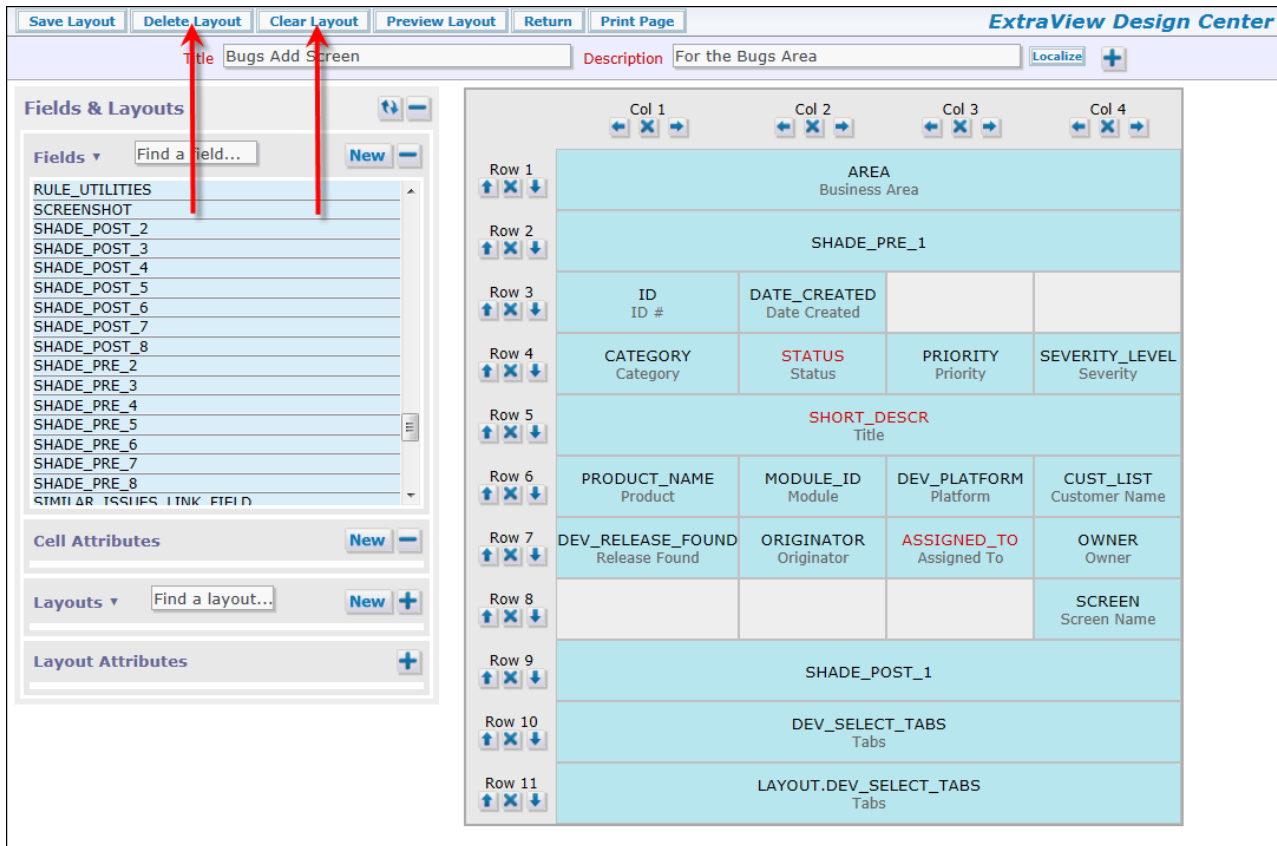
Design Center Screen

Clearing Layouts

1. To clear a layout, press the **Clear Layout** button.
2. Follow the directions found on the alert message than will then appear.
3. Press **Save Layout**.

You will be left in the current layout, but all fields are deleted from the layout.

Deleting Layouts

1. Press the **Delete Layout** button.
2. Follow the directions found on the alert message that appears.
3. Press **Save Layout**.

Note that you cannot delete the following layouts from the Global Area:

- ADD_PROBLEM
- ATTACHMENT_HISTORY
- EDIT_PROBLEM
- EMAIL_BRIEF
- EMAIL_FULL
- HISTORY
- SEARCH_CHART_REPORT
- SEARCH_DETAILED
- SEARCH_EMAIL
- SEARCH_EXPANDED
- SEARCH_QUICK
- SEARCH_QUICKLIST

After confirming that it is OK to delete the layout, it is permanently deleted from the database and is unrecoverable.

Performing a "Save As" Operation

It is often useful to be able to duplicate a layout in a different area, project or role, or to simply save one layout type as a different type. For example, after composing an *add* layout, you may want to save it as an *edit* layout. This operation is easily accomplished by just opening up the layout you want to copy, clicking on the button to the right of the layout description then changing the Business Area, Project, Type or User Role.

You may also change the Title and Description. Once you have done this, simply save the layout. Note that if you are changing an *add* or *edit* layout to a report style of layout, any field whose data dictionary entry for the attribute of **Select for Reports** is equal to *No* will be removed, as these are not supported on report layouts. Similarly, any embedded layout whose usage is not set to equal **Report**, will be removed.

Layout Cell Attributes

- Add new comment

Layout cell attributes allow you to modify the appearance or function of a field or an embedded layout within a layout. This includes creating dependencies between fields and controlling related issue displays.

Layout cell attributes

Note the section on the screenshot that displays the Cell Attributes. This displays the attributes that have been set for the cell, as well as attributes that were set for the field in the data dictionary as global field attributes. The global attributes are displayed with the darker background. You may either use the global attribute inherited from the field definition, or override the global attribute with a local layout cell atrtibute. If you add a new attribute, a screen similar to the following is displayed:

Adding a layout cell attribute

As you click on a button beside an attribute, a help tip appears to give guidance on how to use the attribute; also, you see the selections you need to make to configure the attribute. Not all attributes require selections. For example, the VALIDATE HIDDEN attribute does not need any additional configuration.

Note that you will only see a subset of the complete number of layout cell attributes for a given field or layout selection, these being the ones that are valid for the field display type or layout type that you have selected in the main grid. If a layout cell attribute does not appear when you are editing the attributes, it is because that attribute is not valid for that field or embedded layout. The attribute types are the different class of modifiers that can be applied to each cell. These are:

Attribute	Purpose
ACT ON FIELD CHANGE	This attribute triggers relationship group actions when the field value upon which this attribute is based, changes. This is used when you set up a search layout to bring back results, on an ADD_PROBLEM or EDIT_PROBLEM layout. This configuration relies on an intermediate field, the RID LINK FIELD NAME. You place this attribute on that field, so that when it receives your selection, it uses the information to update the embedded layout that contains the related issues from the results. See the [example here](#) for use of this attribute.
ADD NEW	Adds a * New * value entry to list and popup field types, on ADD_PROBLEM and EDIT_PROBLEM layout types only. It modifies all user defined list fields, the **PRODUCT_NAME** field, the **MODULE_ID** field, the **RELEASE_FOUND** or the **RELEASE_FIXED** fields so that a value of * New * apprears within the list. When this * New * value is selected a modal popup window allows the addition of a new entry into the list. This has the same effect as an administrator using the list manager function to add a new entry to the list, but puts the ability in the hands of an end user without their requiring administrative privileges. This feature can be controlled by user roles.
	If the behavior setting named AUTO_POPULATE_ALLOWED_VALUES is set to YES, the an allowed value relationship is created automatically when the user adds a new value, and the parent value has already been selected on the *add* or *edit* screen.

See the section titled End-User List Management for more information on setting up this feature.

AGGREGATE EXPRESSION	This layout cell attribute implements a method to compute an aggregate expression on a per-row basis on a Related Issue Display. The actual calculation is performed in a user custom exit, so this attribute must be accompanied with Java code. This layout cell attribute only works when used with expression display type fields placed on a Related Issue Display and has no effect in other locations. See the page Field Aggregate Expressions for more information.
AGGREGATE EXPRESSION TITLE	This layout cell attribute provides a title to the aggregate expression. Please see the page Field Aggregate Expressions for more information.
ALTERNATE TITLE	This will rename the screen title in the Data Dictionary to the value in the Field value, for the field in the Field entry. Note that if this is used in a screen layout, you can include HTML in the alternate title. If HTML is included within the alternate title, then the behavior setting of LABEL_WRAP_POSITION is ignored. This allows total control of the formatting of the label within the layout. However, if you access the field through the API or CLI then the HTML will be passed through to the output. As the text for the alternate field title is placed within double quote marks (") in the HTML generated when the form is rendered, you should not use these within the alternate field title. You may use single quote marks (') however, and these are generally interchangeable.

It is a best practice to create a title for all field without any HTML within the data dictionary, and to only use HTML within individual layouts. This makes the fields easier to view under most circumstances, especially report editors and report output.

You should restrict the use of HTML <br> tags within Related Issue Display layouts as they may appear cut off at the bottom of the title. The titles within a Related Issue Display are sized exactly to a single row, so that they may be frozen in position when a long list of related issues is scrolled.

Example: Within an organization, you may label issues as *Defects*, but you may want to label them as *Issues* for your customers. On the new attribute form, you would set *Alternate field name* equals *Issue*

Example: A field on the form has the particularly long title of *"Target Software Release"* that wraps in an ugly way on the screen using the system-wide default you have set with the behavior setting, LABEL_WRAP_POSITION. You can control where the line breaks occur by using something similar to *"Target
Software
Release"*

Example: You want to remove the title to a field on a layout. It is not valid to use a blank title. The solution is to use an alternate field title of * *. This inserts the character set value of a space into the layout. |
| BUTTON ACTION | This attribute is applied as a cell attribute to the **ADD BUTTON** within a related issue display header. The **ADD BUTTON** is configurable to allow a child issue to be added to the current parent issue that is being displayed on an *add* or *edit* layout. The action defines what happens when the user presses the add button. |

- **Target Business Area** - This defines the Business Area that will be used for the *add* screen displayed when the button is pressed
- **Target Project** - This defines the Project that will be used for the *add* screen displayed when the button is pressed
- **Check Required Fields** - If you check this option, then all required fields on the parent issue will be checked for their presence before the child window is displayed. This can be extremely useful if you are going to push down field values from the parent to the child issue, and you want to ensure that the child window is always initiated with a value within a field
- **Presentation** - You may allow the user to create the child issue within a modal popup window, or with an inline operation, where a new row opens up within the related issue display, and the user enters the data for the new child issue within this new row
- **After Submitting the Child Issue** - This option allows you to select the action after the child issue is created. This applies to the modal popup option of the Presentation setting only. You can either have only the related issue display that contains the newly added issue updated, or you can have the parent issue updated and the entire screen updated. The latter is only required when there are some business rules that triggered changes to the parent issue when the child issue was submitted.

  Note that if you select **Inline** for the presentation, the related issue display is not refreshed after the inline add operation. The reason for this is that there may be a sort order on the display, and there may be many issues wiithin the related issue display. If the newly added child issue does not appear in the spot where the user added it, some confusion may exist as to whether the child was or was not added

- **Push Down Field List** - Double click on the fields whose values you want to push down from the parent issue to the child issue. The selected fields are displayed in the right-hand list. Note that you can use the **Find a field ...** option to search for a field within the list
- **Show Generate Script** - This button is provided for advanced usage. It is possible to configure button actions that are more complex than can be done with the **BUTTON ACTION** utility. This option allows you to generate the JavaScript that is close to what you require. You can then copy this JavaScript into the **ONCLICK_JS** attribute, and modify it for your requirement. You should not save the **BUTTON ACTION** if you are using **ONCLICK_JS** and vice versa.

| COLLAPSE REPEATING ROWS | This attribute should only be used on an embedded repeating row layout with a type of report. Typically this is used when you are rendering fields from different repeating row layouts within the emedded layout, and you want the values to align as far as possible, to the top of the embedded layout. As the values from the |

different fields are likely to have originated in different records in the database, they would normally be rendered on different rows. This attribute shifts the values to be aligned from the top of the display for the embedded repeating row

This will modify the display of the value of the item according to a user-defined format. The format can include text as well as the values of other fields. There are specific rules for using this attribute successfully:

- The target field for this attribute must be a field with a display type of *text*
- All the fields names referenced in the format must exist on the layout
- The field should be read-only to users
- There is no checking of field dependencies. For this reason, the value on the rendered screen could become stale during the edit process, for example if one of the referenced fields changes values. Of course, this is not a problem on a report layout.

DISPLAY FORMAT

Example: You can set up a new field on a layout, modifying the *ID* field for display to be of a specific format such as PROJECT + '.' + *ID*. The *ID* will be the standard sequence, but will show a value such as *MyProj.12345*.

DO NOT CLONE

This setting is primarily used in conjunction with mapping fields within one ExtraView installation to another ExtraView installation, vi the peer integration daemon. If an issue that is mapped is subsequently cloned, the mapping identity field should not be propagated to the cloned issue. The reason for this is that it would establish a mapping from one peer issue to two issues (i.e. the original issue and its clone). This is not a fatal error, but the behavior produces an ambiguous result, causing updates to the peer issue for each update to either the original issue or the cloned issue.

To handle this behavior, the data dictionary attribute named DO NOT CLONE prevents the cloning of the field value for specific fields. This attribute is set when the field is used as a mapping field in the integration daemon. Setting the DO NOT CLONE data dictionary attribute for a field means that the containing field's value will not be propagated to any clone of an issue - its value will be null in the cloned issue.

DRILLDOWN

You should only add this attribute to a field when it is to contain an issue ID. Adding the attribute will turn on the ability to click on the field value in order to be able to drill down into that issue and open up an *edit* screen

ENCRYPT

Encrypt the values within the field

EXPRESSION TEXT

This is used to add a calculated expression to a user defined report field.

GRIDEDIT BUTTON

This attribute provides a button with the title **Gridedit**, on the preheader of the Related Issue Display. When clicked, this button renders all the issues on a Related Issue Display in Quickedit mode within an *edit* screen. The user is required to click the **Update** button on any issue before it is saved in the database. If you wish to alter the title on the button, it is controlled with the data dictionary field named GRIDEDIT_BUTTON.

HEIGHT

This applies to text area, log area, print text fields and embedded relaetd issue display layouts. The height is the approximate number of characters high that the field is displayed with when the user first brings up an *add* or *edit* layout.

Example: To create a text area field where the initial display height is 20 characters, add a HEIGHT attribute with a value of 20. Note that RELATED_ISSUE_DISPLAY layouts are automatically sized in height, according to the number of issues they contain, and the FIELD HEIGHT is the maximum height that a related issue display will occupy, before a vertical scroll bar is added.

HIDE VALUES

This attribute allows you to hide individual values within a field which contains a list of values. When you select this attribute, you are presented with a list of all the values within the field, and you select the values you want to hide unconditionally. This may be used with great effect on fields such as AREA on an *add* layout, to constrain a user to only be able to add new issues into specific business areas, and to lock the user out from other business areas. Note that if you configure this attribute, you may not then configure a FIELD VISIBLE VALUES attribute on the same field.

This attribute is only supported on fields with a display type of List (including multi-value list, put not popup lists), Radio Button and Tab.

HIDE VALUES IF

This attribute is similar to HIDE VALUES, except that you can conditionally hide the values in a list, according to a logical expression based on other fields within the form,

HIGHLIGHT VALUE IF	This provides the ability to alter the color of a field value, based upon a logical expression defined within the layout cell attribute. This attribute only works with display types of *text* and *text area*. The color used to highlight the value is the color defined within the behavior setting HIGHLIGHT_COLOR. **Example**: To highlight the ID of an issue, if the issue is a customer issue, you might apply a layout cell attribute to the ID field – **HIGHLIGHT VALUE IF CUSTOMER IS NOT NULL**.
HIGHLIGHTED IF	This is to indicate if a field value is to be highlighted with an alternative color depending on the value of another field. The color may be selected by setting the value for the behavior setting HIGHLIGHT_COLOR. You may have more than one FIELD HIGHLIGHTED IF attribute on a field. A limitation at this time is that you may only use the attribute with a condition of **is not null Example:** We want to highlight the ID of an issue, if the issue is a Customer issue. To achieve this, set the element attribute of Highlighted on the ID field, if the field named Customer is not null.
HOVER VIEW DEFINITION	This attribute is used to define hover text that appears over buttons, when the user holds the mouse steady over the button. The administrator can configure which fields are displayed in the hover popup with this layout cell attribute. When you select the attribute, you will see two field lists. The left-hand field list presents all the available fields. Double click on the fields you wish to add to the hover text and they appear in the right-hand panel. Alternatively, drag the field to the right-hand panel from the left-hand panel. Drag the field out of the right-hand panel into the left-hand panel to remove it from the hover text popup. Typically this is used to bring up the full description, plus any other fields, of an issue for viewing, as a shortcut to the user having to drill down into the issue to see the key information. The definition of the popup is valid on various, but not all buttons. It works on VIEW_BUTTON, EDIT_BUTTON, DELETE_BUTTON, HISTORY_BUTTON and QUICKEDiT_BUTTON, on *add*, *edit*, report and related issue display layouts.
HTML MODIFIER	This will allow the user to provide values for additional attributes within an HTML tag on an *add* or *edit* layout. You may have any number of HTML modifier attributes on a cell. It is likely that you will need a reasonable understanding of HTML tags and attributes to use this layout cell attribute. **Example**: A *DISABLED* HTML modifier to the attribute will disable the field. Most browsers would then show this field as "grayed-out". **Example**: Use the HTML modifier to call a user-defined JavaScript function or to call a pre-defined JavaScript function within ExtraView. The function resides on the server and is automatically included in the HTML that is generated for the layout. The syntax specified within the HTML modifier to call a user-defined JavaScript function will be similar to: *onclick='myFunction(param1, param2)';* For a more detailed explanation of custom building JavaScript functions, please see the ExtraView User Custom Programming Guide. If you want to add a style attribute to a layout element, you should use the **style** (rather than the html modifier attribute) otherwise your cell element may have multiple style attributes, which is invalid HTML syntax. A reference providing a list of all valid CSS styles may be found at http://www.w3schools.com/css/css_reference.asp. Note that this layout cell attribute only works when used within a layout that is rendered as an *add* or *edit* screen.
IS RANK	This defines the field to be a rank field on the layout. Only apply this attribute to numeric fields
LABEL TAG	This allows the administrator to add a new attribute to the HTML <TD … tag that surrounds the label of the cell. This can be used to provide a different style for the label, or to inject any other valid HTML into the cell label tag. It is likely that you will need a reasonable understanding of HTML tags and attributes to use this layout cell attribute.
LAYOUT SEARCH EXPANDED MODE	The attribute may only be added to layouts that have a type of *search* and are used to provide a search and selection mechanism on *add* and *edit* screens. When this attribute is selected for a *search* layout, all the fields on the layout are rendered in expanded mode, allowing the end user to make multiple selections within the fields being used as search filters
LAYOUT SELECTED	This is for indicating which embedded layout to use on a screen depending on a tab or list value. This is one of two mechanisms that allow ExtraView to render alternative layouts, dependent on the value of a tab or list field.

Example: You want to embed a different layout, with a different set of fields on an *edit* screen, dependent on whether the user selects a **Category** of *Software* or *Hardware*.

Note: If you are creating layouts which are selected according to a list or tab value, it is highly advisable to set a default value for the list or tab field. If you do not do so, and the field does not have a value, no embedded layout will be selected.

The precise steps necessary to implement embedded layouts are documented in a following section, entitled **Adding Embedded Layouts**.

MAX FRACTION DIGITS	This attribute only applies within numeric type input fields on ADD_PROBLEM and EDIT_PROBLEM layout types. This provides an integer number that represents the number of digits to display, following a decimal point
MAXLENGTH	This will add a *MAXLENGTH=nnn* attribute to the HTML display tag for the field, controlling the maximum number of characters that can be typed into the field. The default if no attribute is set is 255 characters. Used in this way, the attribute works when used within a layout that is rendered as an *add* or *edit* form. 255 characters is the maximum length of any user defined text field that you create **Example**: Setting an attribute of MAXLENGTH equal to 25 on a field will restrict the amount of characters that can be entered into a field to 25 characters.
MODAL WINDOW	This layout cell attribute only has an effect if used on an EDIT_BUTTON within a Related Issue Display. When this attribute is set, and the user presses the Edit button on the Related Issue Display, a popup, modal window is used to provide the edit capability for the related issue. If this is not set, then the related issue is placed in a separate, new window. This provides control on whether you want to allow users to update a parent issue while editing a child or related issue.
NOT REQUIRED IF	This is the opposite condition to **FIELD_REQUIRED_IF** and is typically used when there are a significant number of values in a field, but the field only becomes not required on a few values. In this circumstance, it is faster to set up **FIELD_NOT_REQUIRED_IF** on a small number of values and set the overall required state of the field to Yes. Note that if you use the *greater than* or *less than* operators, they only work with numeric comparisons.
READONLY IF	This attribute indicates that the field value is to be displayed on the layout in a read only mode, based upon the value of another field. You may have more than one FIELD READONLY IF attribute on a field. For obvious reasons, this attribute only works when used within a layout that is rendered as an *add* or *edit* screen. Note that if you use the *greater than* or *less than* operators, they only work with numeric comparisons. **Example**: We want to display the Customer name within the field named Customer on an *edit* layout, if the Customer has already been added, but make the field read / write if there is no value so the user can enter the Customer name. To achieve this, add an attribute of FIELD READONLY IF Customer is not null. If you place a FIELD READONLY IF attribute on a layout cell that contains an embedded layout as opposed to a field, then all the fields on the embedded layout will be made read only, according to the conditions set with the logic of the attribute.
REMOVE ANY	Removes the * Any * entry from list and user fields. This attribute will cause the * Any * entry in a select list not to be rendered when the layout is used as query filter screen. This can be used to ensure that the user always picks a value from a list before performing a search. Note that this is similar behavior to making a field required on a layout.
REMOVE LABEL NOWRAP	By default, ExtraView does not allow the label of a field to wrap within the user's browser. Instead, there is a behavior setting named LABEL_WRAP_POSITION that allows the administrator to define a specific position where it will wrap. This gives a more consistent result as the user resizes their window on their browser. If you set this attribute, then ExtraView will allow the browser to control the wrapping for the field.
REMOVE NONE	Remove the * None * entry from list and user fields. This attribute causes the * None * entry in a select list on an *add* or *edit* layout to be removed, ensuring that the end user always selects at least one entry. This means that the field will become required, as a value is always selected.
REMOVE VALUE NOWRAP	By default, ExtraView does not allow the value of a field to wrap within the user's browser. If you set this attribute, then ExtraView will allow the browser to control the wrapping for the field. This is typically only an issue for read-only text type fields.
REPORT MAX FRACTION	This attribute may be set to define how many digits will appear in the rendering of a field with a display type of *Number*, *Decimal* or *Currency*, when the field is displayed as part of a report. It also controls how the value

DIGITS	of the field is rounded on reports and when displayed on the *add* or *edit* screen. Note that *Currency* data display types generally have the number of decimal digits defined by the currency type. For example, the US Dollar has two decimal digits.
REQUIRED	This attribute is used to make a field required unconditionally on an *add* or *edit* layout. Contrast this with the FIELD REQUIRED IF attribute which makes a field required conditionally. Note that fields using this attribute are immediately flagged in with a highlighted color on the grid of the layout
REQUIRED IF	This is for indicating if a field value is required depending on the value of another field. Note that Field Required If dependencies only work on fields with a display type of *List*. Further, they do not work if the field has been set in the data dictionary to be a multi-valued list, or if the field is a pop-up list. You may have more than one FIELD REQUIRED IF attribute on a field. For obvious reasons, this attribute only works when used within a layout that is rendered as an *add* or *edit* form. Note that if you use the *greater than* or *less than* operators, they only work with numeric comparisons. **Example**: The field on a form named CUSTOMER_NAME may become required if another field named CUSTOMER_ISSUE has a value of Yes. To achieve this, add a layout element attribute to the field named CUSTOMER_NAME. The value will be *Yes*, the field will be CUSTOMER_ISSUE and the equivalence will be *equals*.
RID FILTERS	This attribute may only be used on a cell containing a related issue display. It allows the administrator to set one or more filters on the related issue display. For example, you might want to filter the related issues to only display the *Open* issues, or you may want a much more complex filter. When you select this attribute, you will a dialog similar to the following. Use the query filters to compose the expression required, then update the attribute. Note that you cannot filter on the KEYWORDS field:

Filtering Related Issues

Note that there is a special value with the title of *** Value from issue ***in the list fields. This indicated that the filter will use the value that the user sets into the field when they are entering data. Consider this to be a dynamic filter. This special value appears for fields with display types of *List*, *User* and *Popup*. If you want to use this feature for other data types, you can enter **$$<ddName>$$** in the field.

	This attribute may only be set on Related Issue Display layouts within *Add*, *Edit* layouts, or sublayouts of these. When this attribute is used in conjunction with the HEIGHT attribute, and a scroll bar appears after issues have been added, the title to the Related Issue Display is fixed as the user scrolls.
RID FIX TITLES	It is not recommended that you place SIZE layout cell attributes on the fields contained within the Related Issue Display layout when you use this attribute. The best results are obtained when the browser sizes all the columns and column titles.

RID GROUP REFERENCE FIELD	If you do not use this cell attribute on the cell in a layout containing the LAYOUT.RELATED_ISSUE_DISPLAY (or other related issue layout), then ExtraView assumes that you are linking issues based upon the value in the ID field. If you have altered the behavior setting named ITEM_ID_DISPLAY to point to ALT_ID rather than ID, then issues will be related on the ALT_ID field rather than the ID field. However, you can use this entry to provide a link based upon the value of any other field on the layout. This is a very flexible feature, allowing arbitrary groups of issues to be linked, dependent upon any field.

This attribute is used when linking selected search results on an ADD_PROBLEM or EDIT_PROBLEM layout to a related issue display which will contain the selected records. This field is an intermediate field which will contain the ID's of the selected records. For most purposes, this field can be hidden completely on the layout, using the following two attributes:

RID LINK FIELD NAME

- FIELD STYLE: Visible if ID = -99
- LABEL TAG: style=display:none

See the example here for use of this attribute.

RID OMIT AUTO POPULATE	Bypasses the automatic population of related issue ID's for the field
RID ONCLICK_JS	This attribute is used on a related issue display to define JavaScript to execute when the user clicks on the RID PREHEADER ADD BUTTON. This can either be JavaScript itself, or a call to a JavaScript method in the UserJavaScript.js file.
RID PAGE SIZE	This attribute may be applied to a related issue display to override the default value in the RID_PAGING_DEFAULTS behavior setting. This value sets the number of issues displayed on a related issue display, once the count of issues exceeds the RID THRESHOLD value

This defines a title for the header section within a related issue display. Use this attribute on the embedded related issue display within a layout. The title will appear as shown in this screenshot:

RID PREHEADER
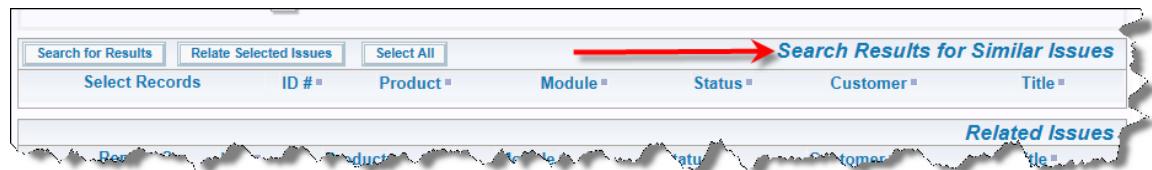
Creating a title for related issue displays

Use this attribute to place an "add" button on a related issue display layout. The button must already exist before you can use this on the related issue display. The button is created within the data dictionary with these attributes:

- Create the field from the **Label** tab of the data dictionary
- Provide a fixed name for the button
- Provide a title for the button - this is used to define the text that appears on the button when rendered as part of the related issue display
- The display type must be **Button**
- Other field attributes can be defined such as permissions, according to your requirements

RID PREHEADER ADD BUTTON

Add button on a related issue display

RID PREHEADER REFRESH BUTTON	See the example here for use of this attribute. This provides a "Refresh" button on a related issue display that contains search results. The user can select new search filters with this button to refresh the results. This attribute does not require a data dictionary field. Simply enter the title for the button that you require as the value to this attribute.

Refresh button on a related issue display
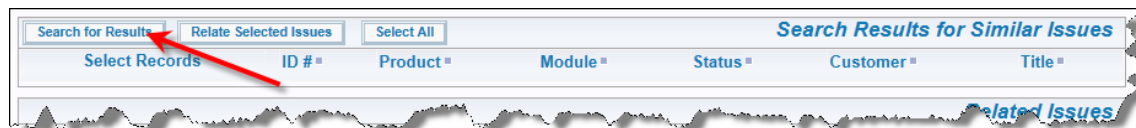
RID PREHEADER SUBMIT BUTTON	See the examples here and here for use of this attribute. This provides the "Submit" button on a related issue display that contains search results. The user moves the currently selected issue(s) from the related issue display to either fill fields on the current parent issue, or to relate the selected records to the current parent record. This attribute does not require a data dictionary field. Simply enter the title for the button that you require as the value to this attribute.
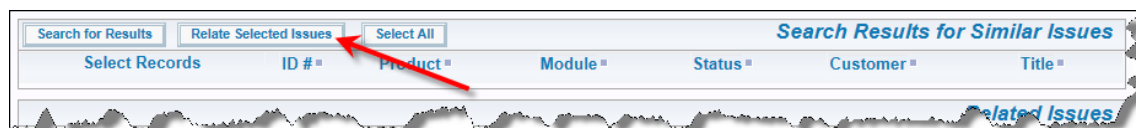
Submit button on a related issue display

RID RELATION TYPE	This cell attribute defines how the related records will be displayed within the RELATED_ISSUE_DISPLAY (or other layout where the name begins with RELATED). This has four possible values: 1. MEMBERS – This is the default value for this attribute if you do not override it with one of the other three possible values. In this case the RELATED_ISSUE_DISPLAY will show all the members of the group(s), containing the reference issue, i.e. the reference issue's parent's, children, and other issues not specifically related to the reference issues in the group(s) 2. CHILDREN – The related issue display will only contain issues that are the children of the reference issue 3. PARENTS – The related issue display will only contain issues that are the parents of the reference issue. Note that in the ExtraView data model it is possible to define more than one parent issue for a specific issue. See the section on Relationship Groups for more information 4. RELATED – In this instance ExtraView will display the parents and the children of the reference issue within the specified group(s), within the RELATED_ISSUE_DISPLAY. The parents and children of these related issues will not be displayed. You may create and use related issue display layouts as long as the name begins with RELATED. 5. SIBLINGS - ExtraView will display all the issues at the same level in the hierarchy. For example, if a parent record has five children, and you create a related issue display with all the children, using a relationship group relation type of SBLING, then you will see the four siblings to the current one. 6. LINKED - This type does not use a relationship type per se, but relies on the Related Issue Display filters to retrieve the associated issues. For example, you might have a filter that selects all the open issues for a particular product, and display these on a related issue display of each issue that you call up. This attribute may only be used on a cell containing one of the following: • the layout named LAYOUT.RELATED_ISSUE_DISPLAY or any other layout where the name begins with RELATED. For example, LAYOUT.RELATED_CUSTOMERS is a valid name for a relationship group name • the field named RELATIONSHIP_GROUP_PARENT, or • the field named RELATIONSHIP_GROUP_CHILD.
RID RELATIONSHIP NAME	It will be ignored on other cells. It refers to the name of the relationship group that is to be used for the field. In case a), it defines the relationship group(s) to search for the related issues; in cases b) and c), it defines the relationship group in which the new relation is inserted. In case a), the relationship group name may be the wild-card "*", which signifies ALL relationship groups. If you do not provide the attribute on a), ExtraView will search ALL relationship groups (as if the value were "*"); for b) and c), ExtraView will substitute the relationship group name stored in the behavior setting named RELATION_GROUP_DEFAULT, on the Workflow Settings screen. In any case, non-wild-card relationship group names must denote an existing relationship group.
RID SEARCH FILTER LAYOUT	This contains the name of the search filter layout that is used to determine the filters for an embedded layout of type *Search*. This is used as an attribute on the layout that will contain the results of the search, thereby

linking the search filters and search results together. This <u>example</u> shows how this attribute is configured

RID SINGLE_SELECT	This attribute is used on a search results layout. It is used to constrain the number of results that are selected by the user to a single one, so that the fields on the selected result are returned directly into fields on the current issue. Business rules transfer the results into the fields on the current issue. The example <u>here</u> shows how this is configured. Note the use of the inbuilt field named RELATIONSHIP_GROUP_CHOOSE_BTN field to provide a radio-button-like function to allow the user to select the result they want to transfer. If this attribute is not set then the RELATIONSHIP_GROUP_CHOOSE_BTN field allows multiple selections, and the use case is likely to be that the entire set of results will be used to create a group of related issues

In this attribute, you specify the sort order of the records presented on the RELATED_ISSUE_DISPLAY. Specifically, you can set up:

1. The field name to order by
2. The rank of this field in sort order (relative to other sort orders in this attribute)
3. The direction (ascending or descending) of the sort order

To specify ordering by more than one field, insert multiple Relationship Group Sort Order attributes. To specify where a sort order field appears in the sort priority, use the rank number. For example, two attributes may be:

RID SORT ORDER

Field Name	Rank / Direction
STATUS	1/ASC
PRIORITY	2/DESC

As usual, the sorting is done on SORT_SEQ within the field's definition, and then the title of the field value.

You can also optionally aggregate values while grouping together values within a field, and providing sub-totals or other attributes for each group. Set the **Group By** field to a value of Y to enable this feature. Now you will generate subtotals as well as a grand total. Clicking on a field title at the top of the related issue display will cause a refresh of the display, change the sort order and the sub-totals

RID THRESHOLD	This overrides the setting within the behavior setting named RID_PAGING_DEFAULTS. The number in this setting defines the number of issues that must exist within the related issue display before paging of the issues is performed.
RID TRANSPOSE LAYOUT	This attribute only works in conjunction with a related issue display. When this is applied to a related issue display, it causes a transposition of the rows and columns on the related issue display.

This has two purposes, depending on the field with which this is used:

- Most commonly, this will add a *SIZE=nnn* attribute to the HTML display tag for the field. The default is 11 characters if no attribute is set. **Example**: You may want to alter the width of a text box on the screen to be narrow. You would achieve this with an element attribute of *SIZE=30* or similar.
- If this attribute is added to the layout named LAYOUT.RELATED_ISSUE_DISPLAY or other layout where the name begins with RELATED, and this is embedded on an *add* or *edit* form, then the value represents the width of the embedded related issue display. Again, this is measured in characters. The default width of a RELATED_ISSUE_DISPLAY is 125 characters. **Example**: The screenshot shows where the SIZE attribute is used on the related issue display.

SIZE

STYLE

This is for indicating the font style of the element, possibly depending on other fields. This is supported within *add* and *edit* layouts only. It is supported when the field is in both read and write mode. If you want to add a style attribute to a layout element, you should use the style (rather than the html modifier attribute) otherwise it will be possible for the element to have two style attributes, which is invalid HTML syntax. It is likely that you will need a reasonable understanding of HTML tags and attributes to use this layout cell attribute.

Example: You may want to display a field on a report in the color red, if the *Priority* is equal to a value of *P1*.

Example: On the Add Issue screen, we want to make the ID field visible only once the issue has been added to the database, i.e. when the user arrives at the add verification screen. To achieve this, set the Attribute type

to Field Visible If, the field to ID, and the Equivalence to Is Not Null. This will have the effect of making only the ID field visible on the Add Issue verification screen.

SUPPRESS ALLOWED VALUES	This only applies to fields placed upon a search filter layout, such as the Quick Search, Full Search or Chart filter layouts. When this layout cell attribute is set, and the field is a child in an allowed value relationship, then the child list will not be filtered by its parent, and all possible values will be displayed. For this cell attribute to work, the parent field of the relationship must also be on the layout.
TEXT SEARCH OPERATOR	This attribute only applies to fields placed upon search filter layouts, such as the Quick Search, Full Search or Chart filter layouts. It allows the administrator to define whether a user is required to enter an exact match within a text field when it is used for a query operation, or whether the user's query is automatically set to use a *contains* operator. For example, consider a field named **Diagnosis** being placed on a search form. The values stored in issues within the database may contain a significant amount of text, and the word *hypertension* may appear in a subset of the issues. When the TEXT SEARCH OPERATOR is **contains**, all the issues that contain the word *hypertension* will be returned, but when the operator is set to =, only records that exactly match *hypertension* are returned. However, you might want to obtain an exact match when searching for a value such as the **ID** or **ALT_ID** of issues, necessitating the use of the = operator. The default is that the search uses the *contains* operator.
USER LIST DISPLAY	This attribute overrides the behavior setting USER_LIST_DISPLAY for fields with a display type of *user*. Valid values are POPUP and LIST, to display the user field using that format
VALIDATE HIDDEN	Validation between tabs allows required fields on a layout in ExtraView to be validated for existence, even if they are not visibly rendered on the currently visible screen. This most often happens when you have designed screen layouts with part of the total record being displayed with different tabs being selected. The need is to validate the existence of values within fields across the entire record, not just the fields visible under the currently displayed tab. This implies that the user will most probably be making entries under several tabs on a screen, before they submit the record. To set the attribute for the entire layout, go to the layout that embeds the subsidiary (embedded) layouts. Select the field named LAYOUT.embedded_layout_name. Add the Validate Hidden layout cell attribute to this field, and all fields within the layout will be checked. To set the attribute for any individual field(s) within the embedded layout, simply add the attribute to the field within the layout. If you place a VALIDATE_HIDDEN attribute on a layout cell that contains an embedded layout as opposed to a field, then all the fields on the embedded layout will be subject to the VALIDATE_HIDDEN logic. **Note**: There is an important interaction between Validate Hidden and fields that have a Visible If attribute. If a layout element has a VISIBLE_IF condition, and if the element has the VALIDATE_HIDDEN attribute, a validation between tabs must be performed. ExtraView will automatically perform this check. **Note**: If the Required If attribute is used within the same field as the Validate Hidden attribute, then it must be specified after the Validate Hidden attribute, so that ExtraView processes the attributes in the correct sequence.
VALUE TAG	This allows the administrator to add a new attribute to the HTML <TD … tag that surrounds the value of the cell. This can be used to provide a different style for the value, or to inject any other valid HTML into the cell value tag. It is likely that you will need a reasonable understanding of HTML tags and attributes to use this layout cell attribute. This tag is supported on *add*, *edit*, detailed report and Quicklist layouts.
VISIBLE IF	This is for indicating whether or not a field or embedded layout is seen depending on the value of another element, including itself. You may have more than one VISIBLE IF attribute on a field. If a field controls a hierarchy of VISIBLE IF dependencies, and it becomes invisible, all the fields below it in the dependency tree will also become invisible according to their conditions. Note that VISIBLE IF dependencies only work on fields with a display type of List. Further, they do not work if the field has been set in the data dictionary to be a multi-valued list, or if the field is a pop-up list. The VISIBLE IF attribute is only supported on layouts that are rendered within *add* or *edit* forms. Note that if you use the *greater than* or *less than* operators, they only work with numeric comparisons. If you place a VISIBLE IF attribute on a layout cell that contains an embedded layout as opposed to a field, then all the fields on the embedded layout will be made visible or not visible, according to the conditions set with the logic of the attribute. This is the simplest method to create several embedded layouts which are to be made visible or invisible according to the selected value in a driver field. Conditional attributes such as FIELD

REQUIRED and VISIBLE IF can be applied multiple times to a single field. When you use this functionality, the validation uses 'OR' logic. For example, you may set two VISIBLE IF rules, such as:

1. The field named action is visible if Status equals pending
2. The field named action is visible if Status equals fixed

With this rule, the action field will be visible if (a) or (b) is true.

When you use a VISIBLE IF layout cell attribute within a Related Issue Display layout, or a report layout, the field referred to as the condition for visibility must also be present on the same layout. Further, the VISIBLE_IF layout cell attribute only works on the EDIT_BUTTON, VIEW_BUTTON, DELETE_BUTTON and HISTORY_BUTTON within these layouts.

VISIBLE VALUES	This attribute allows you to display individual values within a field which contains a list of values, hiding all others. When you select this attribute, you are presented with a list of all the values within the field, and you select the values you want to make visible. This may be used with great effect on fields such as AREA on an *add* layout, to constrain a user to only be able to add new issues into specific business areas, and to lock the user out from other business areas.
	Note that if you configure this attribute, you may not then configure a FIELD HIDE VALUES attribute on the same field. This attribute is only supported on fields with a display type of List (including multi-value list, put not popup lists), Radio Button and Tab.
VISIBLE VALUES IF	This attribute is similar to VISIBLE VALUES, but adds the capability to only display the values based upon a logical expression based on other fields within the form.
UPDATE MAIN ISSUE	This attribute only affects the behavior of an EDIT_BUTTON field when it is placed on a related issue display. It has no function or purpose outside of this and will be ignored as an attribute on all other fields and layouts. The purpose of this attribute is to allow control of updates of the current issue on an *edit* screen when you edit a related issue by clicking on an EDIT_BUTTON in the related issue display. In such circumstances there may be business rules or other logic that triggers when you update the related issue; this logic may update the current issue. Under normal circumstances this would lead to a concurrency error when the user tries to update the current issue after completing the update to the related issue. In such an event, the desired behavior is often to allow the related issue to update the current issue, and allow the user to then update the current issue, but without a warning or error. To support this, there are three possible values for the layout cell attribute. These are applied in the following order: • When the user clicks on the EDIT_BUTTON on the related issue, the edit session for the related issue is set up in the current window, replacing the *edit* screen of the current issue. To achieve this, set the layout cell attribute to a value of **UPDATE_MAIN_ISSUE** • When the user clicks on the EDIT_BUTTON on the related issue, the edit session for the related issue is set up in a modal window on top of the *edit* screen of the current issue. To achieve this, the layout cell attribute named EDIT_ISSUES_IN_MODAL_WINDOW must be set to a value of YES. The layout cell attribute should be set to a value of **UPDATE MAIN ISSUE** • There is no layout cell attribute of **UPDATE MAIN ISSUE** set. This is the default behavior, and no check for concurrency errors is made. The *edit* screen for the related issue appears in a new window. Use this default setting if there is no interaction of rules on the related issue with the current issue.

Note: When comparing attributes, there are rules on how the tests are made. This affects the layout cell attributes that have IF as part of their name, e.g. VISIBLE IF and REQUIRED IF. The rules are:

1. For **equals** and **not equals** comparators, you can compare strings or numeric values
2. For > and < comparators, only numeric field display types can be compared, i.e. NUMBER, DECIMAL, and CURRENCY types

Note: When adding a layout cell attribute of the types VISIBLE IF, REQUIRED IF, NOT REQUIRED IF and similar, and you are operating on a field with a display type of *Checkbox*, and you want to act upon the unchecked value, use a setting such as VISIBLE IF *field_name* NOT EQUAL *Y*.

Layout Cell Attribute Usage

Layout cell attributes are only able to be configured where they will have an impact and have a specific function. This table shows the usage of the various layout cell attributes.

	ADD/EDIT LAYOUTS	REPORT LAYOUTS	SEARCH LAYOUTS	RID LAYOUTS	DATA DICTIONARY
ADD NEW	X				X
ALTERNATE TITLE	X	X	X	X	X
BUTTON ACTION					X
COLLAPSE REPEATING ROWS	X				
DISPLAY FORMAT	X	X	X	X	X
DO NOT USE ALLOWED VALUES			X		
DRILLDOWN		X			X
ENCRYPTED					X
EXPRESSION TEXT		X		X	X
FIELD AGGREGATE EXPRESSION				X	
FIELD AGGREGATE EXPRESSION TITLE	X	X			
HEIGHT	X	X		X	X
HIDE VALUES	X				
HIDE VALUES IF	X				
HIGHLIGHT VALUE IF	X				
HIGHLIGHTED IF	X	X			
HOVER VIEW DEFINITION	X	X		X	
HTML MODIFIER	X	X		X	X
LABEL TAG	X	X	X	X	X
LAYOUT SEARCH EXPANDED MODE	X				
MAXIMUM FRACTION DIGITS	X			X	X
MAXLENGTH	X		X		X
MODAL WINDOW				X	
NOT REQUIRED IF	X				
RANK					X
READONLY IF	X				
REMOVE ANY			X		
REMOVE LABEL NOWRAP	X	X	X	X	X
REMOVE VALUE NOWRAP	X	X	X	X	X
REPORT MAXIMUM FRACTION DIGITS		X		X	
REMOVE NONE	X		X		X

REQUIRED	X				
REQUIRED IF	X				
SELECTED	X				
SIZE	X	X	X	X	X
STYLE	X	X	X	X	X
TRANSPOSE LAYOUT					
USER LIST DISPLAY	X		X		X
VALIDATE HIDDEN	X				
VALUE TAG	X	X		X	X
VISIBLE IF	X				
VISIBLE VALUES	X	X		X	
VISIBLE VALUES IF	X				
VIEW ISSUE HOVER LAYOUT		X		X	
UPDATE MAIN ISSUE				X	
RID ACT ON CHANGE	X				
RID RELATIONSHIP NAME	X				
RID GROUP REFERENCE FIELD	X				
RID RELATION TYPE	X				
RID SORT ORDER	X				
RID FILTERS	X				
RID PAGE SIZE	X				
RID PREHEADER	X				
RID PREHEADER ADD BUTTON	X				
RID PREHEADER REFRESH BUTTON	X				
RID PREHEADER SUBMIT BUTTON	X				
RID LINK FIELD NAME	X				
RID SINGLE SELECT	X				
RID SEARCH FILTER LAYOUT	X				
RID THRESHOLD	X				
RID TRANSPOSE LAYOUT	X	X			
OMIT RG AUTO POPULATION	X				

The following attributes may only be defined within the data dictionary:

USED IN USER PROFILE

USED IN USER PROFILE SORT ORDER

BUTTON ACTION

RANK

ONCLICK JS

ENCRYPTED

Embedded Layouts

Layouts have a type, and many of these types may me embedded within each other. However, some such as the *Add*, *Edit*, *Quicklist* and *Detailed Report* layouts cannot be embedded in other layouts. Typically, new layout types that you create may be embedded within other layouts. For example, if you have many layouts to create that are slightly different, but a significant number of the fields share the same characteristics, you can define a layout and embed it within many other layouts, thereby sharing the configuration and minimizing the amount of work to configure your site. As well as embedding a layout within another layout, you can define alternate layouts that are rendered at run-time, according to a value of a field on the main layout. For example, you may want to embed different layouts with fields relevant to the collection of data about hardware and software, according to the user's choice of *Hardware, Software* or *Documentation*, in a field on the main layout. When the user selects this field, the screen is refreshed, and the appropriate embedded layout will be generated. There are two methods for doing this. First there is the method to embed layouts using the "Layout Visible If" technique. Second there is the "Selected By Layout" method. It is recommended that you use the first technique. The second technique is provided for backwards compatibility. Some general rules need to be observed with embedding layouts within one another:

- You must not place the same field on a layout, and within any layout that is embedded within it. Doing so would lead to unpredictable results and errors. As embedded layouts can be created independently of layouts, and they are interchangeable, this condition cannot be checked by ExtraView prior to execution of the rendering of the screens. When rendering the screens, if ExtraView detects that a field is used more than once over all the screen (including the embedded layouts), then a warning is generated
- You should not place the same field on different embedded layouts of the same type. Fields on embedded layouts are unique across the layout type being used
- It is recommended that any embedded layout that you place on a layout should span all columns of the layout. An embedded layout should never start in any but the first column of a layout. Other placement may work, but there is no guarantee that the layout will render correctly. This is due to limitations of HTML and how different browsers will render complex HTML. You should always test your embedded layouts in all the target browsers that your organization uses
- All the layouts of a single type must be contained within the same Business Area and Project for which the template for the embedded layout is created. For example, if you create a series of embedded layouts all dependent upon the same field for selection, these embedded layouts must all reside in the same Business Area and Project
- An embedded layout should not contain the parent field of an allowed value relationship, if the child field is contained in the outer layout. Both parent and child fields can be in the same layout or the parent can be in the outermost layout with the child in the embedded layout
- With the exception of Related Issue Display layouts, you cannot embed a layout within a *Quicklist* layout
- You cannot embed a layout within a *Related Issue Display* layout
- As a general rule, you should never embed a layout within itself, causing recursion or a condition which will be an infinite loop. ExtraView cannot detect this condition until runtime, and this can cause significant system and behavior problems requiring the restart of the application server.

Embedded Layout Types

- **Simple embedded layouts** - In this instance a layout type that you create is embedded inline within another layout
- **"Visible If" layouts** - Layout types that you create are all embedded within a layout. Each layout is initially hidden with a VISIBLE IF layout cell attribute. This attribute references a driving field, usually a tab or list field such that each of the embedded layouts is made visible one by one, according to the value of the tab or list field
- **Repeating row layouts** - These layout types must be embedded within another layout. You cannot embed another layout within a repeating row layout
- **Related issue layouts** - These layout types must be embedded within another layout. You cannot embed another layout within a related issue display layout.
- **"Selected By" layouts** - These embedded layouts are driven by a tab or list field. A template layout is created and this is used to create a different layout that will appear under each tab or list value. This is done with the LAYOUT SELECTED layout cell attribute. "Visible If" layouts are recommended and simpler to configure. "Selected By" layouts are provided for backwards compatibility only and you should use the **Visible If** layouts going forward for any new configuration.

"Visible If" Layouts

As an example, you may want three cells on a layout to be displayed according to the value selected in a list box. You may set this up with embedded layouts, or with the layout cell attribute named VISIBLE IF on each of the three cells. A guide to implementing the

most effective method is to consider the following:

- If you have several or many fields to make visible, embedded layouts are processed a little faster, resulting in higher performance for the end user
- Once set up, embedded layouts can be re-used on different layouts such as *add* and *edit* screen layouts.

To set up a "Visible If" layout and embed this within another layout, follow these steps:

- Use the **Create and Maintain Layout Types** administration utility to create a new layout type. The usage of the layout should be *Screen*
- Navigate to the **Design Center**, select the user role, business area and project where you will use the embedded layout, and the use the list with the label *Add a new layout for the entire system* to open up a new layout of the type you created in the previous step
- Place the fields onto this new layout within the Design Center. This will become the embedded layout. Save your layout
- Edit the layout (or layouts) within which you wish to embed the layout. If necessary create space for the layout you are going to embed, by creating a blank row. Typically you will also place the layout beneath its driver field. The driver field is the field that will be used to select and make visible, the embedded layout, dependent upon its value
- From the list of layouts in the Design Center, locate the layout you are going to embed, and drag it to its position on the main layout. Extend its width across the main layout, as required
- Now, add a layout cell attribute to the embedded layout. This will be an attribute of the form **VISIBLE IF**. Select the field and its value which will be used to trigger the visibility of the embedded layout
- If you want to embed further layouts, each triggered off a different value in the driver field, you can create a layout type for each of the values of the driver field, embed the layout within the main layout and then use the **VISIBLE IF** attribute to trigger its visbibility based on different values of the driver field.

"Selected By" Layouts

This feature allows the administrator to design a series of layouts, which are generated within a global layout such as the *Add* or *Edit Issue* screen. Each of these layouts is based upon the value of a trigger field. For example, you may want to generate a series of fields within a layout, dependent on the category of issue you are adding. Thus, if you want to render a different embedded layout for each of *Software, Hardware* and *Documentation*, which are different values of UDF named **ISSUE_TYPE**, with a title of **Issue Type**, then you will choose the **Issue Type** field. When the user selects the value of Issue Type, the embedded layout is refreshed, displaying the layout that is tied to the value of the trigger field. One usability feature is that if you have a form within a browser window that is longer than the height of the window, and you choose a different value for the trigger field, ExtraView will place the new embedded layout to display at the top of the screen, thereby preventing the situation where the new layout may be hidden beneath the bottom of the window. The sequence of administrative operations required to create selectable embedded layouts is as follows:

1. First, decide on the field that is to be used to trigger the rendering of the different embedded layouts. For example, if you want to render a different embedded layout for each of *Software, Hardware* and *Documentation*, which are different values of the **Issue Type** field, then you will choose the **Issue Type** field.
2. From the Administration Site Configuration tab, select **Create and Maintain Layout Types** feature. The following screen appears:

Layout Types

3. Press the button to add a new layout type to ExtraView. You must name this layout with the fixed name of the trigger field. In our example, this is **ISSUE_TYPE**. The title should be **Issue Type** and the type must be **Screen**

4. Now navigate to the **Design Center**

5. Navigate to the Business Area and if appropriate, the Project within which you want to use the embedded layouts

6. Next, select the layout type you just created from the **Add a new layout for the entire system** select list. Once again, in our example, this is named **Issue Type**

7. Give this new layout a title and a description. This layout is only used as a reference to create the different embedded layouts. You do not need to add any fields to this layout. Simply **Save** this layout. This now serves as a template for each embedded layout you create. Any fields that are contained in this layout will be propagated to the selectable embedded layouts created in the remaining steps of this procedure

8. **Return** to the initial **Design Center** screen

9. If you are adding the embedded layouts to the Add screen, edit the appropriate *add* screen. Place the **Issue Type** field on the maing grid of the layout, if it is not already there

10. Now place the embedded layout on the main grid, by dragging it from the list of Layouts. In our example, the layout is named **LAYOUT.ISSUE_TYPE**. You should be sure to give the embedded layout a suitable *colspan* within the layout. ExtraView will suggest that it can do this for you at this step, creating an embedded layout that will span the width of the entire layout

11. Step 12 will be repeated for each value of the **Issue Type** field (in our example, *Software, Hardware* and *Documentation*). We will be creating and saving an embedded layout for each of these values, from the template created in step 7

12. Make sure the embedded layout is selected on the main grid and click on the **New** layout cell attribute button. Add a layout cell attribute of LAYOUT SELECTED, for the field named **Issue Type** and then select the appropriate value for each required embedded layout (often there is an embedded layout for every value of **Issue Type**, but that is not necessary). For example, if you want an embedded layout for the *Software* value of **Issue Type**, choose *Software* and Return to the layout

13. **Save** the layout and **Return**. You should see the new layout in the list of layouts, with the **Select by field** containing the title of **Issue Type**, and the **Select by Value** containing the values that were specified in the *SELECTED* attributes

14. You may repeat steps 12 and 13 for other values as your needs change (or you add values to the selector UDF) in the future. From the Design Center you can directly proceed to editing the layout cell attribute of the **LAYOUT.ISSUE_TYPE** field, add the selected value, and then **Save** the layout. The new embedded layout will be automatically created for each new value added to the LAYOUT SELECTED attributes

15. You can delete selectable embedded layouts as you would delete any other layout. However, you should not delete the "template" layout created in step 7, or you will not be able to add new selectable embedded layouts of that type.

Positioning Embedded Layouts

Within ExtraView, a design decision was made to always align the columns of an embedded layout with the columns of the layout that embeds it. This provides the most pleasing presentation of fields to a user. However, the limitations of HTML are such that this works precisely when an embedded layout is defined to start in the first column of the layout within which it is created. If the embedded layout has multiple rows (and most do), then the second and subsequent rows will always be rendered from the first column of the second and subsequent row, which may cause a misalignment. To prevent this, set a *rowspan* on the field at the left of the embedded layout, such that it spans one row greater than the number of rows in the embedded layout. To see this effect, see the following two layouts and the differing rendering of the results:

Layout 1

Layout definition with embedded layout

Note that this layout leads to the wrong rendering of the embedded layout, where the embedded layout titles appear correctly, but the data rows are offset to the leftmost column.

Rendering of layout

Layout 2

Layout definition with padded field in first column

In this layout, not that the OWNER field was set to have a *rowspan* of 2. This corrects the position of the embedded layout rows so they align correctly.

Rendering of layout

Repeating Row Layouts

The basic principal of the repeating row layout is that it provides a repeating block of fields, where each row in the block can be added by the user at runtime, or added by business rules, to add additional field values on the screen. For example, you may want to have a row of fields that refer to a sub-issue, assigning part of the issue to an individual. You can create as many rows as needed, to assign different parts of the issue to different people. Each of these may have a field named RELEASE_STATUS, which can be used to follow the workflow you create.

Note: All layout types to be used as repeating row layouts within screen forms must be created as a repeating row layout type. All fields placed on repeating row layouts must have been created in the data dictionary with the **Field Belongs To** and must have the attribute of *Repeating Row records.*

You may embed repeating row layouts within an ADD_PROBLEM, EDIT_PROBLEM, DETAILED_REPORT, EMAIL_FULL, or HISTORY layout. You may also embed a repeating row layout within another layout which is itself to be embedded within one of these layouts.

You can also create UDF's that are attached to the repeating records, as opposed to being added to the main issue record. Care should be taken when doing this. For example, it is problematic for users and for reporting to add field display types such as Log Area to each line of a repeating record. Log Area fields are therefore not supported on repeating record layouts. The section on the Data Dictionary describes how to set up UDF's, including UDF's attached to repeating records.

Note: There is also an important limitation in that you may not place the same field on multiple repeating row layouts. ExtraView cannot differentiate between these fields and incorrect data would be stored if you configured a system that had the same field appear within different repeating row layouts. The design center cannot distinguish this when you are building each repeating row layout, so take care to only use repeating row fields within a single repeating row layout.

At the same time as this limitation, note that if you move a field defined as a repeating row field from within one repeating row layout, to a different layout, the data entered by end users does not follow the field. The data is stored with a connection to the repeating row layout as well as a connection to the field. You may move the field, but data previously associated with the field will not be visible. It is a better practice on a system that already contains data, to create a new field for the new layout, and remove the field from the old layout. In this way any historic values accumulated for the field will still be available.

Permission Keys

The following keys exist for both **PR_ADD_PROBLEM** and **PR_RESOLUTION**, having their effect on the *Add* screen and the *Edit* screen respectively.

Key	Purpose
RELEASE	This permission key controls access to all the repeating row layouts that appear within the *Add* screen and the *Edit* screens. Giving this key read and write permission for a role enables all the repeating row layouts that appear
ALLOW_ADD_REPEAT_RECORDS	This controls the visibility of the *Add Another xxx* button on the screen. The *xxx* text is determined by the title to the data dictionary field named **RELEASE**

Creating and Using a Repeating Row Layout

1. Each repeating row that appears on a screen to an end user must have a layout type defined, and then the defined layout must be completed with fields. Finally, the defined layout is embedded on the layout which appears to the end user
2. Define the Layout type
    1. First, create the layout type for the repeating row. Click on **Create and Maintain Layout Types** from the Administration menu, under the **Site Configuration** tab
    2. Complete the information required for the layout type, including the repeating row section. When creating a new repeating row type, it is usually convenient to choose the same name and title as the main layout type

Creating a Repeating Row Layout

Note that when you create a layout type, a field with the same name is created within the **Label** of the data dictionary. If you need to alter the title of the repeating row display that appears on **Report** usage layouts, update this within the data dictionary

3. Place fields onto the layout type
    1. You may either create a new repeating row using the layout you just created, or utilize an existing one. Use the **Add a new layout for the entire system** entry within the Design Center to add the new layout type to your current business area, project and role
    2. Add fields to the layout, using the layout editor features. An example might be:

Add Release Layout

The inbuilt fields that can be added are taken from the following list. Note that these fields may only be placed on a single repeating row layout.

     a. RELEASE_FOUND
     b. RELEASE_FIXED
     c. RELEASE_ASSIGNED_TO
     d. RELEASE_OWNER
     e. RELEASE_PRIORITY

       f. RELEASE_SEVERITY_LEVEL

       g. RELEASE_STATUS – note that this field obeys the workflow rules that you set up with the workflow status rules

       h. RELEASE_OWNER

       i. Any UDF that you have created, which is of the repeating record type. Note that a UDF can only be placed on a single repeating row layout, while the above fields can be placed on any number of repeating row layouts.

5. Add the Repeating Record Layout to Appropriate Screens
   1. Navigate to the Design Center
   2. Select the repeating row layout from the list of layoutsby clicking on its name
   3. Drag the layout to the place within the layout where you would like to see the fields
   4. Save your changes and view your results (the repeating row layout is embedded within a layout).
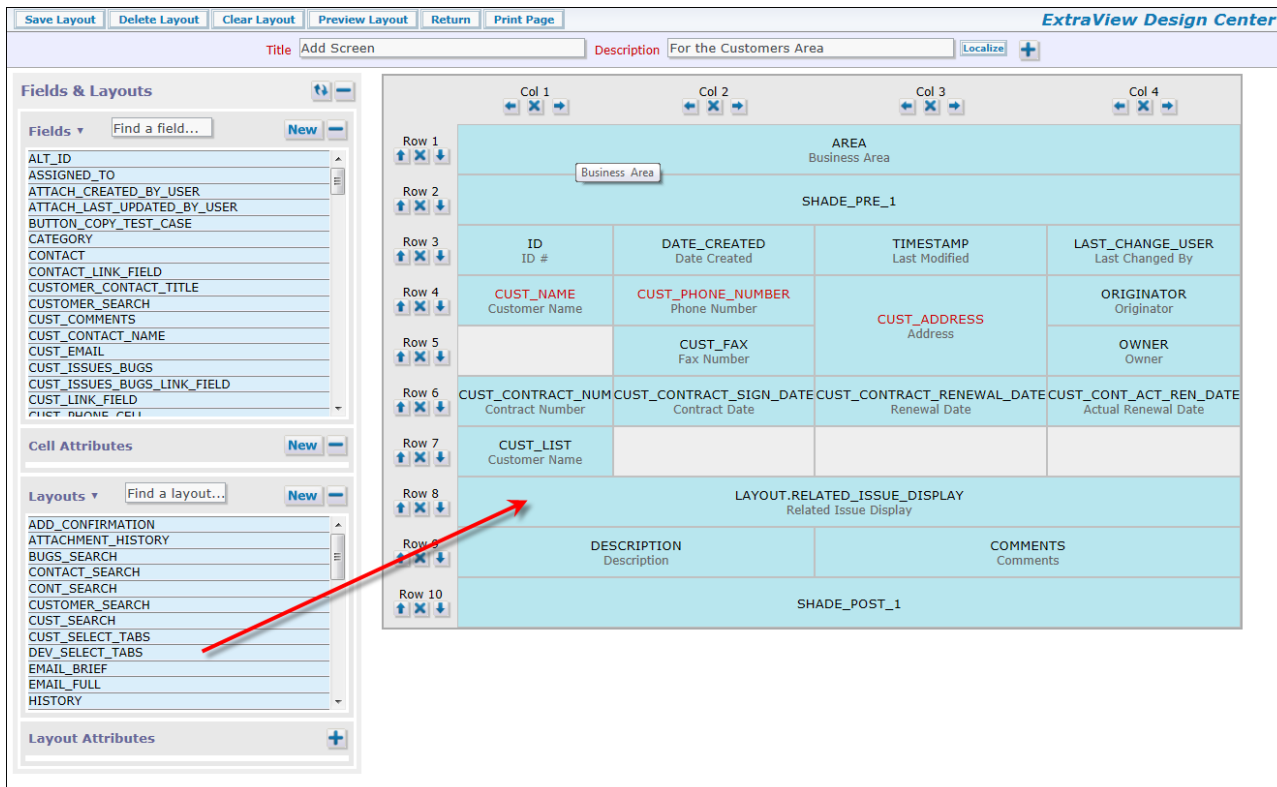
Design Center screen

The Inbuilt RELEASE Repeating Record

Repeating records such as the inbuilt *Release* layout are not designed to be rendered as individual screens, but designed to be embedded within other layouts. This allows you to create a system that either allows you to have issues that can, for example, have multiple release records within each issue. Different layouts can be designed that can be embedded within both the Add and Edit screens if desired.

You should be aware that you should not create recursion by embedding a layout within itself. Note that if you require a system where you require a single repeating row record (e.g. there is only one release found and one release fixed for each issue) then you should not use a repeating row record.

The inbuilt repeating record named **RELEASE** contains fields with the following names and default screen titles:

RELEASE_FOUND	Version Found
RELEASE_FIXED	Version Fixed
RELEASE_SEVERITY_LEVEL	Severity Level
RELEASE_PRIORITY	Priority
RELEASE_STATUS	Status
RELEASE_OWNER	Owner
RELEASE_ASSIGNED_TO	Assigned To
RELEASE_RESOLUTION	Resolution

Each of these fields is protected with a security permission key, one for the *add* screen and one for the *edit* and *query* screens. These are manipulated in the standard way, as explained in the section named **Grant Security Permissions**.

There is a security permission key that allows you to control the presence or absence of a delete checkbox on the repeating row. This is named PR_RESOLUTION.PROBLEM_RELEASE_DELETE. This key controls the presence of a check box on a repeating row record, that allows the user to delete a complete repeating row. Write access is needed to control the checkbox.

Note: There are two security permission keys, named PR_ADD_PROBLEM.RELEASE and PR_RESOLUTION.RELEASE. These keys control the presence of the entire repeating record, and switch the entire structure on or off for an individual user role. To turn on or off access to the whole repeating record, use the *write* privilege.

Note: There are two security permission keys named PR_ADD_PROBLEM.ALLOW_ADD_REPEAT_RECORDS and PR_RESOLUTION.ALLOW_ADD_REPEAT_RECORDS that control the visibility of the **Add another xxxxxx** button beneath the repeating rows on *add* and *edit* screen layouts. You can remove the button to stop users in some roles being able to add additional repeating rows, but have users in different roles being allowed to add the repeating rows. The **xxxxxx** is the title of the data dictionary field named RELEASE. For example, if you provide a title of **issue** to the field, then the button is named **Add another issue**.

Unique and non-Unique Repeating Rows

Differing purposes may require you to implement repeating row records which either have or do not have a unique key field. There are two parts to configuring this capability. First, in the Workflow behavior setting screen, use the ENFORCE_UNIQUE_RELEASES setting to switch on the capability of enforcing a unique key field. Second, you must place the field named RELEASE_FOUND on the repeating row record. This normally has a display type of LIST, although TAB and POPUP are also supported. You can re-title the field in the data dictionary to any suitable title you require for your installation. Also, note that the RELEASE_FOUND and the RELEASE_FIXED fields are the child fields where the PRODUCT_NAME field is the parent. This means that if you have the RELEASE_FOUND or RELEASE_FIXED fields on a repeating row record, then you must have the parent PRODUCT_NAME field on the add and edit screens that contain the repeating row record.

Repeating Rows & Workflow Status Change Rules

The ExtraView Status Change Rules features work with the inbuilt RELEASE_STATUS field. The RELEASE_STATUS field is treated the same as the inbuilt STATUS field with Status Change Rules. One aspect of this is that you may have configured the STATUS field to use Status Change Rules, but you are not placing the RELEASE_STATUS field on the RELEASE layout described on this page. In this case, ExtraView still needs to be able to handle the *** None *** value for the missing RELEASE_STATUS field. To accommodate this, configure the Status Change Rules to accept a *** None *** value, i.e. it is a legitimate update to allow the status of the issue to be changed to a null value, for the appropriate range of Business Areas, Projects and Roles that are configured.

Repeating Rows & Custom Code

There are occasions when working with repeating rows that it is useful to know which row you are targeting with an event. For example, you may have a popup field that returns a value to a specific field on a repeating row. You will need to know which row to target with the return value. To make this task simpler, there is a JavaScript function available on *add* and *edit* layouts that you can call from an HTML modifier to any cell. This function is named getRowNumber and is used in this way:

var rowNum = getRowNumber(this, this.form);

This function returns the following in rowNum:

-1 if the field is a singleton repeating row
0 – nn if the field is in a repeating row with 2 or more rows, where nn indicates which row the field is in.

Field Limitations on Repeating Row Layouts

Repeating row layouts do not support –

- Log Area display type fields
- Multi-value display type fields
- Any data dictionary field that does not have the **Field belongs to** value set to **Repeating Records**
- Any one field cannot be placed on multiple repeating row layouts, whether it is an inbuilt or user defined field. Take care when configuring your layouts as no error can be detected at design time
- HTML Area display type fields are not supported on repeating rows.

A Note on how Repeating Rows are Managed

As far as possible, repeating rows are added and updated on an *add* or *edit* screen using Ajax. This provides the best performance possible. There are some cases where ExtraView cannot use Ajax, and in this case all the repeating rows have to be re-rendered upon a change. This usually occurs if there are field differences between what is rendered on the page and what will be on the newly added record.

For example, suppose you have a FIELD/LAYOUT VISIBLE IF layout cell attribute on a field and the parent field is also on the same repeating row. It is possible that you can make the field invisible on all the rows. When this happens, ExtraView does not display the title for that field and suppress the entire column. If adding a new row makes the invisible column visible, then ExtraView needs to re-draw the entire set of repeating rows.

The following cases show where ExtraView forces a complete rendering of the repeating row layout, as opposed to an Ajax refresh of just a single row:

- The repeating row layout is placed under an embedded layout as the first row in the embedded layout. In this case, there is not a row with the ID of the repeating row in the browser DOM when ExtraView renders the upper-level sublayout. Hence the complete repeating row layout must be rendered.
- The Ajax refresh causes a column to be added or removed from the display. For example, suppose you have a **Visible If** condition on FIELD_A and you have 2 rows on the repeating row layout and FIELD_A is visible on both rows. FIELD_A is on a column by itself within the repeating row layout. Change the Parent value on row 1 to make FIELD_A invisible. The Ajax call takes place and makes the field invisible. Change the Parent value on row 2 to make FIELD_A invisible on row 2. This Ajax call will force the rendering of the entire repeating row. ExtraView does this because the column for FIELD_A will be suppressed if FIELD_A is invisible on all rows.

Relationship Layouts

Relationship groups allow the users of ExtraView to associate individual issues with each other. You may want to create relationships between issues for several reasons, for example, because you want to consider them together when making updates, preparing reports, or because you want to implement a workflow which mandates that you update issues as a group, or because you want a workflow where the parent issue cannot be closed until all the child issues are handled.

Relationship groups are configured with a layout type that is then embedded within an *add*, *edit*, *report* layout or another embedded layout. This layout is configured with layout cell attributes to identify the basic characteristics of the related issue display. You might need to provide some business rules to control the interaction of the related issues. For reporting with related issues you will need to define the reporting hierarchy. Examples of relationship groups:

- A project may be composed of many tasks. The project can be stored in an issue, and each of the tasks may be stored in related issues
- A product release may be supported by many individual features. The product release may be stored in an issue while the features may be stored in related issues
- A reported incident may have a number of corrective actions that must be applied. The incident may be stored in a parent issue and the corrective actions may be stored in related issues.

Relationship groups can be built with many different types of behavior. Examples of how to build installations with related issues at their core is described later in this section of this guide. There are several types of inbuilt relationship groups that can be defined:

Relationship Group Type	Purpose
Bi-Level	This is a simple relationship group where a single issue acts as a parent issue to any number of child issues. This has largely been superseded by the One-to-Many type, but its use will be continued for backwards compatibility. If you are designing a new ExtraView application, or adding relationship groups to an existing installation, it is recommended that you do not use the bi-level groups. To continue using this type of relationship group and to ensure that bi-level groups are updated, the behavior setting named RG_UPDATE_BILEVEL_ONLY should be set to a value of YES. To remove the ability to create new relationship groups with the type of bi-level, set the behavior setting named ALLOW_BILEVEL_GROUPS to a value of NO. Bi-level relationship groups will not operate correctly unless this setting has a value of YES.
One-to-Many	This type also has a single parent issue with multiple child issues, but with some differences and enhancements from the bi-level type of group. This type of relationship group should be used in preference to the bi-level group type for all new applications. If you want the capability to update all related issues when performing the update of an issue, you must set the behavior setting named RG_UPDATE_BILEVEL_ONLY to NO.
One-to-Many Cascade	This type is similar to the One-to-Many relationship group type. The difference is that when a user deletes an issue which is the parent of one or more child issues, then the child issues are also deleted, as opposed to leaving orphaned child issues in the database, once the parent is deleted.

Many-to-Many	In the many-to-many relationship group type, an issue can belong to more than one relationship group, and many issues can be connected in many ways. . If you want the capability to update all related issues when performing the update of an issue, you must set the behavior setting named RG_UPDATE_BILEVEL_ONLY to NO.

Relationship groups go beyond the grouping of issues using report filters. For example, it is always straightforward to produce a report that groups issues where there is a common searchable field, such as all the issues that affect a specific product or have the same priority. Relationship groups share a number of common characteristics:

- You can relate different items or issues together. Users achieve this through the *add* or *edit* screens. The administrator may also perform this from the **Relationship Group Maintenance** screen
- When you update a member of a related issue group, you can configure behavior that will optionally allow the user to update all the related members of the group. You can create a layout that contains the fields that may be updated in all issues when one member of the group is updated
- Users may remove items from groups via the *add* or *edit* screens. Again, the administrator may also perform this from the **Relationship Group Maintenance** screen
- You may view all the items in the relationship group, from the *add*, the *edit* screen or from reports such as a Quicklist or a Detailed report. From any of these places and with permission, you may drill down and view or edit the issues
- With permission, when you update an issue that is a member of a relationship group you may apply updates to the other members of the group
- When issues are cloned, rules can be used to place these issues in a relationship group.

Relationship Types

Simple Relationship

The simplest relationship may be that you want to connect two issues together, for example because they exhibit similar symptoms, or because they were reported by the same customer. This can be represented by the following diagram:
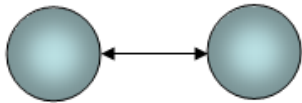
Parent-Child Relationship

Typically, a parent – child relationship group will be implemented with the One-to-Many group type. The most common use cases are where you want a single issue to control the behavior of the child issues, for example, several customer-reported issues may not be closed until the engineering issue that controls the fix to the problem has been solved and closed. In this case the engineering issue would be the controlling parent issue.
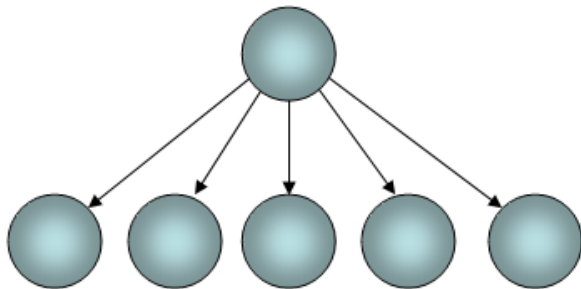
Sibling Relationships

Sibling relationships link issues at the same level in a hierarchy. Typically sibling issues will have a single parent to which they are linked. Each of the siblings may in turn have child issues. A common use case is where you want to perform an action on the parent issue, when all siblings have a common value in a field and you want to reflect this within the parent issue. For example, you might want to automatically set the status of the STATUS field within a parent issue to **Approved** when each of the child issues has a radio button that indicates approval of that child issue. A SIBLINGS relationship allows you to compose a rule that would take action on the parent when all the siblings have the same value indicating that approval is given for that individual sibling.
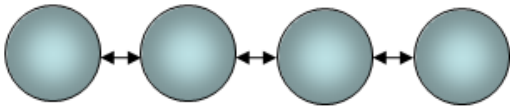
Complex or Arbitrary Relationships

The Many-to-Many relationship group type allows a very flexible structure of relating issues, where you can connect issues in almost any form, and a single issue can be a member of more than one group. The use case here is to allow a very freeform structure where different groups of issues can be connected together. Features exist to allow you to look up and down the structure, to the parent or parents of an issue, to an issue's children, or even to look beyond the direct relationship and include the related issues to the parents or children of the issue. This is represented by the following diagram:
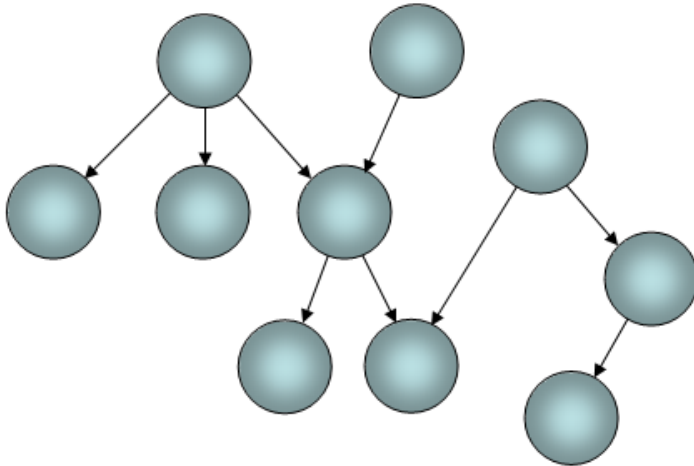
Security Keys for Relationship Groups

Key	Purpose
CF_RELATIONSHIP_GROUP_VIEW_ PROBLEM_BUTTON	Controls the appearance of the View button on the Related Issues Screen. Only used for BI-LEVEL relationship groups.
PR_ADD_PROBLEM.RELATIONSHIP_GROUP	Controls access to the RELATIONSHIP_GROUP button on the *add* screen.
PR_RESOLUTION.RELATIONSHIP_GROUP	Controls access to the Related Issues (the button beside ID on the Edit screen) for BI-LEVEL groups, and to see the RELATIONSHIP_GROUP field on the *edit* screen
PR_RESOLUTION.RELATIONSHIP_ GRP_ADMIN	Controls the visibility of the Manage Relationship Group button on the *edit* screen for BI-LEVEL relationship groups.
CF_RELATIONSHIP_GROUP	Controls the visibility of the Group Issues button on reports (this is next to the Refresh button at the top and bottom of column, summary, Quicklist and Detailed reports

Behavior Settings for Relationship Groups

Key	Purpose
CLONE_RELATION_GROUP	Relationship group name for clone relationships. When an issue is cloned it will automatically be placed into the relationship group named in this setting.
MORE_HISTORY_NUMBER_ROWS	This setting defines the number of rows of related issues to display, before a *more* ... prompt appears within the history record. This is to prevent the situation where you may have hundreds of related issues, and many updates. In this case, the user would mainly see history of the related issues, as opposed to seeing the history of the issue itself.
RELATION_GROUP_DEFAULT	The name of the relationship group that will be used for any issue being placed into a relationship group. This default can be overridden by using a layout cell attribute on the RELATED_ISSUE_DISPLAY, or by using the field

	RELATIONSHIP_GROUP on the *add* or *edit* layout to offer the user a choice of relationship groups
RELATIONSHIP_GROUP_EMAIL_LIMIT	When you update the status of an issue within a relationship group, each issue will be subject to the standard notification process. When there are more than RELATIONSHIP_GROUP_EMAIL_LIMIT entries in the group, notification is only made to the users in the parent issue, and a comment is inserted on the issue with this information
RELATIONSHIP_GROUP_LIST_UDF	This setting is the name of a user defined field with a display type of list. This UDF is used to maintain a list of the names of relationship group for related issue display. The UDF is maintained by ExtraView as relationship groups are created or deleted. The UDF can also be aliased by other UDF's to provide an unlimited number of fields that can be used to identify the relationship group names to support multiple related issue displays on a single edit screen.
RELATIONSHIP_GROUP_MAX_DISPLAY	Maximum number of issues to display on the relationship group screen. 0 means there is no limit. Note that the administration user defined in ADMIN_OVERRIDE_ROLE will always see all the issues, no matter the value of the setting
RELATIONSHIP_LINK_DISPLAY	If the data dictionary entry RELATIONSHIP_GROUP_ LINK is placed on the edit screen layout then either a link or list will appear beside the title. The display of the field is controlled by the behavior setting named RELATIONSHIP_LINK_DISPLAY. This can have a value of either SELECT or LINK. The RELATIONSHIP_GROUP_LINK entry should not be placed on the add screen. If the value is LINK then the field will appear on the edit screen as a link specifying the number of issues in the group. If this link is clicked on, the user is redirected to the Related Issues screen. If the value is SELECT, then the field will appear as a select list. The user can then select the issue they want to modify in the list and then press the link button to go to an edit screen for that issue. If there are more entries in the list than the POPUP_LIST_SIZE behavior setting, then an additional entry of * More * will appear in the list. If the user selects this value and clicks on the link button, then they will be redirected to the Related Issues screen for a full list of the grouped issues.
RG_UPDATE_BILEVEL_ONLY	This behavior setting is for backwards compatibility with ExtraView version 4.x. If you set the value to YES, then related issue updates are only triggered when a change to the STATUS field takes place. If this is set to NO then updates to related issues may be triggered on different events and many updates to different fields are possible.
RID_PAGING_DEFAULTS	Related issue displays may be paged. Currently, an entire RID is shown on the *add* and *edit* screens. Page size and threshold values determine how many issues are shown simultaneously in the related issue display, with other pages navigable through via **Next** and **Previous** buttons at the bottom of the display
SHOW_CLOSED_REL_GROUPS_PERIOD	The list of items on a relationship group will be displayed until all issues have been closed for a minimum number of days specified by this value. Valid values are numbers equal to or greater than 0. 0 means that as soon as all items within a relationship group are of the status specified by STATUS_CLOSED_NAME they will not be displayed on the list of relationship groups.

Data Dictionary Fields for Relationship Groups

Field	Purpose
NO_RELATED_ITEMS_MESSAGE	This is the message that is provided within a related issue display when a search is performed to return related issues, but none exist. The default is **There are no search results**. This message is a label field within the data dictionary and it may be altered and it may localized
RELATED_ISSUE_DISPLAY	This label field provides a means of being able to alter the title to the inbuilt relationship issue display. Note that when you create a new relationship issue display type, then a new field is automatically created as a label field in the data dictionary, also allowing you to provide a title to the related issue display you create

RELATIONSHIP_GROUP_CHILD	When this field is placed on a layout, and the user enters an ID into this field, then the issue whose ID is entered becomes a child to the current issue when the current issue is updated. If there is an error, such as a non-existent child ID, an error will be generated and displayed to the user. If the current issue is already a parent of the child issue ID entered, no error is indicated and no change to the relationship occurs. When the user next views the issue, the related issue display will show the new child issue, assuming the relationship group relation type is either MEMBER, CHILD or RELATED
RELATIONSHIP_GROUP_LINK	A general purpose label to provide the title for relationship group links
RELATIONSHIP_GROUP_PARENT	When this field is placed on a layout, and the user enters an ID into this field, then the issue whose ID is entered becomes a parent of the current issue when the current issue is updated. If there is an error, such as a non-existent parent ID, an error will be generated and displayed to the user. If the current issue is already a child of the parent issue ID entered, no error is indicated and no change to the relationship occurs. When the user next views the current issue, the related issue display will show the new parent issue, assuming the relationship group relation type is either MEMBER, PARENT or RELATED
RELATIONSHIP_GROUP_REMOVE_BTN	A field to provide a checkbox that allows issues to be removed from relationship groups. The field is an inbuilt field with special treatment. It is defined in the data dictionary with a type of LABEL and a display type of BUTTON. Although it is placed on related issue display layouts, it will only appear when the layout is rendered within an *edit* screen. It is ignored on add screens, report layouts and on the POST_EDIT_UPDATE layout. There is a security permission key named PR_RESOLUTION.RELATIONSHIP_GROUP_REMOVE_BTN to control the appearance of the Remove? checkboxes on the edit screens. The user may check any number of related issues and the checked issues will be removed from the relationship when the current issue is updated
RELATIONSHIP_GROUP_TITLE	A general purpose label to provide titles for relationship group
RELATIONSHIP_GROUP_TYPE	A general purpose label to provide the titles for relationship group types
RELATIONSHIP_GRP_PARENT_ID	A field that stores the parent ID of a relationship group
RELATIONSHIP_GROUP_ID	A field that contains the internal ID of a relationship group
RELATIONSHIP_GROUP_PARENT	A field that contains the parent of a relationship group
RELATIONSHIP_GROUP_CHILD	A field that contains the child of a relationship group
RELATIONSHIP_GROUP_TXNS	A field that contains the historic transactions of a relationship group. This field is only valid on a layout of type HISTORY.

Related Issue Layouts

Related issues are displayed with layouts that use the naming convention **RELATED_XXXXX**. To create a new related issue layout, you first create a layout type where the name begins with **RELATED_** Note that the usage of related issue displays must be *Report*. In the normal way for layouts, these may be inherited by any Business Area or Project which will use them. These are used to display the related records within *add, edit* and report layouts. The layout contains the fields you want to display when viewing the related issues. There is a single default related issue display layout within a new ExtraView installation, named **RELATED_ISSUE_DISPLAY**. The default fields on this layout are VIEW_BUTTON, QUICKEDIT_BUTTON, EDIT_BUTTON, RELATIONSHIP_GROUP_REMOVE_BTN, ID, STATUS, ASSIGNED_TO and SHORT_DESCR. You may change any of these fields to meet your requirements.

When you configure a related issue display layout to allow the creation of new related issues, you are able to place a button on a header to the display. This button may have any text, and when pressed will open a modal window where you can add the issue you are creating into the relationship with the current issue on the screen. On reports all related issues are displayed, allowing you to print the results clearly.

On *add* and *edit* screens, the related issues are placed within a scroll box whose dimensions are controlled by the layout cell attributes named SIZE and HEIGHT. If you do not provide a layout cell attribute for SIZE, then the default is 125 characters. Within the layout, you can also use a SIZE layout cell attribute on the individual fields to control their width. If you do not use a SIZE attribute on

individual fields, then the width of each field floats and the user's browser will decide on the optimal width. The height of the related issue display is calculated automatically, to display all the records in the group unless a HEIGHT layout cell attribute is added to the related issue display, to provide a maximum height.

You can fix the position of the column titles of the related issue display with the layout cell attribute named RID FIX TITLES. When you set this attribute it is not recommended that you use the SIZE attribute on fields within the related issue display layout.

Sample Rendering of Related Issue Display Layouts

Some sample layouts rendered on *Add* and *Edit* screens are:

Related issues on a report

Editing related issues on an add or edit screen

Adding new related issues on an add or edit screen

The modal popup to add a related issue

Related Issue Display Layouts

You may create any number of additional related issue display layouts, but you must begin the name of the layouts you create with RELATED_. For example, as well as RELATED_ISSUE_DISPLAY, and as an example, you may create another related issue layout named RELATED_SUPPLIERS. All the layouts to be used as related issue displays must be created with a display type of **Report**, even if they are to be embedded within an *Add* or *Edit* layout. Once you have created the layout type, you then include the layout on the *Add*, *Edit* or Detailed Report layouts.

For the above example, you would include LAYOUT.RELATED_SUPPLIERS on the layout. When you create a related issue layout, a new field is created automatically in the data dictionary, with the same name as your layout. In our example, the field RELATED_SUPPLIERS is created. The field has the usual PR_ADD_PROBLEM and PR_RESOLUTION permission keys, and these control visibility to the the related issue display layout on the *add* and *edit* layouts for the different roles in your installation.

The field RELATIONSHIP_GROUP_REMOVE_BTN is an inbuilt field with special treatment. It is defined in the data dictionary with a type of LABEL and a display type of BUTTON. Although it is placed on RELATED_ISSUE_DISPLAY layouts, it will only appear when the layout is rendered within an *Add* or *Edit* screen. It is ignored on report layouts and on the POST_EDIT_UPDATE layout. There is a security permission key named PR_RESOLUTION.RELATIONSHIP_GROUP_REMOVE_BTN to control the appearance of the **Remove?** checkboxes on the *Add* and *Edit* screens. Its function is to allow the end user to remove a related issue from the relationship group. The user may check any number of related issues and the checked issues will be removed from the relationship when the current issue is updated. Removing an issue from a relationship does not delete the issue.

Embedding Related Issue Display Layouts Within a Layout

As stated above, related issue display layouts are designed to be embedded within *add, edit*, report and other layouts which are then embedded within *add, edit* and report layouts. This screenshot shows a related issue display layout embedded within an *edit* layout:
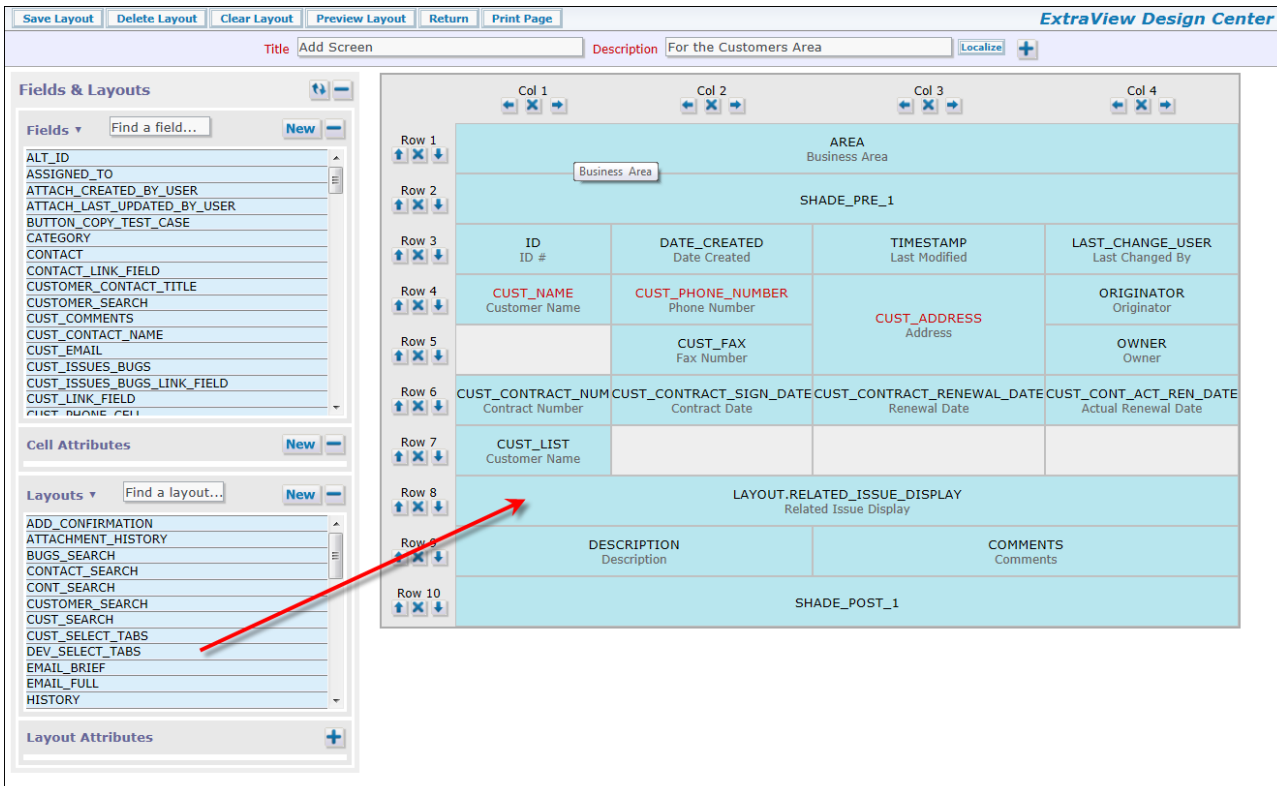
Buttons on Related Issue Display Layouts

The following buttons may be placed on Related Issue Displays. Note that within both *Add* and *Edit* screens the PR_RESOLUTION permission keys are used to control the presence of the buttons, as the information within the displays is always read-only.

- Edit Button - EDIT_BUTTON
- Quickedit Button - QUICKEDIT_BUTTON
- View Button - VIEW_BUTTON

Layout Cell Attributes for Related Issue Display Layouts

Over and above the standard layout cell attributes that can be applied to all cells on layouts, the following attributes allow configuration of embedded related issue display layouts.

Cell Attribute	Purpose
BUTTON ACTION	The **RID BUTTON ACTION** attribute is applied as a cell attribute to the **RID PREHEADER ADD BUTTON**. The action defines what happens when the user presses the add button. This attribute is applied directly to the field's global attributes, defined within the data dictionary, as opposed to being set as a layout cell attribute on the related issue display layout.

- **Target Business Area** - This defines the Business Area that will be used for the *add* screen displayed when the button is pressed
- **Target Project** - This defines the Project that will be used for the *add* screen displayed when the button is pressed
- **Check Required Fields** - If you check this option, then all required fields on the parent issue that will be required by the child issue, are checked that values are filled in, before the child window is displayed. This can be extremely useful if you are going to push down field values from the parent to the child issue, and you want to ensure that the child window is always initiated with a value within a field
- **Action after Submitting the Child Issue** - This option allows you to select the action after the child issue is created. You can either have only the related issue display that contains the newly added issue updated, or you can have the parent issue updated and the entire screen updated. The latter is only required when there are some business rules that triggered changes to the parent issue when the child issue was submitted
- **Push Down Field List** - Double click on the fields whose values you want to push down from the parent issue to the child issue. The selected fields are displayed in the right-hand list
- **Show Generate Script** - This button is provided for advanced usage. It is possible to configure button actions that are more complex than can be done with the **BUTTON ACTION** utility. This option allows you to generate the JavaScript that is close to what you require. You can then copy this JavaScript into the **RID ONCLICK_JS** attribute, and modify it for your requirement. You should not save the **BUTTON ACTION** if you are using **RID ONCLICK_JS** and vice versa.

Note that you may define more than one BUTTON_ACTION on a single related issue display.

DRILLDOWN	This attribute should only be applied to a field that contains the ID of the related issue. You can choose to set the drilldown to either a value of **View** or **Edit** according to the type of drilldown window you wish to open
HEIGHT	This defines the maximum height of the related issue display when it is rendered within an *add* or *edit* layout. The height is input in rows, where each row is estimated to be 20 pixels. If the height required to render the issues is less than this value, then the related issue display is resized to only take the height required. If the height required is more than this value, a vertical scroll bar is placed on the related issue display. The titles to the fields are always fixed in position, so that when the user scrolls the list of related issues, the titles remain visible. The setting has no effect on reports where the entire contents of the related issue display will be rendered.

RID ACT ON FIELD CHANGE	This attribute is not applied to a relationship layout, but is applied to the field (the RID LINK FIELD NAME above) that is used to link the result set from a search to a relationship issue display. When this attribute is present on the RID LINK FIELD NAME, along with an HTML modifier to trigger the action, a related issue display is refreshed, creating links from each issue ID within the RID LINK FIELD NAME, and displaying these issues in the related issue display. A full example of the use of this feature is provided in Example 4.

This attribute is triggered only upon an *onchange* event within a screen, and is not triggered when the *add* or *edit* screen is submitted for an update. A consequence of this is that you should not expect this attribute to work within API calls nor within the File Import process which do not respond to screen events.

RID FILTERS	This cell attribute allows the administrator to specify a set of filters to apply to the selection of the related issues to display. This filter is applied to only the related issue records so the user will only see a subset of the related issues. For example, this screenshot shows the administrator about to select values for filters for the **Business Area** and **Status** fields:
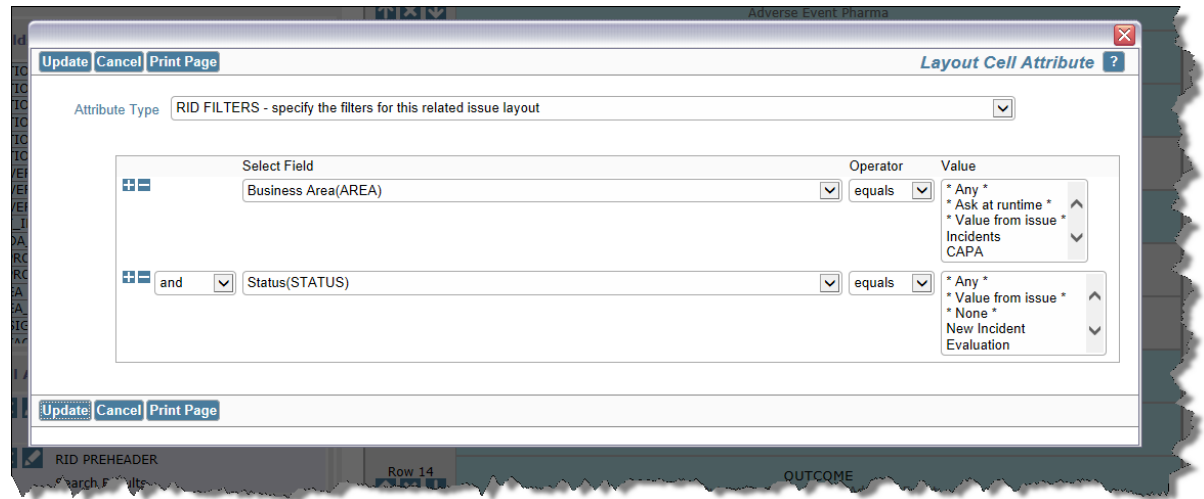
Note that beyond the selection of specific values to use for filters, there are several other possibilities:

- *** Any *** - This value is not used. Setting it is the same as not using the filter
- *** Ask at runtime *** - The user will be prompted when the related issue display is being generated, to provide a value for the filter
- *** Value from issue * -** The value to be used within the filter for the related issue display will be taken from the current value within the field of the same name, that is displayed to the the user on the screen
- **\$\$SYSDAY\$\$ or \$\$SYSDATE\$\$** - These may be used within day or date display type fields to filter on the current day or date. It is not too practical to filter on aa field being equal to \$\$SYSDATE\$\$, as the date includes time to the millisecond, and it's unlikely that issues will match this. Using operators of less than or greater than with date type fields is more useful.

RID FIX TITLES	This attribute may only be set on Related Issue Display layouts within *Add*, *Edit* layouts, or sublayouts of these. When this attribute is used in conjunction with the HEIGHT attribute, and a scroll bar appears after issues have been added, the title to the Related Issue Display is fixed as the user scrolls.

It is not recommended that you place SIZE layout cell attributes on the fields contained within the Related Issue Display layout when you use this attribute. Likewise, it is not recommended that you use this option if you expect that the user will see text that wraps within the Related Issue Display. The best results are obtained when the browser sizes all the columns and column titles.

RID GROUP REFERENCE FIELD	The default for this attribute is the ID field. This attribute may be set to provide a different field for the relationship to be formed between the issues. The field must exist on both the layouts that are being related
RID LINK FIELD NAME	This attribute is used when you are using a search layout on an *add* or *edit* layout to return one or more issues to relate them to the current issue. RID LINK FIELD NAME must be a field with a display type of *text field* or *text area* and the field is typically hidden within the *add* or *edit* layout and is used to transition the list of issued from the search results into the list of issues to be related. The list is a series of issue ID values, delimited by semi-colons. If the RID SINGLE SELECT attribute in this table is set, then there can only ever be a single issue

ID in the list. *Text fields* are capable of storing 255 characters. The link field stores the ID's of all issues selected within the results, plus an extra character as a separator between each ID. They therefore store approximately 40 - 60 ID's. If you are likely to link more than this number of issues, make sure the RID LINK FIELD NAME has a display type of *text area*. These are highly scalable for this purpose.

Linking a Single Result to Populate Fields within the Current Issue

If you are using the RID LINK FIELD NAME to link the search results to a single issue and you then want to take fields within that issue and place their values within the current issue, you are likely to also configure a *link* and a *refresh* business rule to retrieve the field values you require from the search result. See Example 3.

Linking Multiple Search Results to Populate a Related Issue Display

If you are using the RID LINK FIELD NAME to link the search results to multiple issues that you want to relate to the current issue, and place these issues within a related issue display on the current layout, you are likely to also configure RID LINK FIELD NAME with these attributes:

- **RID ACT ON FIELD CHANGE** - This attribute indicates that an action will be taken when this field is updated as a result of being linked to search results
- **RID RELATIONSHIP NAME** - This is the name of the relationhip group into which the records from the search results are to be placed
- **RID RELATION TYPE** - This is the type of the relationship. Usually this will be CHILDREN, but PARENTS may also be appropriate

See Example 4 for details on how to configure this.

RID ONCLICK_JS	**RID ONCLICK_JS** is the JavaScript that is executed when the **RID PREHEADER ADD BUTTON** button is clicked. For most purposes, the script that is executed is created by the **RID BUTTON ACTION** cell attribute and you do not require to use the **RID ONCLICK_JS** attribute. The **RID ONCLICK_JS** attribute is provided for advanced configurations where the administrator wants to create JavaScript that goes beyond the capability of the **RID BUTTON ACTION** attribute. You must not configure both the **RID BUTTON ACTION** and the **RID ONCLICK_JS** attributes on the same button, as they will conflict with each other. The JavaScript for **RID ONCLICK_JS** should be carefully constructed, and there may be differences when wanting to use the JavaScript to initiate a popup to add a new issue when on the *add* and on the *edit* screens. If you want to add related issues on both the *add* and *edit* screens, you may need to configure a different **INCLUDE ADD BUTTON** in the data dictionary for each screen, each with the appropriate **RID ONCLICK_JS** value. It is possible to configure and place multiple buttons to add different types of issues on a single related issue display.

Basic Syntax

Substitutable values are possible in this string value. To create a modal popup window the basic JavaScript you use will be:

```
doEditAEActions(window.parent.extraview_app_home_url , '__SESSION_ID__',
new Array('POPUP_MODAL_ADD', 'add_confirm_page=NO' + rgParameters(),
'RELATIONSHIP_GROUP_REFRESH', '__LAYOUT_TYPE_ID__' ));
```

Note the two substitutable variables SESSION_ID and LAYOUT_TYPE_ID. They must be surrounded by double underlines. This provides the mechanism for ExtraView to substitute the variable at runtime. You set a value of YES or NO to the add_confirm_page parameter, depending on whether you want to display or suppress the verification layout or the **ADD_CONFIRMATION** layout that is typically shown to users after submitting a new issue. If you set NO, then the verification screen is never shown. There is a built-in method that generates the necessary relationship group parameters for the ADD operation. These are needed to force the insertion of the child issue into the correct Relationship Issue Display relationship group using the relation type of the Related Issue Display. This is rgParameters() and it is in scope from the button when the button is inserted into the header of the Related Issue Display.

Updating and Refreshing the Underlying Issue

There are use cases where you might want to update and refresh the parent *Add* or *Edit* screen, after you add a child issue. For example, while adding a child issue, Business Rules might execute that update the parent issue. If you want this behavior, there is an **UPDATE_MAIN** parameter. The workflow might be to go from an *Add* or *Edit* screen, add a child record in a modal popup window, submit the child issue within the modal popup (this

executes Business Rules that update the parent issue), then close the modal popup and finally redisplay the updated parent issue, you will include the **UPDATE_MAIN** parameter as follows:

doEditAEActions(window.parent.extraview_app_home_url , '__SESSION_ID__',
new Array('UPDATE_MAIN','POPUP_MODAL_ADD', 'add_confirm_page=NO
&p_rg_relate_this_to=' + getValueFromParent('p_id') + rgParameters(),
'REFRESH_ID', getValueFromParent('p_id')));

Note that you do not require the **RELATIONSHIP_GROUP_REFRESH** and accompanying **LAYOUT_TYPE_ID** parameters, as the **REFRESH_ID** takes care of refreshing the entire parent issue. If this is executed from an *Add* screen, the parent issue is displayed within an *Edit* screen, as the underlying Business Rules submitted the parent issue to the database, thus meaning it is no longer in the add state, but it is now in a state where it can be updated. The p_rg_relate_this_to parameter and its value are used on the *Add* screen to identify the parent issue to which you want to relate the child issue you are adding, when you are including the **UPDATE_MAIN** parameter. It has no effect otherwise on *Add* screens, or *Edit* screens and may be omitted in these cases.

Setting Additional Parameters to Push Values from the Parent to the New Issue

Other parameters may be added to the array value immediately following the POPUP_MODAL_ADD entry. An example is to set the Business Area and Project of the issue to be added. With rgParameters(), an example becomes:

doEditAEActions(window.parent.extraview_app_home_url, '__SESSION_ID__',
new Array('POPUP_MODAL_ADD',
'p_area=$$MY_AREA$$&p_project=$$MY_PROJECT$$&add_confirm_page=NO
&p_rg_relate_this_to=' + getValueFromParent('p_id') + rgParameters(),
'RELATIONSHIP_GROUP_REFRESH', '__LAYOUT_TYPE_ID__'));

Note that the values for the p_area and the p_project parameters must correspond to the AREA_ID and PROJECT_ID into which you will add the related issue. Another possibility is that you might want to pass the values of one or more fields from the parent issue to the child issue you are creating. To achieve this, you may use either the pushDownFields('p_fld1;p_fld2;p_fld3;...') function to pass several fields at once or use the getValueFromParent() function to pass individual fields. For example, let's say that you want to pass the FIRST_NAME and LAST_NAME fields from the parent to the child. The string then becomes either:

doEditAEActions(window.parent.extraview_app_home_url , '__SESSION_ID__',
new Array('POPUP_MODAL_ADD', 'p_area=$$MY_AREA$$&p_project=$$MY_PROJECT$$
&p_rg_relate_this_to=' + getValueFromParent('p_id') +
pushDownFields('p_first_name;p_last_name') + '&add_confirm_page=NO' +
rgParameters(), 'RELATIONSHIP_GROUP_REFRESH', '__LAYOUT_TYPE_ID__'));

or using individual parameters:

```
doEditAEActions(window.parent.extraview_app_home_url , '__SESSION_ID__',
new Array('POPUP_MODAL_ADD', 'p_area=$$MY_AREA$$&p_project=$$MY_PROJECT$$
&p_rg_relate_this_to=' + getValueFromParent('p_id') +
'&p_first_name=' + getValueFromParent('p_first_name') +
'&p_last_name=' + getValueFromParent('p_last_name') +
'&add_confirm_page=NO' + rgParameters(),
'RELATIONSHIP_GROUP_REFRESH', '__LAYOUT_TYPE_ID__' ));
```

Note that if a field name that you pass within the pushDownFields() or the getValueFromParent() function does not exist, it will be ignored when the user presses the button to add the issue, and no error will be generated. This can be used to advantage in configurations where you want to use a single button for multiple purposes on different forms.

In the above examples, note that you may use the actual values for MY_AREA and MY_PROJECT by looking at the administration screens for the AREA and PROJECT fields to find the values. However, this is not recommended as hard-coded actual values may change within a metadata migration but the hard-coded values in these expressions do not change in a metadata migration exercise.

Other assignment examples:

p_area=$$MY_AREA$$ The AREA field is assigned the value of the current Business Area

`p_area=HELPDESK`	The `AREA` field is assigned the value of the Business Area with the name of `HELPDESK`
`p_area=$$Helpdesk$$`	The `AREA` field is assigned the value of the Business Area with the title of `Helpdesk`
`p_list_field=$$'Ben & Jerry'$$`	In this example, a list field with the name `LIST_FIELD` is assigned the value of Ben&Jerry. The single quote marks are used when the value contains special characters such as &, # or spaces

Opening the Modal Popup Window in Edit Mode

There is a feature within ExtraView that allows the user to open up a new issue in Edit mode, as opposed to the Add mode. This configuration assigns an ID and / or an ALT_ID and saves the initial issue within the database before then placing the user within Edit mode on the screen. This can be configured within the modal popup window, by altering the `POPUP_MODAL_ADD` parameter to `POPUP_MODAL_OPEN_IN_EDIT`.

Repeating Rows Fields

The function getValueFromParent may be modified to access repeating row field values. Repeating rows are indexed, beginning with zero, therefore to access the first repeating row, the call will look like:

getValueFromParent('p_fieldname', 0)

To access the third repeating row, the call will look like:

getValueFromParent('p_fieldname', 2)

Although you may target any repeating row to obtain a value from a field, the value will always be placed in the field on the first row of the child record. The following technique will allow you to get values from successive repeating row fields, and populate these into successive repeating rows within child records:

getValueFromParent('p_fieldname', 2); getValueFromParent('p_fieldname', 3)

This example gets the field value from the third repeating row and places the result in the first repeating row child record, and then gets the value in the fourth repeating row field and places its value in the second repeating row child record

Multiple Add Buttons

Multiple Add buttons on the preheader are supported. Under rare circumstances, you might want to create a specific sort order for these buttons. There is not a place on the user interface to define the sort order of these buttons, but please contact ExtraView Support who will guide you through the process.

RID PREHEADER	This attribute adds a title to a related issue display header to contain an **Add** or other buttons. No title is added to the related issue display if this attribute is not defined. There is one special property of this field. If you use the value of **TITLE** in the value then the description of the related issue display layout will be used as the title in the pre-header. The reason why you might want to do this is solely if you need to provide a localized equivalent for the title in languages other than the base language of your system. The related issue display title may be localized, while layout cell attributes (with the exception of the ALTERNATIVE TITLE) may not be localized.

RID PREHEADER ADD BUTTON	This attribute adds a button to the pre-header. This button is used to add new issues to a relationship. The button is a field that must already be created in the data dictionary. The button must be created on the *Label* tab within the data dictionary and must have a display type of *Button*. The title to the field will be used as the title to the

button when it appears on the pre-header of the related issue display. You may add any number of Add buttons to a single Related Issue Display

RID PREHEADER REFRESH BUTTON	This attribute is used when you are using a search layout on an *add* or *edit* layout to return one or more issues to relate them to the current issue. This places a button onto the related issue display pre-header bar that allows the user to refresh the list of results from the current set of search filters. The button has a default title of **Get Filtered Results**, but you can optionally provide a value to the attribute which will then appear as the label on the button
RID PREHEADER SUBMIT BUTTON	This attribute is used when you are using a search layout on an *add* or *edit* layout to return one or more issues to relate them to the current issue. This places a button onto the related issue display pre-header bar that allows the user to submit the list of results from the current set of search filters. The button has a default title of **Fetch Selected Record(s)**, but you can optionally provide a value to the attribute which will then appear as the label on the button.

Multiple Submit Buttons

Multiple Submit buttons on the preheader are supported, typically to allow different types of issue submissions. Under rare circumstances, you might want to create a specific sort order for these buttons. There is not a place on the user interface to define the sort order of these buttons, but please contact ExtraView Support who will guide you through the process.

RID RELATIONSHIP NAME	This is the name of the relationship group to which the related issues will be added. The relationship group must be created before you add the name to the cell attribute. If you do not set a value for this attribute, then the value in the behavior setting named RELATION_GROUP_DEFAULT will be used
RID RELATION TYPE	This should be set to one of the following values:

- **MEMBERS** - This is the default value for this attribute if you do not override it with one of the other values. In this case the related issue display will show all the members of the group(s), containing the reference issue, i.e. the reference issues parents, children, and other issues not directly related to the reference issues in the group(s).
- **CHILDREN** - The related issue display will only contain issues that are the children of the reference issue
- **PARENTS** - The related issue display will only contain issues that are the parents of the reference issue. Note that in the ExtraView data model it is possible to define more than one parent issue for a specific issue. See the section on Relationship Groups for more information.
- **RELATED** - In this instance ExtraView will display the parents and the children of the reference issue within the specified group(s), within the related issue display. The parents and children of these related issues will not be displayed.
- **SIBLINGS** - ExtraView will display all the issues at the same level in the hierarchy. For example, if a parent record has five children, and you create a related issue display with all the children, using a relationship group relation type of SIBLINGS, then you will see the four siblings to the current one.
- **GRANDPARENTS** - This allows the display of the grand-parents to the current level in the hierarchy
- **GREATGRANDPARENTS** - This allows the display of the parents of the grand-parents in the hierarchy
- **GREATGREATGRANDPARENTS** - This allows the display of the parents of the great-grand-parents in the hierarchy
- **GRANDCHILDREN** - This allows the display of the grand-children to the current level in the hierarchy
- **GREATGRANDCHILDREN** - This allows the display of the children of the grand-children in the hierarchy
- **GREATGREATGRANDCHILDREN** - This allows the display of the children of the great-grand-children in the hierarchy
- **LINKED** - This type does not use a relationship type per se, but relies on the Related Issue Display filters to retrieve the associated issues. For example, you might have a filter that selects all the open issues for a particular product, and display these on a related issue display of each issue that you call up.

If you are using the related issue display to add new issues, the value should be CHILDREN or PARENTS. If you choose another relationship type then the issue will be added, but will not be displayed as the relationship is indeterminate

RID SEARCH FILTER LAYOUT	This attribute is used when you are using a search layout on an *add* or *edit* layout to return one or more issues to relate them to the current issue. This field holds the name of the search layout that is being embedded on the same layout as the related issue display. Typically you will embed this search layout just above the related issue display layout

RID SEARCH ON NO FILTERS	The layout element attribute RID SEARCH ON NO FILTERS allows an embedded search report to be created even when there are no filters added in the embedded layout. Normally, at least one such filter must be present for the search to occur to prevent the potential return of a huge numbers of issues. By putting this attribute on a Related layout element that references a RID SEARCH FILTER LAYOUT a search may be done even when no filters are set.
RID SINGLE SELECT	This attribute is used when you are using a search layout on an *add* or *edit* layout to return one or more issues to relate them to the current issue. The default is that the search will allow you to return all the issues the user selects from the results. If you set this attribute, then the user will only allow a single issue to be selected to return the results into the related issue display
RID SORT ORDER	This attribute determines a sort order, based on a field that is displayed on the related issue display. You can set the value to ASCENDING or DESCENDING
RID TRANSPOSE LAYOUT	The default is that the related issues will be presented on the display by row. If you select this attribute, then the issues will be presented by column
SIZE	This controls the width of the related issue display on *add* and *edit* screens. The size is an estimate of the number of characters to be displayed. If you do not provide a layout cell attribute for SIZE, then the default is 125 characters. Beginning with version 10, you can no longer use a SIZE layout cell attribute on the individual fields within the related issue display to control their width. This allows the enabling of a new feature, to freeze the titles of the related issue display. This is very beneficial when you have a significant number of related issues within the display, as the titles always remain visible.
VISIBLE IF	This allows you to make the entire related issue display visible or not visible depending on the value of another field on the layout within which the related issue display is embedded

Refresh Button on a Related Issue Display

There are occasions when it is useful to have a refresh button on the related issue display preheader. This can be accomplished with the following steps:

1. Within the Labels tab of the the data dictionary create a field with a suitable name, e.g. **MY_REFRESH,** with a display type of **Button** and a title of **Refresh**
2. Edit the field you have just created within the data dictionary, and add a Global Attribute of **RID ONCLICK_JS**. This should have a value of:

```
doMyRGRefresh(__LAYOUT_TYPE_ID__);return false;
```

3. Navigate to the layout that contains the related issue display where you want the button
4. Add a new layout cell attribute of **RID PREHEADER ADD BUTTON**, and select the **MY_REFRESH** button to display
5. Save the layout.

Order of Precedence on Related Issue Display Operations

Given that the end user may carry out a combination of operations on several, and sometimes independent, related issue displays with an *add* or *edit* screen, there might be some conflict in these operations, with records that might appear in more than a single related issue display. For example, what happens if you update an issue in one relationship, but remove the same issue in another relationship? The RELATIONSHIP_GROUP_REMOVE_BTN is given precedence over other transactions. Transactions are processed in the following order: removes, inserts, deletes (where the deletes are relationship group remove button transactions).

Related Issue Configurations

Relationship groups are extremely adaptable and can be configured in many different ways, to meet the needs of many different business problems. Here are some of the most common ways in which you can configure relationship groups:

- **Arbitrary addition of records to form a relationship** - Add an existing issue ID to the current issue, to relate the issues together, where the issue you are updating becomes a child of the issue ID that you enter. A typical use case is where you might

have an issue which has some elements in common with another issue and you want to treat them together. For example, two customers may report the same fundamental issue, and you want to group these together. You require a mechanism to add issues to the common group, and to display the related issue when you are looking at any one of the issues. This forms Example 1

- **Structured addition of child records to a parent record** - Add any number of new issues as children to the current issue. A typical use case might be that you want to add a number of contacts to a customer's definition as you might find in a CRM system. The customer record is the parent issue, and each of the contacts is a child record to the customer record. This forms Example 2
- **Search for a single record and populate the current record with fields from the result** - Create an *add* or *edit* layout which allows the end-user to search for results that are returned, and then insert values from the result into fields within the current issue. A use case may be to have a layout that creates customer complaints, but before entering the specifics of the complaint, the user is able to search for the appropriate customer and return all the details required into the current complaint. This forms Example 3
- **Search for multiple records matching specific criteria and relate the results to the current record** - Example 4 shows how you might search a large set of test cases for a subset, and relate this subset to a current record, forming a test plan, where the records added as children form the test cases for the test plan.

Example 1

This example describes how to set up a relationship group that enables an issue to be related to a parent issue in an ad-hoc way – simply by entering an issue ID into a text field on an *add* or *edit* screen of the child issue. Each issue will then display the related issue in an embedded layout, where you have the option to remove each related issue, or to drill down and edit the related issue.

Here is a screenshot of the *edit* screen of the parent issue after setting up this example:
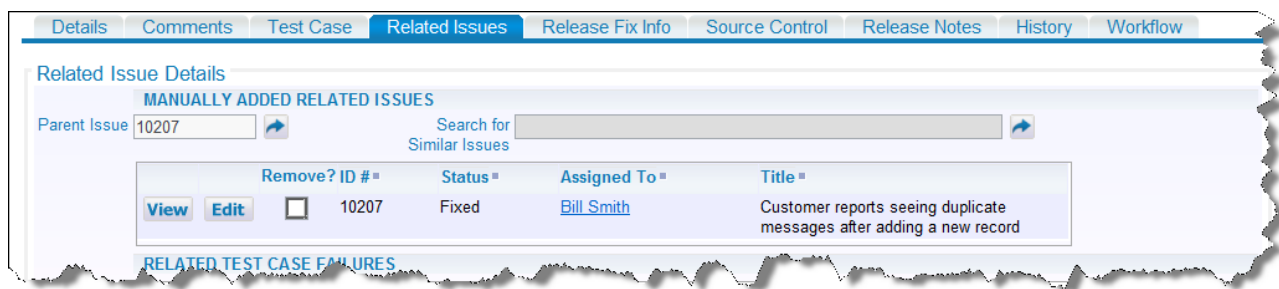
Example related issue display with its connecting parent issue ID

Step 1 – Setting up The Relationship Group

In this example, we will set up a relationship group that named BEST_DATA_RG. This will be a One-to-Many type of relationship group and uses a Parent / Child structure for the related issues.

1. Go to **Administration ▶ Site Configuration ▶ Relationship Group Maintenance** and ensure the BEST_DATA_RG group exists – you may need to check the box next to **"Show relationship groups that have no issues or only contain Closed issues"**.

Relationship Group Maintenance Screen

2. If the BEST_DATA_RG relationship group is not present, create it by clicking on the **Add** button
3. Set the Fixed name to BEST_DATA_RG and give it a Title of *General Relationship Group.* Select "One-to-Many" from the "Type" drop-down list. Click Add to complete the operation. You may use a different name for the relationship group, but remember this as you will need it later.

Step 2 – Create a Field in the Data Dictionary

Now that the relationship group is defined, we will create a field in the Data Dictionary that will handle any related issue number entered by the user.

The new field

We also turned on **Display as URL** for the field, and set the **URL** to:

?p_action=doEditDisplay&p_id=$$VALUE$$&p_option=Display&p_from_action=search &p_from_option=search

This provides a drilldown button by the field so we can see the parent issue by clicking on the button.

Step 3 – Grant Security Privileges

Next, make sure to set read/write permissions for any role that will be accessing this field.

1. Go to **Administration ▶ Site Configuration ▶ Grant Security Privileges**
2. Select the permission keys PR_RESOLUTION.MY_PARENT_ID
3. Give Read and Write permissions for the user roles that will be using the related issues capability.

Step 4 – Place Field and Related Issue Display layout

Now we need to place these fields on a layout so they are accessible.

Ensure the Related Issue Display is available

- If the layout does not exist, or you want to override the one that is present in the Global Area, create this layout:
  1. Select **Related Issue Display** from the * **Select Layout Type to Add** * list
  2. Add the desired fields you want displayed in this layout. This is an example of a related issue display layout:

Related Issue Display layout example

- After saving the Related Issue Display layout, edit the *Edit Issue* screen for this business area
- Place the field that you created earlier onto this layout. Also, place the Related Issue Display layout onto the *edit* layout
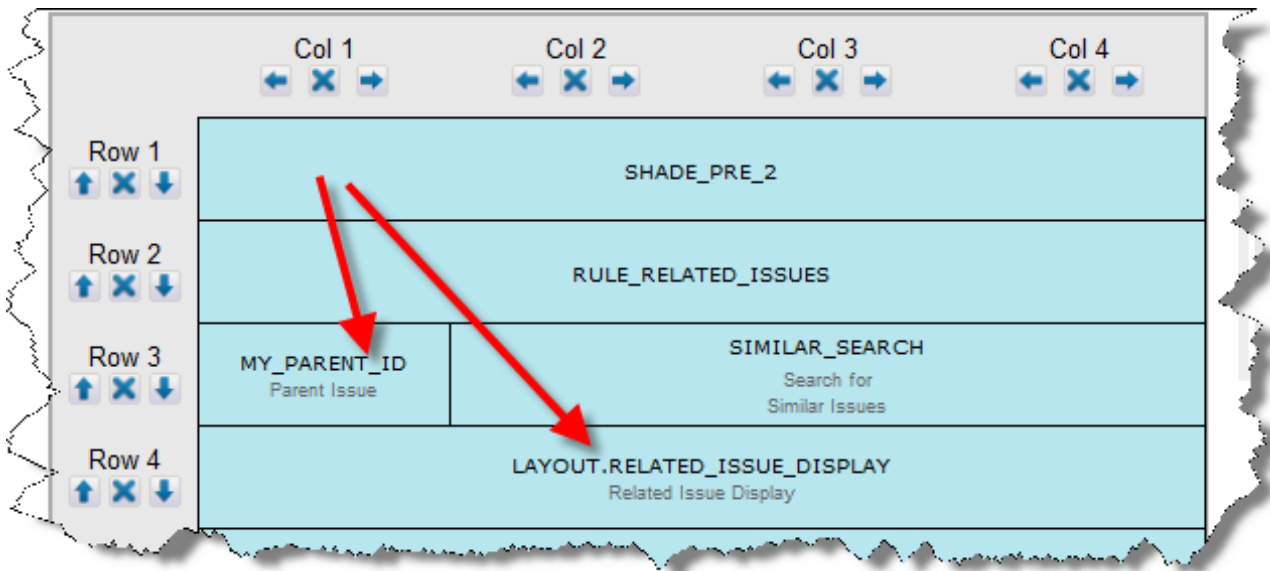
The fields and related_issue_display on the Edit Issue Layout

- Add the following layout cell attributes to the MY_PARENT_ID field by clicking on the link button next to the field name:

Attribute	Value
RID RELATIONSHIP NAME	BEST_DATA_RG
RID RELATIONSHIP TYPE	Related

- Add the following layout cell attributes to the RELATED_ISSUE_DISPLAY by clicking on the link button next to the field name:

Attribute	Value
RID RELATIONSHIP NAME	BEST_DATA_RG
RID RELATIONSHIP TYPE	Related
VISIBLE IF	ID is not null
HEIGHT	30

- Save the layout.

Step 5 – Updating Related Issues when Updating the Parent Issue

If you want the capability to update all related issues when performing the update of an issue, continue with the following steps. Otherwise, ad hoc related issues are now configured on your site.

1. Go to **Administration** ◆ **Site Configuration** ◆ **Behavior Settings** and make sure the behavior setting named RG_UPDATE_BILEVEL_ONLY is set to a value of NO
2. Go to **Administration** ◆ **Site Configuration** ◆ **Design Center**. Ensure that you have a layout with a type of POST_EDIT_UPDATE defined. If your installation does not have a layout of this type defined, use the Create and Maintain Layout Types entry on the Fields & Layouts administration menu to define this
3. Add the layout with the POST_EDIT_UPDATE type to your current business area and project, or in a location from which it will be inherited
4. Place the fields on the layout that you want to be available during the update process for related issues. It should look something like the following screenshot.
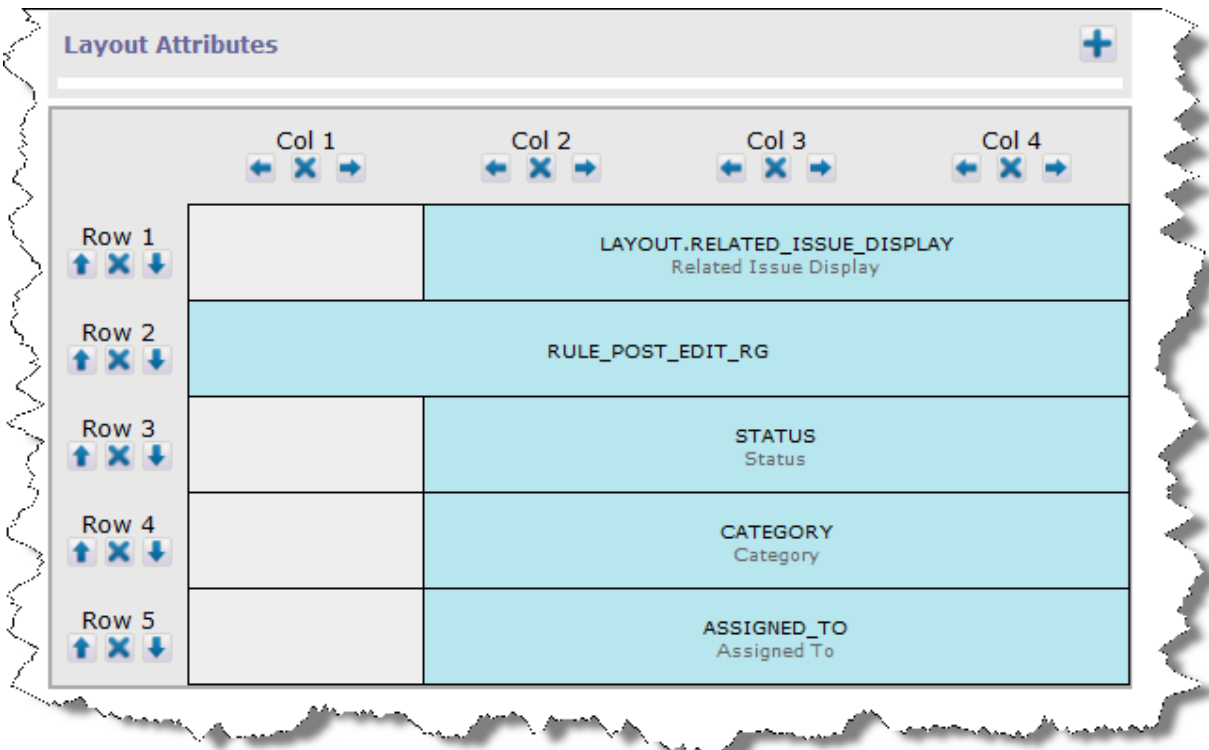
Note that the fields named STATUS, RESOLUTION and ASSIGNED_TO are the fields that will be updated in the related issues. It is essential that you place the embedded layout, LAYOUT.RELATED_ISSUE_DISPLAY on this layout. Also note that the field named RULE_POST_UPDATE_RG is defined as a custom field and has its help text defined as **4 - Check the fields in the list below that you want to update to the same values in the issue being updated.** The new value is shown. Remember that you may need to set the RID RELATIONSHIP NAME, RID GROUP REFERENCE FIELD and the RID RELATION TYPE layout cell attributes for the LAYOUT.RELATED_ISSUE_DISPLAY. When you edit the parent issue, the user will have the option to update the related issues at the same time.

Example 2

- [Add new comment](#)

A common requirement is to have one issue (the parent issue) create related issues as children. For example, you might have a process that spins off an arbitrary number of actions to employees around your organization, and for each of these employees to complete a task before the parent issue can be closed. You want to have an **Add** button on the parent issue to add the child issues one-by-one. This example shows how a parent customer screen can add any number of contact records to different people as child records. At the same time, some fields from the parent issue are "pushed down" into the child record, to pre-populate the child record with values that are required.
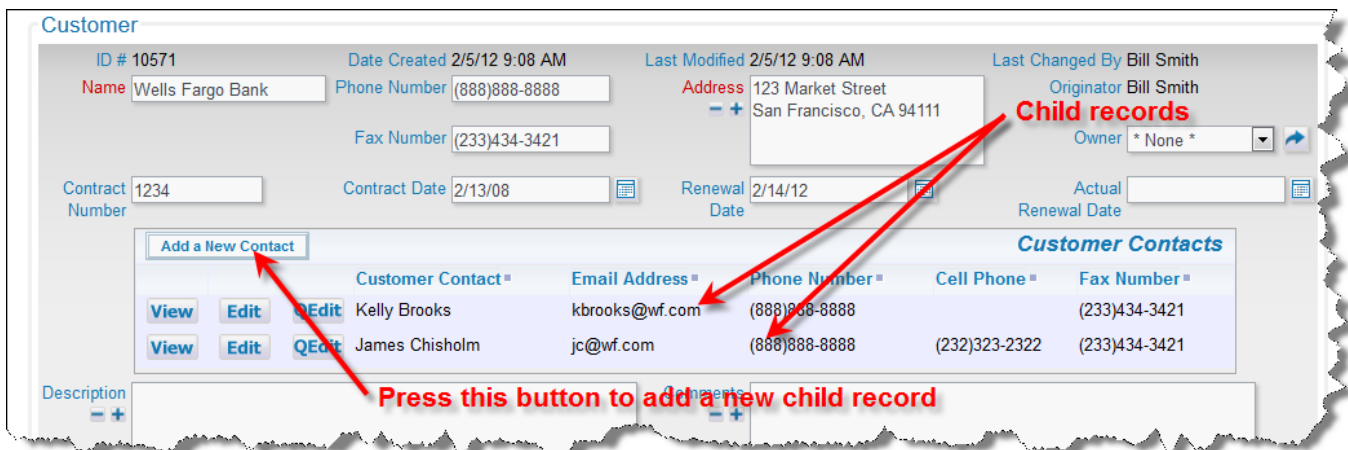
Adding related issues from an Add or Edit screen

The popup Add window where you add the child record

To setup this feature, use the following sequence as a guide:

- This feature presently only works when your installation uses the ID field, as opposed to the ALT_ID field, as the unique identifier for all issues
- Definitions and screens for parent issues to which you want to add the children. For this example let's say this parent issue is in a Business Area titled **Customers** and the data is stored in the Project titled **Customer Information**. The name for the Business Area is **CUSTOMERS** and the Project is named **CUSTOMERS_DATA**
- Definitions and screens for child issues to be added to the parent. For this example let's say the child issues are in the same Business Area titled **Customers** but the data is stored in a separate Project titled **Customer Contact**. The Project is named **CUSTOMERS_CONTACT**
- Have a one-to-many relationship group defined for the link. For our example, we will name this **CUSTOMER_CONTACTS**
- You need a button defined as a field in the data dictionary, which is used to add the child records. This must be created as a field on the **Label** tab and can have any name. Its display type must be **Button**. The title of the field is used as the text for the button on the screen. In our example, the field is named **CONTACT_ADD_BUTTON**. There is an attribute that you can define for this button field, which controls a number of elements:
    - Which Business Area and Project into which you want to add the child record
    - The behavior when you submit the child issue, for example, whether to just refresh the parent window with the information from the new child record, or whether update the parent record with information from the new child record
    - To define a set of fields from the parent record that you want to "push down" to the child record, so the values of these fields can automatically be assigned to fields in the child record

This attribute is the RID BUTTON ACTION attribute and should be configured under the Global Attributes tab within the Data Dictionary. The configured attribute will look like this:

Configuring the Button Action

- Create a layout type to hold the related issues or use the inbuilt related issue display layout named RELATED_ISSUE_DISPLAY. If you create your own layout type, the name must start with **RELATED_**, followed by other characters. For example, name the layout type **RELATED_CONTACTS** and make sure that its usage is **Report**. In our example we will just use the inbuilt RELATED_ISSUE_DISPLAY
- Add the layout you are using into the **Customers** Business Area and the **Customer Information** project. In our example, this is the **RELATED_ISSUE_DISPLAY**. Onto this layout, add the fields from the child issue that you want to display on the related issue display on the parent issue. This might look like this:

The Related Issue Display

The layout cell attributes that you see are used as follows:

- **FIELD HEIGHT** - This provides a maximum height before the child issues will be displayed with a scroll bar. This is approximately the number of rows of data
- **RID RELATIONSHIP NAME** - This should be **CUSTOMER_CONTACTS** for our example
- **RID RELATION TYPE** - This should be **CHILDREN** for our example
- **RID PREHEADER ADD BUTTON** - The name of the button you created in the data dictionary
- **RID PREHEADER** - The title you want to place at the top of the related issue display, to the right of the *add* button
- **FIELD SIZE** - The width of the related issue display, approximately in characters
- Test the configuration by adding the essential fields to the parent record, then clicking the *add* button and adding a child record.

Example 3

It is often useful to search for related issues in order to identify information that you want to copy into the current issue. For example, you might have a business area that contains all your customers, and another business area where you want to create issues that any customer might report. This feature allows you to enter some key identifying information about a customer, such as their name, email address or telephone number, bring back all matching records, and then select the specific customer that you want to associate the customer issue with.

Initial customer search

Entering the search criteria and seeing the results

Selecting the result and populating the issue

When you look at the above screenshot, you will see that you have a search capability, a search results area, and the underlying *add* screen into which you want the selected record to appear. The basic configuration is this:

- Start by creating the *add* or *edit* screen in the normal way
- Create a search layout to be used as the available filters for the search
- Create a related issue display to hold the results of the search
- Assembling the search layout and related issue display layout on the *add* or *edit* screen
- Associate the fields on the underlying *add* or *edit* screen with the fields to be populated from the search results

The Search Layout

First, create a new layout type. This may have any name and any title, but it must have a usage type of **Search**. This is used to contain the search filters that will be displayed to the user. For our example, we will use the name CUSTOMER_SEARCH. Next, add the new layout to the area and project where it is needed, creating a new layout. Onto this layout, you will place the fields to be used as filters. No special layout cell attributes are required, but you may use whatever attributes help you provide an aesthetically pleasing display. Save this layout.
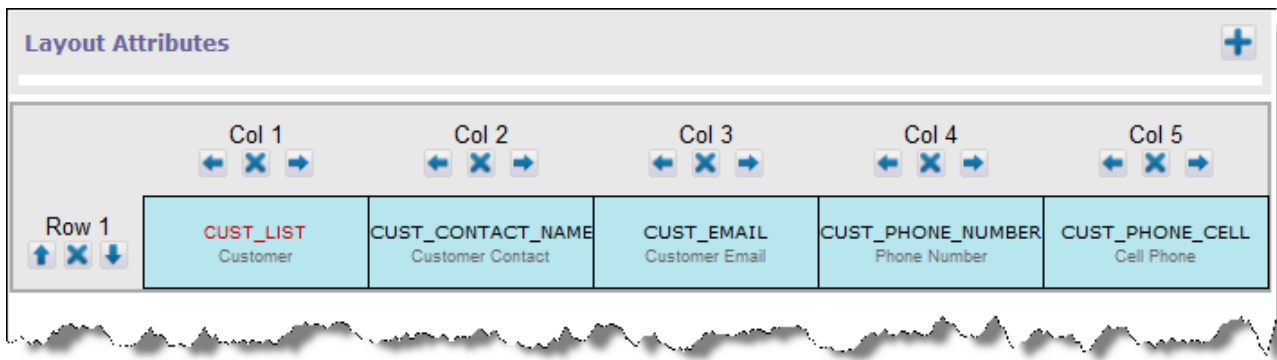
The search layout

The Related Issue Display Layout

This is used to contain the results of the search. Note that the fields displayed do not need to match the fields within the search layout, although it is common to see several fields in common. Create a new layout type to contain the related issue display. By convention, begin the name of the new layout type with **RELATED_** and followed by remaining characters that provide an unique name for the layout. The usage type of this layout must be **Report**. For our example, we name this layout type RELATED_CUSTOMER_ISSUES. Now, add the new layout to the area and project where it is needed, creating a new layout. In the first column of the layout, place the inbuilt field with the name **RELATIONSHIP_GROUP_CHOOSE_BTN**. This is used as the selector for the record(s) whose results you want to place into the *add* or *edit* screen. It is recommended that you only place the **VIEW_BUTTON** as any other button on this layout. You should not place other buttons such as **EDIT_BUTTON** or **QUICKEDIT_BUTTON** on this layout. Place the remaining fields that you want to display on the related issue display on the layout, and save the layout.
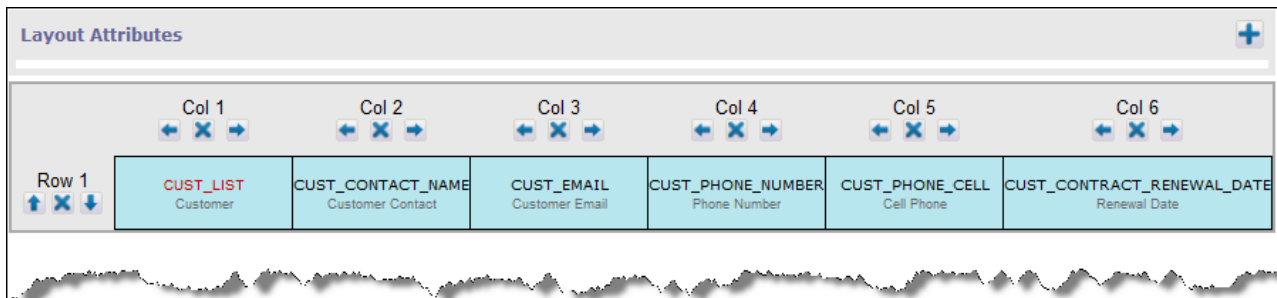
The related issue display layout

Configuring the *Add* or *Edit* Screen to Use the Search and Related Issue Display Layouts

A link field is required to pass the results from the selection made on the Related Issue Display layout, so begin by creating this field, with a display type of text. Give this field read / write permissions for the roles that will be adding customers. This field must be placed on the *add* or *edit* screen, but it need not be visible. For example, you might provide this field with two layout cell attributes to hide it. Set a FIELD STYLE of **display:none** and a FIELD LABEL TAG of **display:none** as layout cell attributes, thus ensuring the field can never be seen on the screen by any user. Note that you cannot use the FIELD/LAYOUT VISIBLE IF layout cell attribute. This is because this attribute only places the field into the HTML form when it becomes visible. We need the field on the form even when it is invisible to be used as the link field. In our example, we use the name CUST_LINK_FIELD.

Place your search layout on the *add* or *edit* screen. This layout does not require any layout cell attributes, but you might decide to hide it using FIELD/LAYOUT VISIBLE IF, once you have made your selection from the results of the search.

Place your related issue display layout on the *add* or *edit* screen. This is the place where you configure several layout cell attributes that link that bind all the elements together:

- RID RELATION TYPE - You must set this to a value of **Linked**
- RID SEARCH FILTER LAYOUT - This identifies the Search Layout you created above. Simply select its name from the list of values
- RID PREHEADER - This attribute provides an area above the related issue display that contains the function buttons, as well as providing a title to the related issue display. The value you provide with this attribute becomes the title. Unless you define a RID PREHEADER attribute, the RID PREHEADER REFRESH BUTTON and the RID PREHEADER SUBMIT BUTTON will not appear, as the RID PREHEADER is used as a container for these buttons

- **RID PREHEADER REFRESH BUTTON** - This places a button onto the related issue display pre-header bar that allows the user to get the search results from the current set of filters, and to refresh the results if the user changes the filters. The button has a default title of **Search**, but you can optionally provide a value to the attribute which will then appear as the label on the button. You must define a RID PREHEADER for this button to be visible
- **RID PREHEADER SUBMIT BUTTON** - This places a button onto the related issue display pre-header bar that allows the user to update ID's for the selected values within the display into the FIELD NAME. The button has a default title of **Connect**, but you can optionally provide a value to the attribute which will then appear as the label on the button You must define a RID PREHEADER for this button to be visible
- **RID LINK FIELD NAME** - This is the name of the link field that receives the ID's of the selected issues from the related issue display. Simply select the field name from the list presented to you when you add the attribute
- **RID SINGLE SELECT** - Select this attribute if you want to restrict the user to only make a single selection from the list of search results. You will typically select this if you want the destination of the search result selection to be fields within the current *add* or *edit* screen. If it is your intent to allow the user to add an arbitrary number of related issues, then do not add this attribute, thereby allowing multiple search results to be selected.

The search layout

Developing Business Rules to Associate the Issue with the Results

The business rules work by getting the values of the ID's within the CUST_LINK_FIELD and linking the current issue to the selected values. There are three basic premises; you are retrieving a single result into fields in the current issue, you are retrieving multiple results into repeating row records or you are retrieving multiple results into related issues.

Placing Results Within Fields in the Current Issue

In this case the CUST_LINK_FIELD will contain a single value which is the ID of the record we will associate with the current issue. This is done with business rules similar to the following:

```
<== link customerSearchLink ==> AREA='Customers', ID=CUST_LINK_FIELD

<== refresh ==>
if (CUST_LINK_FIELD.{changed}) {
  CUST_NAME          = (customerSearchLink).CUST_NAME;
  CUST_CONTACT_NAME  = (customerSearchLink).CUST_CONTACT_NAME;
  CUST_PHONE_NUMBER  = (customerSearchLink).CUST_PHONE_NUMBER;
  CUST_EMAIL         = (customerSearchLink).CUST_EMAIL;
}
```

The link directive establishes a link based on the ID of the issue we want to retrieve being equal to the value contained in the CUST_LINK_FIELD field we created.

The commands within the refresh directive establish the assignments to be made when the CUST_LINK_FIELD is populated from the search results.

Example 4

It is often useful to search for issues in order to make a selection from the result set, and relate this selection to the current record. The selection may be a subset of the result set. For example, you might have a business area that contains a library of test cases, and you want to create any number of test plans from this library. Each test plan might have a different subset of records chosen from the library.

Initial presentation of screen

In the above screen, you will observe a search layout; this is used to provide a filtered set of search results within a related issue display. Note the buttons in the header of the related issue display.

Entering the search criteria and seeing the results

Once you have selected filters, use the **Get Filtered Results** button to display the results in the related issue display. With your mouse, you can click on any of the results to select that record. There is also a button to **Check All** the results at one time.

The selected results now related to the parent issue

Use the **Fetch Selected Record(s)** to move all the selected records into the final result set where they will be stored with the parent issue. Note that the search layout and results are reset. You may repeat the search and selection process to add additional child records to the parent record. The basic configuration involves setting up 3 embedded layouts, one for the search, one for the search results and one for the selected records. A text or text area field is used to connect the selected valus from the search results to the selected records:

- Start by creating an *add* or *edit* screen(s) in the normal way
- Create a relationship group to define and hold the relationship
- Create a search layout to be used as the available filters for the search
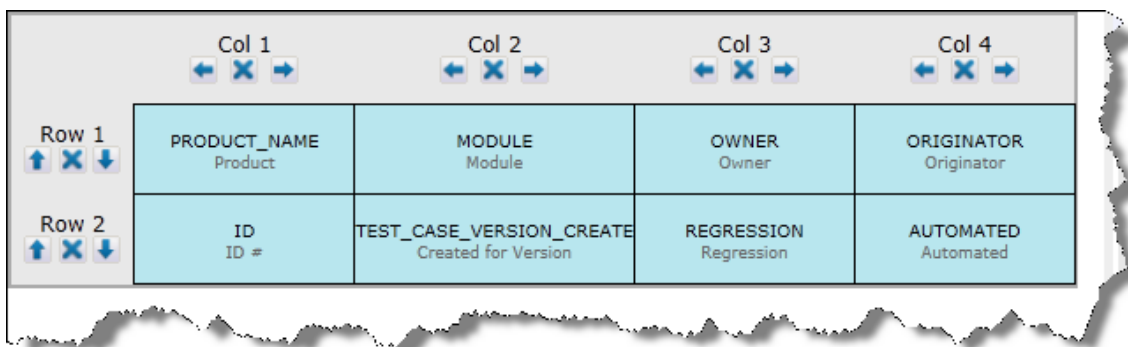
- Create a related issue display to hold the results of the search
- Create a related issue display to hold the final selected records as children to the parent issue
- Assemble the search layout and related issue display layout on the *add* or *edit* screen
- Create and place a link field on the *add* or *edit* screen to act as a transition of the record ID's from the search results to the final related issue display

The Relationship Group

This is created within the Administration utility, **Relationship Group Maintenance**. This should be of the type **One-to-Many**. In our example, we named this **TEST_CASES_TEST_PLANS** and gave it a title of **Test Cases / Test Plans**.

The Search Layout

A prerequisite to making the search layout operate is to configure the Quickfind feature. Once this has been done, create a new layout type. This may have any name and any title, but it must have a usage type of **Search**. This is used to contain the search filters that will be displayed to the user. For our example, we will use the name TEST_CASE_SEARCH. Next, add the new layout to the area and project where it is needed, creating a new layout. Onto this layout, you will place the fields to be used as filters. No special layout cell attributes are required, but you may use whatever attributes help you provide an aesthetically pleasing display. Save this layout.
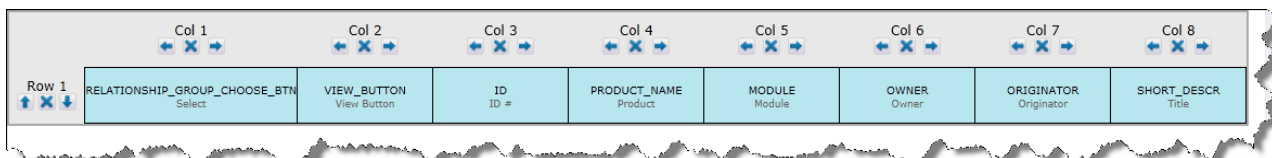
The search layout

The Related Issue Display Layout Containing the Search Results

This is used to contain the results of the search. Note that the fields displayed do not need to match the fields within the search layout, although it is common to see several fields in common. Create a new layout type to contain the related issue display. By convention, begin the name of the new layout type with **RELATED_** and followed by remaining characters that provide an unique name for the layout. The usage type of this layout must be **Report**. For our example, we name this layout type RELATED_TEST_CASE_RESULTS. Now, add the new layout to the area and project where it is needed, creating a new layout. In the first column of the layout, place the inbuilt field with the name **RELATIONSHIP_GROUP_CHOOSE_BTN**. This is used as the selector for the record(s) whose results you want to place into the *add* or *edit* screen. It is recommended that you only place the **VIEW_BUTTON** as any other button on this layout. You should not place other buttons such as **EDIT_BUTTON** or **QUICKEDIT_BUTTON** on this layout as this layout only displays search results. Place the remaining fields that you want to display on the related issue display on the layout, and save the layout.

The related issue display layout with the results

The Related Issue Display Holding the Selected Records

This layout is typically similar to the previous one you created, in terms of the fields you display, but you will likely want to see different buttons. Most typical is to include a **VIEW_BUTTON**, and an **EDIT_BUTtON**. A **QUICKEDIT_BUTTON** is often useful.

The related issue display layout that displays the selected records

Configuring the *Add* or *Edit* Screen to Use the Search and Related Issue Display Layouts

A link field is required to pass the results from the selection made on the Related Issue Display layout with these characteristics:

- Create the link field, with a display type of *text area*. In our example, we use the name TEST_CASE_RESULTS_LINK_FIELD. Give this field read / write permissions for the roles that will be adding the test cases
- This field must be placed on the *add* or *edit* layout you are configuring, but it need not be visible - normally you will want to hide this. The following layout cell attrbutes are recommended:
    - STYLE - set to **style=display:none** to hide the field. Note that you cannot use the VISIBLE IF layout cell attribute. This is because this attribute only places the field into the HTML form when it becomes visible. ExtraView needs the field on the form even when it is invisible to be used as the link field
    - LABEL TAG - set to **style=display:none**
    - RID ACT ON FIELD CHANGE - This ensures that the link field updates the final layout with the selected issues by triggering a refresh of the related issue display with the selected results
    - RID RELATIONSHIP NAME - This is the relationship that you created in the earlier step. In our example, this is **TEST_CASES_TEST_PLANS** with the title of **Test Cases / Test Plans**
    - RID RELATIONSHIP TYPE - Set this to a value of **Children**
    - RID OMIT AUTO POPULATE - This ensures that when you delete an item from the relationship that any value in the original link field is not used, thereby re-populating the child issue back into the relationship.
    - Place your TEST_CASE_SEARCH layout on the *add* or *edit* screen layout. This layout does not require any layout cell attributes, but you might decide to hide it using FIELD/LAYOUT VISIBLE IF, once you have made your selection from the results of the search. Place your RELATED_TEST_CASE_RESULTS related issue display layout for the search results on the *add* or *edit* screen layout. This is the place where you configure several layout cell attributes that link that bind all the elements together:

Cell attributes on the results layout

- HEIGHT - The approximate number of rows to display on the results layout, before displaying a vertical scroll bar. This is optional. If you do not provide this, all the results will be displayed
- RID RELATION TYPE - You must set this to a value of **Linked**
- RID FILTERS - This is optional. You may provide filters here which will be used in addition to filters the end user enters on the *add* or *edit* screen. In our example, we always want the results to be obtained from the **Test Cases** project within the **Test Cases** business area, so we create these filters within the layout
- RID PREHEADER ADD BUTTON - This button is usually used to provide a "Submit" capability on a related issue display. This functionality is not required on search results, so we use this button to provide an alternative function, to *Check All* and *Uncheck All* records in the search result. This field is available in the Data Dictionary on the **Labels** tab. It has a name of **SELECT_ALL_BTN**, and a title of **Check All** in the English locale. It also has a **Global Attribute** of ONCLICK_JS with a value of toggleRIDSelectButtons();. This provides the JavaScript that allows all the records to be selected or deselected. You must define a RID PREHEADER for this button to be visible
- RID PREHEADER - This attribute provides an area above the related issue display that contains the function buttons, as well as providing a title to the related issue display. The value you provide with this attribute becomes the title. Unless you define a RID PREHEADER attribute, the RID PREHEADER ADD BUTTON, the RID PREHEADER REFRESH BUTTON and the RID PREHEADER SUBMIT BUTTON will not appear, as the RID PREHEADER is used as a container for these buttons
- RID PREHEADER REFRESH BUTTON - This places a button onto the related issue display pre-header bar that allows the user to get the results from the current set of filters, and to refresh the results if the user changes the filters. The button has a default title of **Get Filtered Results**, but you can optionally provide a value to the attribute which will then appear as the label on the button. You must define a RID PREHEADER for this button to be visible
- RID PREHEADER SUBMIT BUTTON - This places a button onto the related issue display pre-header bar that allows the user to update ID's for the selected values within the display into the FIELD NAME. The button has a default title of **Fetch Selected Record(s)**, but you can optionally provide a value to the attribute which will then appear as the label on the button. You must define a RID PREHEADER for this button to be visible
- RID LINK FIELD NAME - This is the name of the link field that receives the ID's of the selected issues from the related issue display. Simply select the field name from the list presented to you when you add the attribute
- RID SEARCH FILTER LAYOUT - This identifies the Search Layout you created above. Simply select its name from the list of values
- SIZE - This is an optional attribute with the width, in characters, of the display of the related issue display layout

The Related Issue Display with the Related Records from the Results

Here, you place layout cell attributes on the layout named **LAYOUT.RELATED_TEST_CASE_RESULTS**. This layout displays the records you selected within the search results. The attributes this layout needs are:

- AGGREGATE EXPRESSION - This is an optional attribute that, in our example, is used to provide a total of the hours on the related issue display
- HEIGHT - This is an optional attribute. It provides an approximate row count that is displayed, before a vertical scroll bar is introduced onto the related issue display
- RID RELATIONSHIP NAME - This is the relationship that is being updated. This was created in an earlier step. In our example, this is **TEST_CASES_TEST_PLANS** with the title of **Test Cases / Test Plans**
- RID RELATIONSHIP TYPE - This attribute defines the type of relationship. This should be **Children**
- RID LINK FIELD NAME - This is the name of the link field that receives the ID's of the selected issues from the related issue display. Simply select the field name from the list presented to you when you add the attribute
- SIZE - This optional attribute provides a width, in characters, of the related issue display.
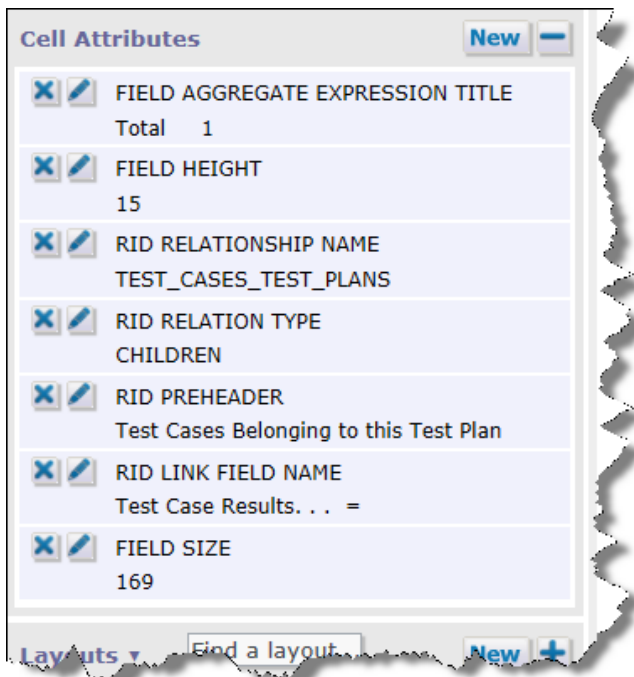
Cell attributes on the child record layout

Updating Related Issues as a Group

The feature on this page is being deprecated from version 9.0 onwards. The functionality will continue to work, but should not be configured on new sites. In due course, support for the feature will be removed entirely. The functionality has been superceded with the ability to handle related issues upon issue updates within Business Rules.

This feature allows you to apply updates to all issues related to the current issue, after you complete the updates to the current issue. The updates can be applied to a set of fields you place on a layout. You may optionally add a comment to each of the related issues as part of the updates. The setup for this involves a number of steps.

1. If you are using BI-LEVEL relationship groups, set the RG_UPDATE_BILEVEL_ONLY behavior setting to YES. For all other types of relationship groups, set the value of this behavior setting to NO
2. Ensure that you have a layout with a type of POST_EDIT_UPDATE defined. If your installation does not have a layout of this type defined, use the **Create and Maintain Layout Types** entry on the **Fields & Layouts** administration menu to define this
3. Add a layout with the POST_EDIT_UPDATE type to your system, either in the business area and project you require, or in a location from which it will be inherited
4. Place the fields on the layout that you want to be available during the update process for related issues. It should look something like the following screenshot. Note that the fields named STATUS, RESOLUTION and ASSIGNED_TO are the fields that will be updated in the related issues. It is mandatory that you place the embedded layout, LAYOUT.RELATED_ISSUE_DISPLAY on this layout. Also note that the field named RULE_POST_UPDATE_RG is defined as a custom field and has its help text defined as 4 - **Check the fields in the list below that you want to update to the same values in the issue being updated. The new value is shown.** Remember that you may need to set the RID RELATIONSHIP NAME, RID GROUP REFERENCE FIELD and the RID RELATIONSHIP TYPE layout cell attributes for the LAYOUT.RELATED_ISSUE_DISPLAY.

Layout to update related issues

5. When the user updates an issue that is a member of a relationship group, based on the above layout, the following screen appears:

Rendered layout when updating related issues

The user will first choose whether to update just the single issue or all the related issues. A comment may optionally be added at this time that will be applied to all the related issue updates. The user will see all the related issues and can check or uncheck the fields to update in the related issues. Lastly, notification options may be set, to notify only the users on the current issue or the users on all the related issues or notify only the users on the parent issue.

Mass Updates and Audit Trails

Related Issues and Mass Updates

When a user performs a mass update and some of the records are part of relationship groups, no attempt is made to update the related issues to any single item. The reason for this is that there could well be thousands of issues in the mass update, and each one of them may in some way be related to other issues, causing ExtraView to ask the questions for each issue as to whether the related issues should be updated.

Related Issues and Audit Trails

You may add an embedded layout prepared to display related issues to a History Layout. As explained in the previous section about preparing layouts for Related Issues, these layout types always begin with the character string RELATED_. This produces a related issue display at the point in time on the history record when the issue was updated, not at the current time. When you configure this layout on the History Layout, remember that you may also need to set the relationship group name and the relationship group type.

There is a special field named RELATIONSHIP_GROUP_TXNS. This may be added to the History layout to see all the relationship group operations that occurred prior to and including the listed transaction time. Note that the audit trail for related issues cannot be displayed if ABBREVIATED_HISTORY is set to NO as the abbreviated history does not use any layout for its display.

Related Issues and Report Output Types

When a report contains a related issue display, it is treated as follows for the various report output types. The reason for this is that some output types are deliberately formatted to make it easy to use the data as input to other processes or programs. Having a variable number of items of data from the related issues within these reports will make these operations impossible.

Report Output Type	Constraint
Browser	No constraints, all fields are displayed
Microsoft Excel	Fields from the related issue display are suppressed
Microsoft Word	Similar to other data presented on Microsoft Word reports, buttons such as the View and Edit buttons are suppressed, but all alphanumeric information is displayed.
Text	Fields from the related issue display are suppressed

Calculations on Related Issue Layouts

Field Aggregate Expressions (FAE's) permit calculations to be performed on a per-row basis on Related Issue Displays. The FAE's are added to fields within a Related Issue Display as layout cell attributes with a type of FIELD AGGREGATE EXPRESSION. There are also custom user exits for retrieval of data, accumulation of results, and the display of results. The display of the accumulated results is generated at the end of each group or collection of rows in a Related Issue Display, as an additional row with a user-defined title. The title is added as a layout cell attribute of type FIELD AGGREGATE EXPRESSION TITLE.

Creating and Using a FIELD AGGREGATE EXPRESSION

The following are the steps for creating and using a FAE:

Example

1. Create your Related Issue Display layout and save this in the usual way
2. Edit the Related Issue Display layout and select a column to which you want to apply an aggregate function. Click on the link button to add a new layout cell attribute
3. Choose the **FIELD AGGREGATE EXPRESSION - generate an aggregate value for this row** entry. Select the appropriate Value for your function from the list - *Count, Sum, Minimum, Maximum* or *Average*. The *Use User Custom Code* value is only valid if you have user custom code in place that will populate the results of the Field Aggregate Expression. See later on this page for details
4. The **Field value** should be set to a number - this will be tied to the FIELD AGGREGATE EXPRESSION TITLE on the Related Issue Display, as well as to the order in which multiple FIELD AGGREGATE EXPRESSION rows should be displayed. It is recommended you start with 1 for the first entry, and increment from there. The number you set must match the number you set in step 7
5. Save your changes to the Related Issue Display layout
6. Navigate to a parent issue that contains the Related Issue Display and verify the correctness of the results of the aggregate expression you created. It will not have any title or grouping at this moment, but you should see the results of the aggregation
7. To create a title on the Related Issue Display, navigate to the *add* or *edit* layout that embeds the Related Issue Display. On the cell that contains the Related Issue Display, you create an attribute of type FIELD AGGREGATE EXPRESSION TITLE where the **Value** is the Title of the row to be displayed, and the **Field value** should be a number (start with 1). If you have more than one FIELD AGGREGATE EXPRESSION defined on the Related Issue Display, this number is the order in which they will appear. The **Field value** should map to the **Field value** you selected in step 4
8. You can also optionally aggregate values while grouping together values within a field, and providing sub-totals or other attributes for each group. On the *add* or *edit* layout that contains the Related Issue Display, add a layout cell attribute to the Related Issue Display within the layout, of the type RID SORT ORDER. Select the field in the related issue display that you wish to group by, and select **Group by** equal to *Yes*. Now you will generate subtotals as well as a grand total. Clicking on a field title at the top of the related issue display will cause a refresh of the display, change the sort order and the sub-totals
9. Again navigate to your parent issue that contains the Related Issue Display and verify that you now see the title and the optional grouping.
10. In this example, we want to add totals to a Related Issue Display as shown in this screen shot:

Totalling a field on a Related Issue Display

The key steps are as follows:

1. On the Related Issue Display, add the following Layout Cell Attributes to the field that you want to total (*HOURS_INCURRED* in the example). Add a **FIELD AGGREGATE EXPRESSION** with a Value of **Sum** and a **Field value** of **1**. Save the changes
2. On the *edit* layout that contains the Related Issue Display, add two additional Layout Cell Attributes. These are in addition to other attributes that most probably exist to provide the RID RELATIONSHIP NAME and other factors for the layout.
   - RID SORT ORDER - This provides the initial sort order for the display
   - FIELD AGGREGATE EXPRESSION TITLE - This provides the title for the expression. Note that this should be the same value as set in the FIELD AGGREGATE EXPRESSION. The screen for the Layout Cell Attributes will be something similar to the following. Note that only the RID SORT ORDER and the FAE_TITLE pertain to the calculations. The other entries have their own purpose and function.

Layout Cell Attributes for the FAE

This gives the result shown in the screen shot above. Note the initial sort order. Also note that if you click on another field title within the Related Issue Display, there is a re-sort, and the totals for the field are automatically recalculated. This is shown below, where the display is now sorted on the *Category* field.

Remove?			Source Id	Category ▲	Customer	Customer Project	Date	Hours	Details
☐	QEdit	Edit	10227	Billable	BroadSoft	Initial Implementation	May 18, 2011	2	implemented Business Rules
☐	QEdit	Edit	10228	Billable	Hubspan	Initial Implementation	May 26, 2011	0.5	conference call regarding workflow diagram
☐	QEdit	Edit	10159	Billable	Mach	Lotus Notes integration	May 26, 2011	0.75	conference call to gather requirements
Total hours								3.25	
☐	QEdit	Edit		General/Admin			May 20, 2011	5	general IT tasks
Total hours								5	
☐	QEdit	Edit	10228	Nonbillable	Hubspan	Initial Implementation	May 26, 2011	1	answered question about where to download software
Total hours								1	
Total hours								9.25	

Resorting the display

Implementation Notes

- Expressions will only work with Related Issue Displays that contain 1,000 rows or less, if you implement grouping with aggregate functions
- The *Count* function works on all field types
- The *Sum, Average, Minimum and Maximum* functions work only on numeric display type fields - Number, Decimal and Currency
- The User Custom function requires custom code before it will achieve anything
- You can only group by one column in the related issue display RID SORT ORDER layout cell attribute
- You will get incorrect results if you group on a multi-valued field, or on a repeating record field, for calculations such as Average

Utilizing the User Custom Code Exits

This is an advanced topic and it is assumed the administrator is thoroughly familiar with programming within the User Custom Code environment of ExtraView. Please contact ExtraView Support for additional details on how to implement this feature.

Setting up field aggregate expressions is similar to the above, but there are a few additional steps.

The key step is that the aggregate expression performed in user custom code must be linked to a field on the Related Issue Display. This is achieved by creating a user defined field within the data dictionary, of type **Expression**. This may have any valid display type and will correspond with the returned value from the custom code. This provides your environment with a field name, a display title, security permissions and other attributes. You may have more than one custom aggregate function, with each being linked to a separate user defined **Expression** field.

To implement, you provide Java code for the 3 user custom exits for data selection, accumulated value computation and accumulated value display, and compile the code into the user custom class used in the installation.

The presence of FIELD AGGREGATE EXPRESSION layout cell attribute in the Related Issue Display layout directs the reporting logic to invoke three FAE-specific custom exits: data selection, value computation, and value display. In addition, the FAE-specific rows are added to the end of the report rows i.e. on the right for transposed Related Issue Displays and on the bottom for non-transposed Related Issue Displays.

Each FAE is associated with a specific field in the Related Issue Display, as part of the attributes on the Related Issue Display layout: this defines the row on a transposed Related Issue Display, or the column on a non-transposed Related Issue Display of the FAE result display at the end of the report. There may be more than one FAE associated with a field, resulting in additional columns on transposed Related Issue Displays or rows on non-transposed Related Issue Displays.

The three FAE user custom exits are as follows:

1. ucFAESelect: called during generation of SQL for the report to permit insertion of SQL into the detail report query for later consumption by the FAE value aggregation. **Signature:**

```
public String ucFAESelect (
            SesameSession session,
            Connection conn,
            String faeName,
            String ddName,
```

```
                    DetailReportElement re,
                    HashMap distinctTables,
                    String[] locParts,
                    EVHierarchyLevel ehl,
                    HashMap dataAlias,
                    String itemTableAlias,
                    String itemIdKey
        ) throws Exception
```

2. ucFAEAggregate: called for each row in the Related Issue Display to permit aggregation of the values in the report – this could include summation, averaging, or special operations as required. **Signature:**

```
    public void ucFAEAggregate (
                    SesameSession session,
                    Connection conn,
                    String faeName,
                    String ddName,
                    HashMap row,
                    ResultSet resultSet
        ) throws Exception
```

3. ucFAEDisplay: called once at the end of the report for each FAE in the Related Issue Display, to allow custom generation of display values for the FAE. **Signature:**

```
    public String ucFAEDisplay (
                    SesameSession session,
                    Connection conn,
                    String faeName,
                    String ddName
        ) throws Exception
```

Mass Update Report Layouts

There is a special report layout type named MASS_UPDATE_REPORT. This layout controls the fields that are visible to users who have permission to use the Mass Update utility. This layout does not exist by default within an ExtraView installation, but if one or more of these layouts are created with a layout type of this name, they will display the fields you place on the layout(s), as opposed to the fields which appear as these defaults:

- VIEW_BUTTON - View Button
- ID - The issue ID
- SHORT_DESCR - The title to the issue
- The specific fields chosen by the user to update
- AREA - The Business Area
- PROJECT - The Project

If you have not already created a MASS_UPDATE_REPORT layout type, begin by creating this layout type using the admin utility named **Create and Maintain Layout Types**. It should have a type of *Report*. Then, add a layout using this type, into the specific business area and/or project where you want users to see different fields on the mass update screen. You use the *Design Center* for this purpose. If you add the layout into the Global Area / Master Project, then the layout is used system-wide.

It is recommended that you place the ID field as the field in the first column of your MASS_UPDATE_REPORT layout. This will then appear along with the VIEW_BUTTON at the left. Following this will be the field(s) on your layout.

At the right-hand of your layout, the fields that you are updating will appear. If you do not place the ID field on your layout, it will still appear, but to the right-hand of the layout, along with the VIEW_BUTTON. Following this will be the fields that you are updating.

Search Layouts Within Add & Edit Layouts

There are two use cases in searching for records within an *add* or *edit* screen. Typically you are searching for records that exist within another Business Area:

1. Search for a single record and copy all, or a subset of its field values into the current record
2. Search for one or more records and copy field values from a number of records into new related issues.

The basic principal is that you embed a search layout into an *add* or *edit* screen layout. The results from the search are returned into a second embedded layout. The user can then select one or more results from this layout, and copy the required results into their

destination. There are two examples of this feature:

- [This page](#) gives assistance on how to configure this feature so that you can return a single search result into field values within the current issue
- [This page](#) gives assistance on how to configure this feature so that you can set up the results from one or more search results as related issues to the current issue.

Field Properties within Layouts

Not every field can be added successfully to every layout, and there are limitations regarding the placement of many fields on repeating record layouts. The following table gives an indication of all inbuilt fields and their restrictions in use. Note that User Defined Fields are capable of being placed on all layouts, subject to the field belonging to the correct type of *issue records* or *repeating row records* for the layout, and as long as the data dictionary entry for the field is defined correctly. Note that there are limitations with layout cell attributes between fields of different types. You may not have a layout cell attribute where a field of type *issue record* is dependent upon a field of type *repeating row*. The renderer, in this case, would not know which repeating row to use for its logic check.

Data dictionary field name- DATABASE field types	Notes
ALT_ID	Any layout except repeating record layouts
AREA	All *Add* and *Edit* layouts belong to a specific Business Area. If the AREA field is placed on the layout, then the field and its values will be presented in the normal way. If the AREA field is not placed on the layout, then ExtraView processes the layout as if the value of the Business Area is set to the same value to which the layout belongs. This helps the usability for many applications where there is more than one Business Area, but you want to restrict the user to only working within a single Business Area at a point in time, for example, when you have a popup window to enter a child issue which will be placed within a specific Business Area. If you want to restrict the user to only being able to address a subset of the available Business Areas within an installation, you may use the FIELD HIDE VALUES and the FIELD VISIBLE VALUES to restrict the displayed values.
ASSIGNED_TO	Any layout except repeating record layouts
ATTACHMENT_ID	Internal use only
ATTACH_CONTENT_TYPE	Will be placed automatically on the layout as part of the attachment record. Controlled by security permissions
ATTACH_CREATED_BY_USER	Will be placed automatically on the layout as part of the attachment record. Controlled by security permissions
ATTACH_DATE_CREATED	Will be placed automatically on the layout as part of the attachment record. Controlled by security permissions
ATTACH_FILE_DESC	Will be placed automatically on the layout as part of the attachment record. Controlled by security permissions
ATTACH_FILE_NAME	Will be placed automatically on the layout as part of the attachment record. Controlled by security permissions
ATTACH_FILE_SIZE	Will be placed automatically on the layout as part of the attachment record. Controlled by security permissions
ATTACH_PATH	Will be placed automatically on the layout as part of the attachment record. Controlled by security permissions
ATTACH_THUMBNAIL	Will be placed automatically on the layout as part of the attachment record. Controlled by security permissions
AVAILABLE_FOR_DOWNLOAD	Not available for any layout
CATEGORY	Any layout except repeating record layouts
CONTACT	Any layout except repeating record layouts
DATE_CLOSED	Read-only on the *edit* screen only

DATE_CLOSED_SINCE	Reporting filters only
DATE_CODE_FREEZE	Not available for any layout
DATE_CREATED	Read-only on *add* and *edit* screens. Not to be used on repeating records
DATE_CREATED_DAY	Not used on *add* screens, *edit* screens or repeating records. Only used for reporting purposes
DATE_CREATED_MONTH	Not used on *add* screens, *edit* screens or repeating records. Only used for reporting purposes
DATE_CREATED_SINCE	Not used on *add* screens, *edit* screens or repeating records. Only used for reporting purposes
DATE_CREATED_TRUNC	Not used on *add* screens, *edit* screens or repeating records. Only used for reporting purposes
DATE_CREATED_WEEK	Not used on *add* screens, *edit* screens or repeating records. Only used for reporting purposes
DATE_CREATED_YEAR	Not used on *add* screens, *edit* screens or repeating records. Only used for reporting purposes
DATE_FIRST_CUSTOMER_SHIP	Not available for any layout
DATE_LAST_STATUS_CHANGE	Read-only on *edit* screen. Not used on *add* screen. Not to be used on repeating records
DATE_LAST_STATUS_CHANGE_SINCE	Reporting filters only
DATE_RELEASE_TO_QA	Not available for any layout
DAYS_IN_STATUS	Reports only
DAYS_OPEN	Reports only
HIST_RANGE_END	May only be used as a filter on a search layout
HIST_RANGE_START	May not be used on any layout. Only used in the CLI
ID	Read-only on *add* and *edit* screens. Not to be used on repeating records
ITEM_ID	Should not be used on any layout
LAST_CHANGE_USER	Read-only on *edit* screen. Not used on *add* screen. Not to be used on repeating records
LAYOUT	Not available for any layout
LAYOUT_TYPE	Not available for any layout
MODULE_ASSIGNED	This field is deprecated and should not be used
MODULE_DATE_CREATED	This field is deprecated and should not be used
MODULE_ID	Not to be used on repeating records
MODULE_NAME	Should not be used on layouts
MODULE_PRODUCT	This field is deprecated and should not be used
MODULE_STATUS	This field is deprecated and should not be used
MODULE_TIMESTAMP	This field is deprecated and should not be used
MODULE_TITLE	Should not be used on layouts
MODULE_TYPE	This field is deprecated and should not be used
MODULE_VERSION	This field is deprecated and should not be used
MONTHS_IN_STATUS	Reports only
MONTHS_OPEN	Reports only

ORIGINATOR	Any layout except repeating record layouts
OWNER	Any layout except repeating record layouts
PRIORITY	Any layout except repeating record layouts
PRIVACY	Any layout except repeating record layouts
PROBLEM_RELEASE_ID	Should not be used on any layout
PRODUCT_LINE	Any layout except repeating record layouts
PRODUCT_NAME	Any layout except repeating record layouts
PRODUCT_NAME_HIST	Should not be used on any layout
PROJECT	All *Add* and *Edit* layouts belong to a specific Project within a Business Area. If the PROJECT field is placed on the layout, then the field and its values will be presented in the normal way. If the PROJECT field is not placed on the layout, then ExtraView processes the layout as if the value of the Business Area is set to the same value to which the layout belongs. This helps the usability for many applications where there is more than one Project within a Business Area, but you want to restrict the user to only working within a single Project at a point in time. If you want to restrict the user to only being able to address a subset of the available Projects within a Business Areas, you may use the FIELD HIDE VALUES and the FIELD VISIBLE VALUES to restrict the displayed values.
RELATIONSHIP_GROUP_ID	Should not be used on any layout
RELATIONSHIP_GROUP_OWNER	Should not be used on any layout
RELATIONSHIP_GROUP_TITLE	Should not be used on any layout
RELATIONSHIP_GROUP_TXNS	This field is only valid on a layout of type HISTORY.
RELATIONSHIP_GROUP_TYPE	Should not be used on any layout
RELATIONSHIP_GRP_PARENT_ID	Should not be used on any layout
RELEASE	Should not be used on any layout. The LAYOUT.RELEASE field may be embedded on the *add* and *edit* screens to support repeating records
RELEASE_ASSIGNED_TO	Repeating records and reports only
RELEASE_DATE_CREATED	Repeating records and reports only
RELEASE_DIRECTORY	Should not be used on any layout
RELEASE_FIXED	Repeating records and reports only
RELEASE_FOUND	Repeating records and reports only
RELEASE_FOUND_HIST	Should not be used on any layout
RELEASE_OWNER	Repeating records and reports only
RELEASE_PRIORITY	Repeating records and reports only
RELEASE_PRODUCT	Repeating records and reports only
RELEASE_RESOLUTION	Repeating records and reports only
RELEASE_SEVERITY	Repeating records and reports only
RELEASE_STATUS	Repeating records and reports only
RELEASE_TIMESTAMP	Read only on repeating records and reports
RELEASE_TYPE	Deprecated – do not use
REL_GRP_DATE_CREATED	Should not be used on any layout
REL_GRP_TIMESTAMP	Should not be used on any layout
RESOLUTION	Any layout except repeating record layouts

SEVERITY_LEVEL	Any layout except repeating record layouts
SHORT_DESCR	Any layout except repeating record layouts
START_DATE	Reports only. Note that the data dictionary setting **"Filter criteria"** must be set to **Yes** for this field to work
START_UPDATE	Reports only. Note that the data dictionary setting , **"Filter criteria"** must be set to **Yes** for this field to work
STATUS	Any layout except repeating record layouts
STATUS_HIST	Should not be used on any layout
STOP_DATE	Reports only. Note that the data dictionary setting **"Filter criteria"** must be set to **Yes** for this field to work
STOP_UPDATE	Reports only. Note that the data dictionary setting **"Filter criteria"** must be set to **Yes** for this field to work
TIMESTAMP	Read-only on *add* and *edit* screens. Not to be used on repeating records
TIMESTAMP_DAY	Not used on *add* screens, *edit* screens or repeating records. Only used for reporting purposes
TIMESTAMP_MONTH	Not used on *add* screens, *edit* screens or repeating records. Only used for reporting purposes
TIMESTAMP_SINCE	Not used on *add* screens, *edit* screens or repeating records. Only used for reporting purposes
TIMESTAMP_TRUNC	Not used on *add* screens, *edit* screens or repeating records. Only used for reporting purposes
TIMESTAMP_WEEK	Not used on *add* screens, *edit* screens or repeating records. Only used for reporting purposes
TIMESTAMP_YEAR	Not used on *add* screens, *edit* screens or repeating records. Only used for reporting purposes
WEEKS_IN_STATUS	Reports only
WEEKS_OPEN	Reports only

Data dictionary field name- LABEL field types	Notes
CC_EMAIL	Should not be used on any layout. This field is automatically added to the *add* and *edit* layouts with the notification fields
DELETE_BUTTON	Reports only
EDIT_BUTTON	Reports only. Note that this field is automatically added to any custom report
EMAIL	Should not be used on any layout
FILTER_CHILD_VALUES	Should not be used on any layout
GENERATE_EMAIL	Should not be used on any layout. This field is automatically added to the *add* and *edit* layouts with the notification fields
HISTORY_BUTTON	Reports only
KEYWORD	Report filters only
PROBLEM	Should not be used on any layout
RELATIONSHIP_GROUP_LINK	Should not be used on any layout
REPORT	Should not be used on any layout
REPORT_BY	Should not be used on any layout

REPORT_NAME	Should not be used on any layout
REPORT_OUTPUT	Should not be used on any layout
REPORT_TYPE	Should not be used on any layout
SORT	Should not be used on any layout
VIEW_BUTTON	Reports only. Note that this field is automatically added to any custom report

Data dictionary field name-SCREEN field types	Notes
ADD_PROBLEM	Should not be used on any layout
ADD_PROBLEM_SUMMARY	Should not be used on any layout
ADMINISTRATION	Should not be used on any layout
ATTACHMENT	This is the field that places attachment records on *add* and *edit* screens. It can also be placed on detailed reports and Quicklists, but not on report filter layouts. Security permissions should be set for the individual fields such as ATTACH_FILE_NAME and ATTACH_FILE_SIZE to control their visibility
BATCH_COMMANDS	Should not be used on any layout
COMPANY_NAME	Should not be used on any layout
CUSTOM_EMAIL	Should not be used on any layout
HOME_PAGE	Should not be used on any layout
PAGE_SIZE	Should not be used on any layout
PROBLEM_RELEASE_DELETE	Should not be used on any layout
PROBLEM_SUMMARY_EDIT	Should not be used on any layout
PROMO	Should not be used on any layout. This field is automatically included in the sign on screen, and you can provide HTML within the title of the field that will be rendered within the page. This HTML may include images and JavaScript
QUICK_LIST	Should not be used on any layout
RELATIONSHIP_GROUP	Should not be used on any layout
SEARCH_REPORT	Should not be used on any layout
SECURITY_GROUP	Should not be used on any layout
SECURITY_KEYS	Should not be used on any layout
SECURITY_PERMISSION	Should not be used on any layout
SIGN_ON	Should not be used on any layout
STATUS_CHANGE	Should not be used on any layout
SYSTEM_LOG_TYPE	Should not be used on any layout
UDF	Should not be used on any layout
USER_ACCOUNTS	Should not be used on any layout

Data dictionary field name- SESSION and SPECIAL field types	Notes
USER	Should not be used on any layout
SYSDATE	Should not be used on any layout

Adding Shaded Regions to Layouts

Layouts with a plain background can look bland. You can improve the look of *add* and *edit* layouts by using shaded regions. Shaded regions allow you to set a background color or a background image to a contiguous group of rows within a layout. The shaded region can also be given a legend. To set up a shaded region on an *add* and *edit* layout, use the following procedure:

- Create two fields in the data dictionary.
  - The first field is used to mark the beginning of the shaded region on the layout. The field must use a naming convention, beginning with the characters SHADE_PRE_. For example, SHADE_PRE_1 is a valid name for the field. The title should be a single space character and the display type should be **Custom**. Give the field read-only permission for each role that will use the shaded region
  - The second field is used to mark the end of the shaded region on the layout. The field must use a naming convention, beginning with the characters SHADE_POST_. For example, SHADE_POST_1 is a valid name for the field. The title should be a single space character and the display type should be **Custom**. Give the field read-only permission for each role that will use the shaded region
- Place the SHADE_PRE_ and the SHADE_POST_ fields on the layout, giving each a colspan that extends the entire width of the layout. Obviously, the SHADE_PRE_ field is placed at the beginning of the region and the SHADE_POST_ field is placed at the end of the region
- To provide a legend for the shaded region, add an FIELD HTML MODIFIER layout cell attribute to the SHADE_PRE_ field. This attribute should be of the form **LEGEND:Title to the Region**
- If you require to add a legend in a language other than English, add an attribute in the form **LEGEND_ALT:Title to the Region**. If the user's locale is not English, this legend will be used rather than the title defined in LEGEND. However, if a language other than English is used within the LEGEND attribute, and no LEGEND_ALT attribute is defined, then this is used as the title to shaded regions. Unlike most of ExtraView's localization capability, this does not provide the flexibility to add any number of localized language legends to the legend, and is limited to a single language beyond the default language
- To provide a background color to the shaded region, add a FIELD HTML MODIFIER layout cell attribute to the SHADE_PRE_ field. This attribute should be of the form **BGCOLOR:#nnnnnn** where **#nnnnnn** is a valid hexadecimal color
- To provide a background image or other CSS style for the shaded region, add an FIELD HTML MODIFIER layout cell attribute to the SHADE_PRE_ field. This attribute should be of the form **STYLE:'background-image:url(../locales/en_US/images/images_light_blue/TableBg.png)'**
- To add a CSS style to the legend, add a FIELD HTML MODIFIER layout cell attribute to the SHADE_PRE_ field. This attribute should be of the form **CSSCLASS:classname**. There are two inbuilt CSS styles named *legendSmall* and *legendLarge* that can be used for the **classname**.
- This will have the following effect on a layout. Note the legend **Customer Details** and the light yellow background around the fields:

Shading a region on a layout

- You may nest SHADE_PRE_ and SHADE_POST_ fields to create multiple shaded regions within each other on a form. Here is an example:

Multiple shaded regions on a layout

Layout Cell Attribute Summary of HTML MODIFIERs

LEGEND:*title*	This HTML MODIFIER provides the title to the shaded region
LEGEND_ALT:*title*	This is an alternative title which is used if the user's locale is not English. This allows for a non-English title to be set for the layout
CSSCLASS:*classname*	This is a CSS class name that is applied to LEGEND and LEGEND_ALT. The recommended value is **legendLarge**
BGCOLOR:*color*	BGCOLOR is added as an HTML MODIFIER to provide the background shading color for the region. The *color* may be an HTML color name or a hex value such as #123456
STYLE:*style*	This HTML MODIFIER is used as an alternative to BGCOLOR. You should not use both attributes. The STYLE attribute adds a CSS style to the background of the shaded region. Any valid CSS may be entered.

Combining Shaded Regions with Folds / Pages and HTML Pre / Post Fields

There are limitation on how shaded regions may be nested within Accordion Folds, Paging and HTML Pre / Post fields. If the shaded region SHADE_PRE and SHADE_POST fields are on the main *add* or *edit* layouts, **and** they are **outside** the scope of Accordion Folds, Paging and HTML Pre / Post fields they are always rendered. If the SHADE_PRE and SHADE_POST fields are **within** the scope of the other fields, they will not render. You can overcome this limitation by using sub-layouts, with the SHADE_PRE and SHADE_POST fields, plus the fields the enclose within the sub-layout.

Adding Accordion Folds to Layouts

- [Add new comment](#)

Accordion Folds on *Add* & *Edit* Layouts

Accordion folds may be added to your *Add* and *Edit* layouts to provide a flexible user interface, where sections will open and close at the click of a button. The accordion folds may be decorated with titles and themed to your own choice. You may also choose whether they render on the initial layout in a closed or an open state. You can include any number of accordion folds within a layout. Compare the use of accordion folds with [paged layouts](#). Accordion folds allow multiple sections within a form to be open and closed at will by the end user, whereas paged layouts have multiple sections, only one of which may be visible at any one time. Note that when accordion folds are placed on layouts, they have no effect on the requiredness of fields they contain. A required field is required, whether the accordion fold is expanded or collapsed.

Closed Accordion Fold

Open Accordion Fold

To set up an accordion fold on an *Add* or *Edit* layout, use the following procedure:

- Create two fields in the data dictionary.
    - The first field is used to mark the beginning of the accordion fold on the layout. The field must use a naming convention, beginning with the characters FOLD_PRE_. For example, FOLD_PRE_1 is a valid name for the field. The title should be a single space character and the display type should be **Custom**. To provide a legend for the accordion fold, use the Help Text field. Providing the legend for the fold works best if defined in this manner, if you intend to use the same fold in several places throughout the installation. You may alternatively define the title within a layout cell attribute on a layout that uses the fold. This provides the flexibility of being able to reuse the same FOLD_PRE_ and FOLD_POST_ fields on many layouts, defining the title for each within the layout, and not in the data dictionary. If no HTML modifier with the syntax **LEGEND:fold title** is defined, the title in the data dictionary is used. Give the field read-only permission for each role that will use the shaded region
    - The second field is used to mark the end of the shaded region on the layout. The field must use a naming convention, beginning with the characters FOLD_POST_. For example, FOLD_POST_1 is a valid name for the field. The title should be a single space character and the display type should be **Custom**. Give the field read-only permission for each role that will use the shaded region
- Place the FOLD_PRE_ and the FOLD_POST_ fields on the layout, giving each a colspan that extends the entire width of the layout. Obviously, the FOLD_PRE_ field is placed at the beginning of the region and the FOLD_POST_ field is placed at the end of the region
- If you place the fields to be rendered within a fold in an embedded layout, between the FOLD_PRE_field and the FOLD_POST_field, and you have the HTML modifier COLLAPSE, the fields within the accordion fold are not rendered when the layout is first displayed, but only when the user opens the fold. This is shown here within the design center:
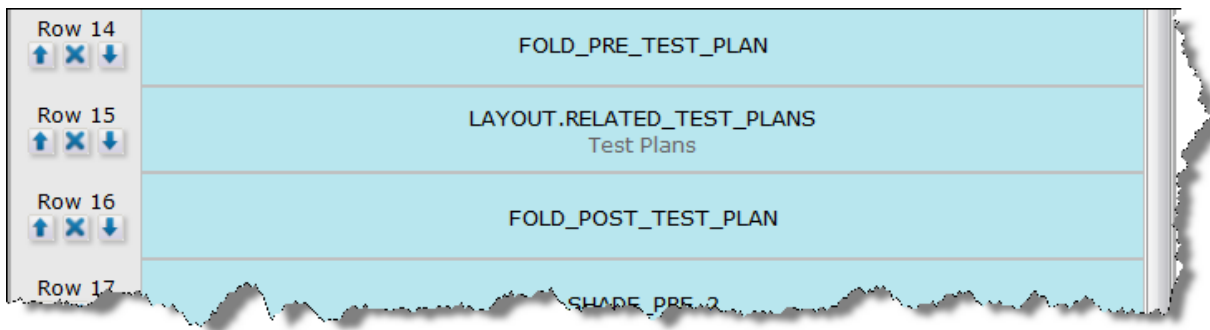
- The advantage here is that if you have many fields within many embedded layouts it may take a considerable amount of time to render the screen to the end user. Using this technique will save the rendering time when the *add* or *edit* screen is first displayed, but will incur a small amount of time for the end user as they open each fold. The overall performance to the end user seems much improved by breaking down the initial delay into several much smaller delays.

Fold Configuration Options

The following layout cell attributes may be applied as HTML MODIFIERs to the FOLD_PRE_ field, to provide styling and positioning of the fold as it is rendered on *add* or *edit* screens.

BGCOLOR:*color*	Provides a background color to the accordion fold. To implement, add an HTML MODIFIER layout cell attribute to the FOLD_PRE_ field. This attribute should be of the form BGCOLOR:#nnnnnn where #nnnnnn is a valid hexadecimal color or HTML color name
BORDER_COLOR:*color*	To provide a border color to the accordion fold, add an HTML MODIFIER layout cell attribute to the FOLD_PRE_ field. This attribute should be of the form BORDER_COLOR:#nnnnnn where #nnnnnn is a valid hexadecimal color or HTML color name
COLLAPSE	By default, the accordion fold will be expanded when rendered on the layout, unless you include an HTML MODIFIER with a value of COLLAPSE, which will collapse the relevant accordion fold upon rendering and upon screen refreshes. Ajax and JavaScript refreshes will not affect the collapsed or expanded state of the accordion fold. Note that rules may be used to control the oen or closed state of the fold
FOLD_USER_IMG_PATH:*path*	By default the standard **dn-arrow.png** and **rt-arrow.png** images from the appropriate image folder set in the IMG_HOME behavior setting are used on the accordion fold, depending upon whether the fold is collapsed or expanded. If you wish to use your own images, create a new folder called **user_images** in the images directory and place your images within this, named in the same manner. Then, include an HTML MODIFIER with a value of FOLD_USER_IMG_PATH, and the files in that folder will be used instead of the defaults
HDR_BGCOLOR:*color*	To color the background to the text on the accordion fold, add an HTML MODIFIER layout cell attribute to the FOLD_PRE_ field. This attribute should be of the form HDR_BGCOLOR:#nnnnnn where #nnnnnn is a valid hexadecimal color
HDR_TXT_COLOR:*color*	To color the text provided as the title to the accordion fold, add an HTML MODIFIER layout cell attribute to the FOLD_PRE_ field. This attribute should be of the form HDR_TXT_COLOR:#nnnnnn where #nnnnnn is a valid hexadecimal color
LEGEND:*title*	An HTML MODIFIER of the form **LEGEND:This is my fold** will place the text on the header to the fold. The text will be visible when the fold is both closed and open. Clicking on the text or the surrounding area is what causes the fold to open and close
NO_INDENT	By default, the accordion fold will be indented to line up with the column of the first field on the layout, as opposed to the column of the first field's label. If you include an HTML MODIFIER with a value of NO_INDENT, the accordion fold will not be indented
SIZE:*nn*	By default, the width of the accordion fold is determined by the rendering of the relevant layout, but you may manually set its width with the SIZE layout cell attribute. The value will be converted to a pixel width for the Fold Section Note that by overriding the default width handling, the Administrator is responsible to ensure that the fields within the accordion fold fit within the rendered section. This factor is also affected by a user's personal **Text size** setting, so be sure to check the look of your layouts with the **small**, **medium** and **large** text sizes

Controlling Fold States with Business Rules

As stated above, the initial state of the contents of an accordion fold is that all the contents are visible, unless you have placed a layout cell attribute of the type HTML MODIFIER and a value of **COLLAPSE** on the **FOLD_PRE_xxxx** field. It is often useful to set the initial state of the contents programmatically to open or closed, based upon one or more values within the issue you are adding or updating. This is accomplished with four components:

- The entire contents of the fold must be placed within an embedded layout placed on the *add* or *edit* layout
- The **FOLD_PRE_xxxx** and **FOLD_POST_xxxx** fields must be placed on the *add* or *edit* layout surrounding the embedded layout.

Rules may also be used to open and close folds, depending on conditions you set. For example, when a user opens one fold, you may close other open folds. In this way you might configure a screen so that only a single fold is open at any one point in time. See the page on Rules & Accordion Folds for details.

Accordion Folds & Required Fields

When you first generate an *add* or *edit* screen that contains accordion folds in the closed state, the fields within the fold are not rendered to the browser immediately, but are only rendered when the user opens the fold. This is for performance reasons - consider that you might have dozens of folds containing hundreds of fields on a large, complex layout. In this instance the initial rendering is optimized to only send the folds themselves and not their content to the browser. Upon opening an individual fold, an Ajax call to the server gets the data for just that fold, typically fetching a modest number of fields in a short time. However, some closed folds may contain required fields and the user might never open these folds. Even if these fields are not in the initial load of the screen form, ExtraView will check for any required fields within these folds when the user submits the issue and ensure they are entered by the user.

Accordion Folds & Paged Layouts

It is not recommended to nest Accordion Fold fields within Paged layouts, or vice versa. This does not provide a very usable user interface, and you might encounter some unexpected behavior.

Accordion Folds on Detailed Report & History Layouts

Accordion folds may be added to Detailed report and History layouts. They differ in their configuration from folds implemented on *add* and *edit* layouts. There are several reasons for this:

1. When the user creates a Detailed report, it might contain different issues from different Business Areas, each with different saved layouts
2. On both Detailed reports and History output, you do not want to open the same fold in each issue / section of the output when you click on a fold within a single issue / section

Further, for technical reason, the folds must be inserted as the reports are being rendered to the browser, long after the layouts are processed. At this time, there is no access to layout cell attributes to provide the same configuration options as the folds on *add* and *edit* layouts. Folds on Detailed reports and History layouts are configured as follows:

- Create two fields in the data dictionary.
  - The first field is used to mark the beginning of the accordion fold on the layout. The field must use a naming convention, beginning with the characters REPORT_FOLD_PRE_. For example, REPORT_FOLD_PRE_1 is a valid name for the field. The title should contain the text you wish to see on the fold, and the display type should be **Custom**. You must set **Select for Reports** and provide permission to read the field
  - The second field is used to mark the end of the shaded region on the layout. The field must use a naming convention, beginning with the characters REPORT_FOLD_POST_. For example, REPORT_FOLD_POST_1 is a valid name for the field. The title should be a single space character and the display type should be **Custom**. You must set **Select for Reports** and give the field read-only permission for each role that will use the shaded region
- Place the REPORT_FOLD_PRE_ and the REPORT_FOLD_POST_ fields on the Detailed report layout, giving each a colspan that extends the entire width of the layout. Obviously, the REPORT_FOLD_PRE_ field is placed at the beginning of the region and the REPORT_FOLD_POST_ field is placed at the end of the region
- The folds are always initialized in the open state
- No layout cell attributes are used on the layouts for these fields and will be ignored if set.
- Business rules cannot be used to control the folds

Adding Paged Layouts

Paged layouts are ideal when you have a very large number of fields on a layout and wish to separate them into logical groups which the user can page forward and back through, using forward and backwards buttons. Compare this to the use by users, of [accordion folds](#) that open and close individual sections of the screen. Paged layouts are also ideal to use on mobile device layouts where there is only room on a device to display a handful of fields at a time, before the user has to scroll up and down.
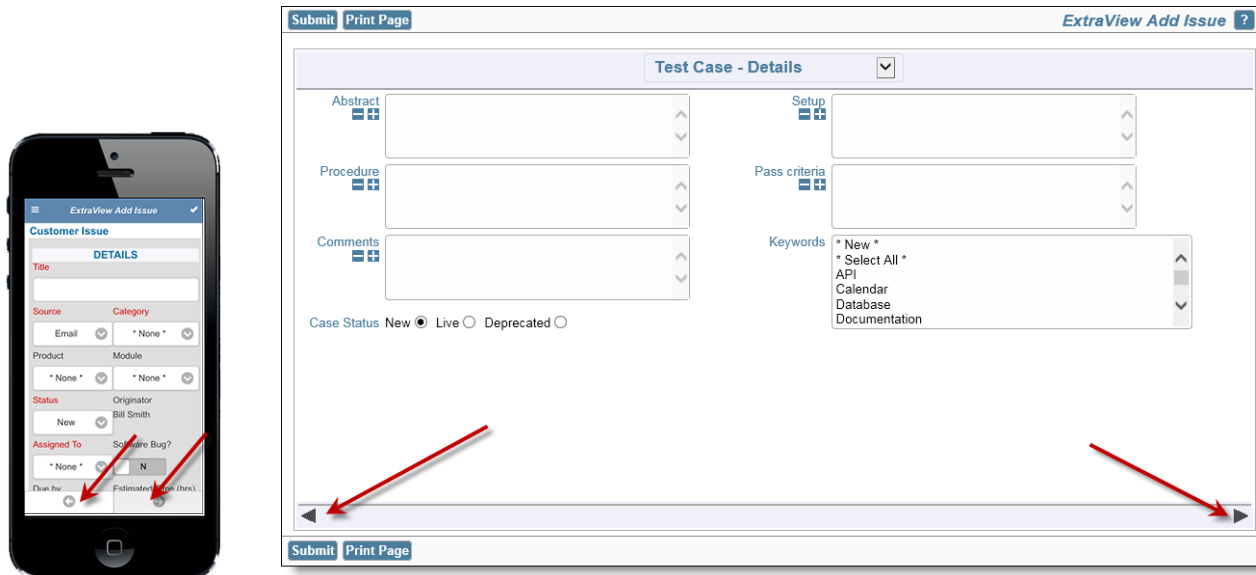
Page layouts on mobile and desktop clients

Paged layouts have several important criteria associated with their use:

- Only one page within a sequence may be visible at any one time. This is termed the **active page**.
- Each page is defined with two fields, one beginning with the characters PAGE_PRE_ and one beginning with the characters PAGE_POST_. For example, fields defining a page might be named PAGE_PRE_ORDER_INFO and PAGE_POST_ORDER_INFO. An obvious point is that each *add* or *edit* layout that contains pages, should have at least two pages defined. The two fields surround all the fields or embedded layouts that compose each page
- Each page within the sequence may optionally have a title. This is defined within the field's data dictionary entry within the Help Text. If no title is defined, pages will be titled **Page 1**, **Page 2**, etc. The title is displayed at the top of the page within the layout
- The PAGE_PRE_ and the PAGE_POST_ fields must extend the complete width of the layout that uses them
- Each page may contain any number of fields or may contain an embedded layout. Note that you might have some fields before or after the last page you have defined. If you configure the layout in this way, then the fields before the first PAGE_PRE_ and after the last PAGE_POST_ fields will always be visible to the end user
- A complete page, within a sequence of pages, may have a *visible if* layout cell attribute associated with it, allowing the page to be skipped upon data entry, dependent on the values controlling the *visible if* attribute chosen by the end user
- The order of the pages defined within the *add* or *edit* layout is significant. The order defines the sequence with which the pages will be presented to the end user
- When an *add* or *edit* screen is first opened by an end user, the first visible page is opened. All other pages will be closed. A visible page is one where at least a single field is visible - fields are subject to the usual constraints such as security permissions
- Care should be taken when using allowed value type relationships, to ensure the parent field is set on the same page, or a prior page to the child field. It would be difficult for the end user to navigate back up the pages to enter a child value in a field
- If the current page which a user is viewing becomes invisible, due to workflow constraints, the first, visible previous page becomes the active page and is made visible
- Care should also be taken with required fields within pages. These fields will still be required when the user submits the issue, even if they are not on the current active page
- Nested pages are not supported. This implies that each PAGE_PRE_ field must be followed by its corresponding PAGE_POST_ field, before another PAGE_PRE_ field is encountered
- You cannot create pages within "Selected By" layouts. This is not a supported configuration
- You cannot create pages nested within accordion folds. This is not a supported configuration
- You cannot create pages nested within shaded regions. This is not a supported configuration
- You can style the pages with the use of layout cell attributes on the PAGE_PRE_ field within the layout. The potential layout cell attributes are implemented as HTML Modifier attributes, that begin with the following characters and end with the appropriate value:

BGCOLOR:*color*   The background color of the header and footer rows. The default color is the value of the behavior setting named BG_ALT_COLOR. Any valid HTML color name or hex value in

the form #123456 may be used

BORDER_COLOR:*color*	The color of the border around the paged layout. The default value is #A3A9B0
COLOR:*color*	The color of the text of the title in the header of the page. The default color is the value of the behavior setting named LABEL_COLOR
HIGHLIGHT_BGCOLOR:*color*	The background color of the text of the title in the header of the page, when the user places their mouse over the page title. The default color is #666666
HIGHLIGHT_COLOR:*color*	The color of the text of the title in the header of the page, when the user places their mouse over the page title. The default color is the value of the behavior setting named TAB_FONT_ON_COLOR
PAGE_BACK_IMG:*image-name*	The default image is named left-*arrow.png* and is a standard image within the *images* folder that your installation is using - the path in the IMG_HOME directory specifies where the image is stored. You may place any image in the value to the layout cell attribute, but you should use the full path to the image, wherever it is stored.
PAGE_FWD_IMG:*image-name*	The default image is named *rt-arrow.png* and is a standard image within the *images* folder that your installation is using - the path in the IMG_HOME directory specifies where the image is stored. You may place any image in the value to the layout cell attribute, but you should use the full path to the image, wherever it is stored.
PAGE_HEIGHT:*nn*	Layouts within ExtraView are typically auto-sized. This attribute will size the page height to the number of pixels in the value of the attribute. The greatest use of this attribute is to use the same value on each PAGE_PRE_ field you define in the layout, thereby ensuring all pages are rendered at exactly the same height, giving a pleasing effect for the end user.
	If the value you provide is less than the height required by the fields within the page, the page will render to the height required by the fields.
	Tip: Apply the layout cell attribute to the tallest page within your layout first, to make sure you have a suitable value. Then apply the same value to all the other pages in the layout.
PAGE_WIDTH:*nn*	Layouts within ExtraView are typically auto-sized. This attribute will size the page width to the number of pixels in the value of the attribute. The greatest use of this attribute is to use the same value on each PAGE_PRE_ field you define in the layout, thereby ensuring all pages are rendered at exactly the same width, giving a pleasing effect for the end user.
	If the value you provide is less than the width required by the fields within the page, the page will render to the width required by the fields.
	Tip: Apply the layout cell attribute to the widest page within your layout first, to make sure you have a suitable value. Then apply the same value to all the other pages in the layout.

Paged Layouts & Accordion Folds

It is not recommended to nest Accordion Fold fields within Paged layouts, or vice versa. This does not provide a very usable user interface, and you might encounter some unexpected behavior.

Adding Ruler Areas to Layouts

Ruler areas or ruler fields are used to generate headings within the *add* and *edit* screens. Ruler fields provide spacing and provide an opportunity to place text as assistance to the user.

Ruler fields on the add screen

Data dictionary field	Purpose
Name	This defines the name of the field. For example, a field named RULE_WORKFLOW is valid. You may define any number of ruler fields, as long as the name for each one begins with the characters RULE_.
Display Type	Must be **Custom**
Title	Typically this is a space character
Help Text	This is the text to place upon the ruler on the *add* or *edit* screen.

The standard distribution of ExtraView that ships with the user custom class named CustomCodeBase.java contains this functionality.

AJAX Techniques

With traditional web-based applications, it was often necessary to refresh entire screens to repopulate individual fields according to various triggering of business rules, such as populating a child list with values based upon a change in its parent value. ExtraView has maintained a rich set of features in addition to the management of parent-child relationships, such as **visible if** and **required if**. As powerful as many of these features were, they required ExtraView to execute the feature by returning the values from the browser to the server where the logic was used to send a complete page of information back to the browser. This is known as a "screen refresh" and whilst it is capable of executing complex logic, the process takes time, from a half-second to four or more seconds, depending on a variety of factors. These factors include the speed of the network and the speed of the hardware as well as the size and complexity of the ExtraView layouts that are being processed.

Using a technology known as Ajax, most screen refreshes are eliminated and the processing time is typically an order of magnitude less than the time for an entire screen refresh. The key places where ExtraView takes advantage of Ajax while processing issues on *add* and *edit* layouts are:

- To process layout cell attributes such as FIELD VISIBLE IF and FIELD REQUIRED IF. This includes the use of visible if to control entire embedded layouts
- To process layouts controlled by a tab or list field using the LAYOUT SELECTED mechanism
- To execute business rules that result in changes to values within fields on forms based upon field selections or data input

AJAX Limitations

The screen layout renderer in ExtraView always removes rows of fields where there is no visible content. This avoids unsightly gaps in the screen for yor users. If you have a row that is entirely suppressed, and there are one or more fields on that row which are invisible due to a visible if cell attribute, then the row is not generated at all when the field or fields do not meet the criteria to become visible.

In this circumstance where the entire row is not rendered, there is insufficient information in the layout to use Ajax to populate the field when it becomes visible. As a result, ExtraView will revert to a full screen refresh. Thus the functionality is correct, but Ajax is not and cannot be used to make the field visible. If keeping the speed of the Ajax refresh is important to your application, consider altering your layout to include a field which will be visible at all times on the row of your layout so that something is always visible. This could even be a label field with a blank value. Although there is no obvious effect from this, it is actually visible.

Detailed Report Layout Selection

Detailed reports are generated on a different basis, in that each individual issue on a detailed report will use the detailed report layout associated with the issue's business area and project. The user's current role is also used in the selection. Inheritance is still used, but the net effect is that different issues may use different layouts, and will use the layout most associated with the issue.

Quicklist Report Layout Selection

A Quicklist may contain individual issues that belong to many different business areas and projects. Only one layout is selected for the entire report. This is selected from the user's current business area, project and role, unless one of the following conditions exists:

- If a filter selecting a single business area (AREA) is chosen, ExtraView will select the layout associated with PROJECT 0 of this area. Inheritance is used if no layout has been defined at this level
- If a filter selecting a single business area and a single project is chosen, then ExtraView will select the layout associated with this specific area and project. Inheritance is used if no layout has been defined at this level
- If zero or more than one business area is chosen, and/or the project selected conflicts with the business area, the layout associated with the user's current business area and project is selected

Attachment Layouts

Security Permission Keys

Field/Security permission key	Purpose
ATTACHMENT	This key controls the overall ability to create attachments. Write permission to this key is needed by the user role, for attachments to be able to be created or maintained. If there is no read permission, the attachment layout is not visible. This field is the only field that should be placed on reports where you want to display attachments, such as the detailed report. The remaining fields for the attachment will be placed onto the report by ExtraView, according to whether the field has read permission. When you have read permission to attachments on reports, a View attachment button is automatically placed on the report, allowing you to view the contents of all attachments to an issue directly.
ATTACHMENT_ADD	This controls the Add button for attachments. Write access is needed to control this
ATTACHMENT_DELETE	This is to enable and disable the delete attachment button
ATTACHMENT_EDIT	Controls whether a user role may edit the attachment description of a previously uploaded attachment. Write permission is needed for this
ATTACHMENT_HIST	Controls the presence of a History button on the edit screen within the attachment. This button presents a window with a layout defined in the layout editor and named ATTACHMENT_HISTORY, which presents the historic records showing the updates and accesses to attachments. Normally, this layout is defined in the global area, and inherited by all business areas and projects.
ATTACHMENT_VIEW	Controls whether a user role may view a previously uploaded attachment. Read or write permission is needed for this
ATTACHMENT_SAVE	Ths button invokes the browser's ability to download and save an attachment, as opposed to viewing the attachment with the ATTACHMENT_VIEW button. Note that this button behaves

differently within different browsers:

Google Chrome	Downloads the attachment into the user's default download directory
Mozilla FireFox	Displays a Save As dialog box from which the user can select the destination into which the attachment may be saved
Microsoft Internet Explorer 11 running under Microsoft Windows 10	Displays a Save As dialog box from which the user can select the destination into which the attachment may be saved
Microsoft Internet Explorer 11 running under Microsoft Windows 7	This is not supported by the browser
Microsoft Edge	Downloads the attachment into the user's default download directory

ATTACHMENT_ID	This is an internal ID used by ExtraView. It is not likely that you will give permission to this key for any user role.
ATTACH_CONTENT_TYPE	This is the mime content type of the attachment. You can control the visibility of this key with read permission. You cannot write to this field.
ATTACH_CREATED_BY_USER	This is the name of the user who created the attachment. Once again, you cannot write to this field, and its presence or absence is controlled with the read flag.
ATTACH_DATE_CREATED	This is the date that the attachment was created. Once again, you cannot write to this field, and its presence or absence is controlled with the read flag.
ATTACH_FILE_DESC	This is the description of the attachment that the user enters. Both write and read operations are allowable on this field.
ATTACH_FILE_NAME	This is the name of the file that is attached to the issue. Once again, you cannot write to this field, and its presence or absence is controlled with the read flag.
ATTACH_FILE_SIZE	This is the size, in bytes, of the attachment. Once again, you cannot write to this field, and its presence or absence is controlled with the read flag.
ATTACH_PATH	This is the path in which the file was originally stored on the local file system of the creator's computer. Once again, you cannot write to this field, and its presence or absence is controlled with the read flag.
ATTACH_THUMBNAIL	This previews the attachments of image types (JPG, PNG and GIF) at a small size, in order that the user may see the image while they are inserting and updating issues, as well as seeing the field on reports when they are looking at attachment records.

Add & Edit Layouts

There is a default layout in the Global Area, Master Project that defines the layout for file attachments. This layout is named ATTACHMENT_SECTION. This may be embedded within any *add* or *edit* layout. Before any alterations, this looks like the following:
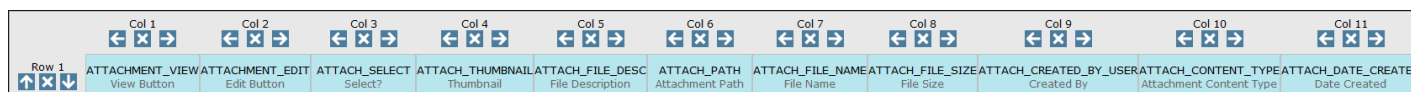
The security permission keys PR_ADD_PROBLEM.ATTACHMENT and PR_RESOLUTION.ATTACHMENT control the overall visibility of the layout, and the following security permissions control the ability to place attachments from within the *add* and *edit* screen layouts. Security permission entries exist for both the PR_ADD_PROBLEM set of keys and the PR_RESOLUTION set of keys, except for ATTACHMENT_DELETE, ATTACHMENT_EDIT, ATTACHMENT_HIST, and ATTACHMENT_VIEW which only exist for the PR_RESOLUTION set of keys.

Recommendation

If you are configuring mobile layouts, consider saving the ATTACHMENT_SECTION layout as a mobile layout and removing fields which are going to clutter up small screens, especially on phone targets. You can configure the fields on the layout vertically as opposed to horizontally, similar to this:
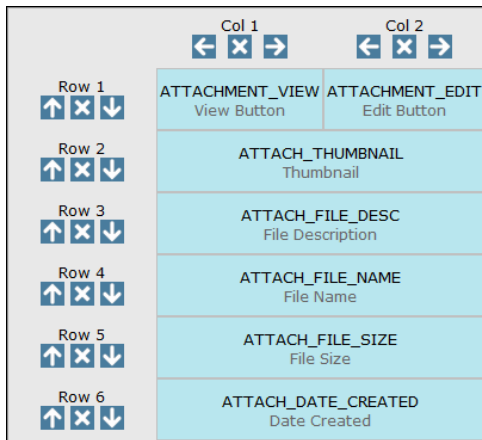
Multiple attachments can be added to issues, on both the *add* screen and the *edit* screen, if the user role has permission. However, attachments cannot be deleted from the add issue screen. If the user incorrectly adds an attachment during the add issue process, they should continue, add the issue to the database, edit the issue and then delete the attachment. Note that if an attachment is added to an issue when you are updating an issue, the issue itself is immediately updated.

Portion of edit screen showing attachments

Portion of detailed report showing attachments

A behavior setting named THUMBNAIL_MAX_SIZE controls the size of the thumbnail. This is the maximum number of pixels in either the horizontal or vertical direction to which thumbnail images will be generated. The aspect ratio of the original image will be retained. Thumbnail images are generated for file attachments and for fields with a display type of image. If you change this value, existing thumbnail images will remain at their original size and new thumbnails will be generated at the changed value.

Detailed Report Layouts

You should not embed the ATTACHMENT_SECTION layout within Detailed Report layouts. Instead, there is a field with special properties, named ATTACHMENT. This field should be placed on the Detailed Report layout. You then can control the presence or absence of the individual fields that comprise the attachment information with their security permission key. This leads to a layout for detailed reports that looks similar to this:

History Layouts

Before discussing history layouts, it is important to understand the behavior setting named HISTORY_DISPLAY. This offers the ability to configure one, two or three different methods to display the historic audit trail of each issue. A value of ABBREVIATED for this setting will show changed fields only in history records and will not use the history layout to display the audit trail. A value of SIDE_BY_SIDE will display the same information as ABBREVIATED, but with the addition that you have a side-by-side comparison of the current value at each point in time along with the previous value. The LAYOUT setting is designed to show all the field values as the issue moves through its process. The Report – History layout will show the fields that you want to see as part of your audit trail process. The following points are important with history layouts:

- The audit trail is kept on all fields, not just those on the layout. Thus you can add fields to the layout at any time, and the new fields will show the audit back to the creation of the issue.
- If you have a checkbox field on the *edit* layout, but not on the *add* layout, and do not make a change to its value on the first occasion you update the issue, the field will still show a change to **N** or whatever the off title is in the data dictionary. The reason for this is that ExtraView interprets the field as having moved from a null condition on the add screen to **N** on the *edit* screen. You can prevent this by making the default value for the field **N** in the data dictionary
- Before a field can be displayed on a history layout, it must have the **Select for Reports** attribute set to **Yes** in the data dictionary
- The field named RELEASE must have the **Select for Reports** attribute set to **Yes** in the data dictionary before you can display any repeating rows on the history layout
- You may not place any buttons on the history layout
- Use the behavior settings named HIGHLIGHT_COLOR, HIGHLIGHT_COLOR_ADD, HIGHLIGHT_COLOR_DELETE and HIGHLIGHT_COLOR_UPDATE to set the colors that indicate each of the conditions for a field within the history layout

Placing the Audit Trail for Related Issues on the History Layout

You may add an embedded layout prepared to display related issues to a History Layout. As explained in the section about preparing layouts for Related Issues, these layout types always begin with the character string RELATED_. This produces a related issue display at the point in time on the history record when the issue was updated, not at the current time.

Attachment History Layouts

The history button available on *edit* screens controls access to a pop up screen that shows the history of additions, updates and deletions to attachments. In addition, the screen shows a record of all users who have downloaded the attachment to their own computer.

The same ATTACHMENT_HISTORY LAYOUT is typically used throughout an installation, and can be defined in the global layout area only. Other Business Areas and Projects will automatically inherit it from there. Of course, different ATTACHMENT_HISTORY layouts may be defined in each Business Area and Project.

Attachment History Screen

The following fields may be placed on the layout, and they must have read permission to be viewable by the user role:

Field Name	Purpose
ATTACH_CHANGE_TYPE	The operation on the attachment. This is one of Delete, Insert, Update, View
ATTACH_CONTENT_TYPE	The content type of the attachment
ATTACH_CREATED_BY_USER	The name of the user who created the attachment
ATTACH_DATE_CREATED	The date the attachment was created
ATTACH_FILE_DESC	The description of the attachment
ATTACH_FILE_NAME	The filename of the attachment
ATTACH_FILE_SIZE	The size, in bytes, of the attachment
ATTACH_HIST_TIMESTAMP	The timestamp of the event, such as when a user views the attachment
ATTACH_LAST_DATE_UPDATED	The timestamp that the attachment metadata was altered, or the attachment deleted
ATTACH_LAST_UPDATED_BY_USER	The user who performed the operation
ATTACH_PATH	The original path on the client machine where the attachment was stored
ATTACH_THUMBNAIL	A thumbnail image if the field is an image

Notification Section Layouts

There is a default layout in the Global Area, Master Project that defines the layout for notifications. This may be embedded within any *add* or *edit* layout. Before any alterations, this looks like the following:

The security permission keys named PR_ADD_PROBLEM.NOTIFICATION and PR_RESOLUTION.NOTIFICATION control overall presence of the layout on the *add* and *edit* screens. The functionality of the fields within the layout are controlled by their individual security permission keys.

Field/Security permission key	Purpose
NOTIFICATION_GENERATE_EMAIL	This is an overall checkbox which acts as a switch, allowing the user to send or inhibit the sending of email notifications when the current issue is inserted or updated. The default for the checkbox may be set globally by the behavior setting named GENERATE_EMAIL_BOX or by a business load directive rule. Further business rules may alter the value of the checkbox during the refresh and preupdate stages
NOTIFICATION_CC_EMAIL_USERS	This provides an input text box with search capability, allowing the user to enter the email addresses or user ID's of users to notify, when these users are not on the notification mailing list generated
NOTIFICATION_INT_LIST_SELF	This checkbox allows the current user to place themselves on an interest list for the current issue, no matter who else may be notified during the processing of the issue. The user can later change this value through their personal options - on the Notification tab
NOTIFICATION_INT_LIST_OTHERS	This checkbox allows the current user to place others on an interest list for the current issue, no matter who else may be notified during the processing of the issue. These users can subsequently change this value through their personal options - on the Notification tab
NOTIFICATION_TO_CUSTOMER	This allows the current user to determine when to send a notification from the current issue to users who are members of the DEFAULT_USER_ROLE. This email is obviously subject to the permissions of this role, and will often use a simpler layout customized to the communication needs of your customers
NOTIFICATION_MAILING_LIST	This shows a list of the users who are on the standard distribution list for the current issue, as determined by the ORIGINATOR, ASSIGNED_TO, CONTACT and LAST_UPDATED_BY fields
NOTIFICATION_EMAIL_ATTACHMENT	This checkbox is used in conjunction wi the **Select ?** attachment checkboxes. When attachments are selected and this checkbox is checked, the selected attachments are sent with the email notification.

Buttons on the Menubar

A menubar is placed at the top and the bottom of each *add* and *edit* screen. All the available options have a button displayed here. If a button is not available, then it means that either the option is not valid at that point in time, or the user's role does not have permission to access the button. Write access to the security permission key is required to see the button on the menubar.

Button Title	Screens	Control	Purpose
Update	Edit	PR_RESOLUTION.EDIT_BUTTON	Without this button, the user's role may not update the issue
Submit	Add	MENU.ADD_PROBLEM	Without permission, the user's role cannot access the *add* screen and cannot create new issues
Update & Continue	Edit	PR_RESOLUTION.SAVE_AND_CONT_BUTTON	Allows the user's role to update an issue and continue editing the same issue, after drilling down from a report
Update & Prev	Edit	PR_RESOLUTION.ALLOW_EDIT_NEXT_PREVIOUS	Allows the user's role to update an issue and move to the previous issue, after drilling down from a report
Update & Next	Edit	PR_RESOLUTION.ALLOW_EDIT_NEXT_PREVIOUS	Allows the user's role to update an issue and move to the next issue, after drilling down from a report
Delete	Edit	PR_RESOLUTION.DELETE_BUTTON	Allows the user's role to delete the currently displayed issue. Confirmation is required
Clone	Edit	PR_RESOLUTION.CLONE	Allows the user to clone the current issue. Note that business rules may have been defined that control at a field level precisely which fields are cloned
Email	Edit	PR_RESOLUTION.EMAIL_BUTTON	Accesses the ad hoc email feature, to email a list of users about the current issue. Templates may have been defined to make this communication simpler
History	Edit	PR_RESOLUTION.HISTORY_BUTTON	Provides access for a user role to view the historic audit trail for the currently displayed issue
Close	Edit	-	Allows the user to discard any current changes and exit the window
Print Page	Add / Edit	-	Prints a copy of the current issue

Compatibility with Prior Versions

There are two items to be aware of when upgrading ExtraView from versions prior to 6.5.

Business rules that required an HTML modifier **onchange=submitChange()**	The HTML modifier was used to trigger a screen refresh to execute a business rule on the server. These are no longer required and you should remove these HTML modifiers from your layouts. If you do not remove the modifier, there will still be a screen refresh, but the logic will still be executed correctly
Working around a known bug in Internet Explorer	If you find that Internet Explorer renders your screens too wide, it is typically caused by an embedded layout where the fields are arranged in columns and the fields within one or more of the columns all use a colspan greater than one. The solution is to alter the offending layout (or layouts) to only use a colspan of one within the fields within these columns

Data Dictionary

The Data Dictionary is the central place where all field definitions are stored and maintained. The data dictionary is found within the System Configuration administrative menu.
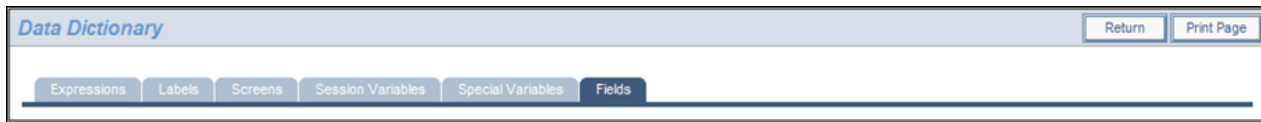
Data Dictionary screen tabs

Fields

These are the fields within ExtraView that are used within all processes. There are two types of fields, built-in fields and users defined fields (UDF's). Built-in fields are provided as part of all ExtraView installations. Although you may modify various attributes of built-in fields, such as their title and their default value, you should never delete an built-in field. You may create as many UDF's as you require for your installation. There is no limit on the number of UDF's that may be created. All fields have four basic properties, their scope, a fixed name, a title and a display type. It is the display type that gives a field most of its functionality. See later in this section for a complete explanation of all the field types. Once any UDF is defined in the data dictionary it will behave the same as if it were an inbuilt field. You will be able to attach permissions to the field, place it on any layout, report on the field, etc. End users will not be able to detect the difference between an inbuilt field and a UDF. When viewing a list of fields in the data dictionary, you will only see the fields in the Business Areas and Projects to which you have permission. This typically includes all the fields which are global in their scope.

As a technical point, there is no real limit to the number of UDF's you create. In addition, creating a UDF does not result in any change to the underlying database schema. There is no system degradation in creating a large number of UDF's.

Expressions

Expression fields are used in reporting to derive or calculate the value of a reporting column. Each expression may have a default expression which is then automatically used on any report which contains the expression field. Expression fields also have a range of display types, although this range is more limited than a field display type. You may define any number of additional expression fields, or reuse the same expression fields for different purposes on different reports.

Labels

This section is where field screen elements that are not database fields can be defined or altered. Once again, it is unlikely that the administrator will remove fields from this section. Labels can also be added as User Defined Fields, so it is unlikely that the administrator will add new fields to this section of the data dictionary.

Screens

These fields are for the inbuilt screens within ExtraView. A few additional supporting fields in this area affect screen displays. Administrators should not need to add or remove entries from this section, but can rename the individual screens within their installation.

Session Variables

Fields found in this area pertain to options having to do with the user's current, active ExtraView session. Fields defined in this section are principally used in reporting

Special Variables

These fields generally are those that manage functions involving calculations of some kind, such as the internal data/time stamping functionality.

Key Uses of the Data Dictionary

This core component of ExtraView controls many of the attributes of each field, such as:

- Display type of each field
- Display title for each field. The title that is used for all screen labels can be altered at will for each field
- Whether the field is selectable for reports
- Setting up field dependencies

- Populating the values into lists
- Default values
- Help text

Built-In & User Defined Fields

Built-In fields are part of the basic ExtraView product and exist in all databases. Most of these fields have special properties. For example there is a built-in field named **ID** that is used to provide a unique identifier for every issue stored. Built-in fields should not be deleted from the database else functionality may be lost, or in the worst case, ExtraView will not function. User Defined Fields are fields that are created within your ExtraView installation and do not exist in the basic product. Since you may want to customize your own site to include fields that are more specific and appropriate to your needs, UDF's satisfy this requirement. UDF's are a highly efficient and extensible mechanism. An unlimited number of UDF's may be created. All User Defined Fields are created and maintained in the Data Dictionary. UDF's must be typed, i.e. they will take on one of many display types such as *text, number* or *date*. The display type is the most importanat property of each UDF. In addition, UDF's that have a *list* type can retain multiple values at one time, with the values being individual members of the whole list.

Scope

Fields may be created and assigned to be global, or created and assigned to an individual Business Area, or created and assigned to an individual Project within a Business Area. This is termed the scope of the field. All inbuilt fields are global in their scope. When a field is restricted in its scope to an individual Business Area, or to a Project within a Business Area, then you will not see that field unless you have permission to read fields in that Business Area and Project. Similarly, you cannot write to a field unless you have write permission to the field in the Business Area and Project within which it is held.

Field Properties

Fields have properties, or attributes, that enhance their capabilities, and provide options as to how they should work. Not all properties are applicable to all field display types. The following is a complete list of properties. Only the properties for any field display type will be shown on the user interface. As you choose a field's display type, the appropriate properties will be displayed.

Field Property	Explanation
Business Area & Project	Sets the scope for the field. If Global Area is selected, then the field will be available over all Business Areas. If a specific Business Area is selected, along with its Project Defaults, then the field will be available throughout that Business Area. If a specific Business Area and a specific data project is selected, then the field will only be available within that Business Area and Project
Field Belongs To	Enter the record type that the UDF is to be associated with against the prompt *Field belongs to*. There are presently two choices: sea *Issue Records* – this indicates the UDF is to be associated with the main record for each issue in the database *Repeating Records* – this indicates the UDF is to be associated with each repeating row added to the main issue. Repeating rows maintain a many-to-one relationship with issues, therefore the UDF you create will be available for each repeating record that is added to the main issue. **Note**: You cannot change this property once the field has been created
Fixed Name	The *Name* is the fixed name of the field. This field is required. *Names* may be from one to thirty characters long, and must be composed of the characters – A – Z 0 – 9 _ The first character of a name must be alphabetic. You must not have consecutive underscore characters in a fixed name. Field names must be unique across an installation
Title to Display	The *Title* will appear on screens and reports. It may be from 1 to 255 characters in length and is required. It may contain any text. If you have turned on multiple locales, this field may be translated into any valid language and its length may be up to 3,800 characters. The title may contain HTML for formatting purposes but it is a best practice not to include HTML within the field title. You should provide a plain text title within the data dictionary, and use HTML to embellish the field title using an ALTERNATIVE TITLE wiithin the layouts where the field is used. If you do include HTML within the field title and access the field through the API or CLI then the HTML will be passed through to the output. If you do not want to display a title, use a space character as the title
Display Type	The data type of the field, controlling its appearance and behavior. Example display types are text, number, checkbox and date. For a complete list of display types, click here
Allow Selection on Reports	A switch to allow or disallow the field to be selectable for reports. If this option is not set to **Yes**, then the field will not be available in reporting and on the abbreviated history display. This is useful in order to prevent fields used as labels and for formatting appearing in field lists within reporting. It's important to note that not all fields should be

allowed on reports. For example, you may not use all of ExtraView's inbuilt fields within your installation and therefore want to hide them from the view of users (note you can also do this by changing the field's permission so it cannot be read). There are also fields in the Data Dictionary, such as images, buttons and screens that should not be selectable for reporting. It is highly recommended that you set all fields to **No** if they are not used in your installation. This avoids the lists of fields that your users can select from to be populated with unneeded entries

Remember Last Value	If you set this option, the last value that individual users enter or select on both the Add Issue and the Search screens will be remembered. This serves the purpose of remembering values that may not change frequently. For example, many users may work with the same *Product* for extended periods of time. Note that you can only set this feature for fields where the value of "field belong to" in the data dictionary is **Issue records**. The feature is not supported with **repeating row records**
Display as URL	This option need not be completed for any UDF. You can use either the name or value associated with the field as part of the URL that you generate. This allows you to link any field value on a form with any remote application (or an inbuilt ExtraView function that can be accessed with a URL) and optionally to use values on the form to pass as a parameter to another application. If a field with this attribute is rendered read-only on an *Add* or *Edit* screen or a browser-based report, then it becomes a link to the URL specified. To activate a field with a link to a URL, click the **Display as URL** radio button to Yes.

Parameter	Purpose
$$VALUE$$	Pass the current value of the field as a parameter in the request
$$DDNAME.VALUE$$	Pass the current value of the field named DDNAME as the parameter in the request
$$DDNAME.NAME$$	Pass the name of the item for the field named DDNAME as it is stored in the database as the parameter. For example, if you have a field named **STATUS**, with a name of **FIXED** and a value of **Fixed**, then $$STATUS.NAME$$ will pass **FIXED** as the parameter.
$$DDNAME.TITLE$$	Pass the title of the item for the field named DDNAME as it is stored in the database as the parameter. For example, if you have a field named **STATUS**, with a name of **FIXED** and a value of **Fixed**, then $$STATUS.TITLE$$ will pass **Fixed** as the parameter. This is equivalent to the $$DDNAME.VALUE$$ parameter.
$$URL$$	Pass the un-escaped value of the current field as a parameter in the request. The URL form is used to link the field to a URL that is fully formed, without escaping special characters. For example, with a data dictionary URL value of $$URL$$, a field containing http://www.yahoo.com would link directly to the Yahoo! site when the button for the Display as URL button is clicked
$$DDNAME.URL$$	Pass the un-escaped value of the field named DDNAME as the parameter in the request

    a. Type the appropriate URL in the field below the **Display as URL** radio button.
    b. You can pass values from the data dictionary field you are defining or from other fields on the form. The form of the values that you can pass as parameters are as follows –
    c. Note that the trailing $$ on an entry is optional
    d. Use the tag $$APP_HOME$$ to include the current path to the ExtraView instance that you are running
    e. If the DDNAME is the name of a data dictionary Special Variable (e.g. SYSDATE) then the value passed is the value of the Special Variable
    f. If the DDNAME is the name of a data dictionary Session Variable (e.g. USER) then the value passed is the value of the Special Variable
    g. You may call other functions within ExtraView and cause them to take the appropriate action. For example, you can set up a field with a link that utilizes the search function and display the results in a Quicklist report

    Example 1 – Pass a value to a remote application as a parameter –

    http://search.yahoo.com/search?p=$$VALUE$$

    This will pass the current value of the field to Yahoo, and perform a search of the value. The results will be displayed in a new window.

    Example 2 – Display a window with a user's details –

    ?p_action=showUserDetails&p_option=admin.UserAccountsDisplay&p_user_id=$$NAME$$

    This will access ExtraView's pop-up display for a user's details. Use this URL in conjunction with a User display type field.

Example 3 – perform a search with a keyword, and display a Quicklist with the results –

?p_action=doRunQuicklist&p_option=search.SearchDisplay&searchword=$$VALUE$$
&product_name=$$PRODUCT_NAME$$&assigned_to=$$ASSIGNED_TO$$

This example is placed in the URL value of the field (all on one line). This will access ExtraView's search class and produce a Quicklist report, using the value in the display field named searchword, and the current values of the product_name and assigned_to fields

Example 4 – Edit an issue whose ID is in the field with the display as URL –

?p_action=doEditDisplay&p_option=Display&p_id=$$VALUE$$&p_from_action=search
&p_from_option=search&p_close_win=true&ev_menu=off

This will open an edit screen window using the value of the field with the Display as URL as the ID to the issue. Note that the parameter p_close_win=true is used to close the edit screen window once the issue has been updated. Also note that the parameter ev_menu=off is used to suppress the ExtraView navigation bar in the new edit screen window.

h. The URL form is used to link the field to a URL that is fully formed, without escaping special characters. For example, with a data dictionary URL value of $$URL$$, a field containing http://www.yahoo.com would link directly to the Yahoo! site when the button for the Display as URL button is clicked.

Image for Display as URL	This optional entry is used in conjunction with the **Display as URL** setting. If provided, the entry is the file name of an image in the directory specified in the IMG_NAV_BAR_HOME behavior setting. When this image is provided and **Display as URL** is set to Yes, then your image is used rather than the inbuilt LinkButton.gif image. Care should be taken not to provide an image that is too large. The recommended size is 18 x 18 pixels.
URL	An optional URL which can be used to link the field to any other application
Default Value	A field must be created before you may give it a default value. This is because default values are used most commonly in list fields, and you need to populate your field with these values before you can select one to be the default. If you provide a default value for a field, the value specified will be auto-selected whenever you are adding a new issue. Note that if you also have remember last value set on the field, the last value will be used in preference to the default value. The default value is used to populate a field in a record, when the field is visible on the *add* layout for an issue. You may also use a **load** business rule to set a default value on a field. **Note**: Default values for fields of display types Checkbox, List, Pop-up, Radio Button – Horizontal, Radio Button – Vertical, Tab and User must be entered using the **Select Default** button. Entering text into the box for these fields does not work. The reason for this is that in addition to the value that is displayed, ExtraView also needs to know the internal ID value of these fields. You may enter text directly for other display types, such as Text. Default values may only be entered after the field has been created, by editing the field.

Entering Default values

Help Text	This field provides a facility that when you mouse over the label adjacent to a field you can provide a tool tip message that will appear. You can have up to 2,000 bytes in a tool tip, without localization being turned on in your ExtraView instance. If localization is turned on, then the tooltips can store up to 3,800 bytes. However it is not recommended that you have large tooltips, because of the way in which Windows and other operating systems display these. Typically they are displayed for around 2 seconds, so you should not store more information that can be read in that time
Help URL	With a Help URL, you can link this to a bookmark or page in your own online help system. If you are an ExtraView customer hosted by ExtraView Corporation, note that this URL need not be on our server. You can store and access these files anywhere over the Internet
Currency unit	This block of information only appears if you have chosen a field display type of **Currency**. Choose the currency symbol to display for the field. The list of currency units is complete, and you may select one for each field. You cannot use a field with a currency display type if you want it to signify different currencies. You can choose how to display negative currency values from one of the negative display options shown below

Negative Display	This block of information only appears if you have chosen a field display type of **Decimal**, **Currency** or **Number**. Simply choose the formatting you require for the field using the radio buttons
Symbol	This block of information only appears if you have chosen a field display type of **Currency**. This determines whether the currency symbol will appear along with the number when the field is displayed
Thousands Separator	This block of information only appears if you have chosen a field display type of **Decimal**, **Currency** or **Number**. Select whether you do or do not want a thousands separator to appear with the field
Rounding mode	This block of information only appears if you have chosen a field display type of **Decimal**, **Currency** or **Number**. The default rounding mode is ROUND_HALF_UP for **Decimal** and **Number** display types. For **Currency** display types, the default rounding mode is the rounding mode of the currency instance for the chosen currency locale. This table shows the result of rounding data input to one digit, with the given rounding mode:

Input Number	UP	DOWN	CEILING	FLOOR	HALF UP	HALF EVEN	HALF DOWN	UNNECESSARY
5.5	6	5	6	5	6	5	6	throw Exception
2.5	3	2	3	2	3	2	2	throw Exception
1.6	2	1	2	1	2	2	2	throw Exception
1.1	2	1	2	1	1	1	1	throw Exception
1.0	1	1	1	1	1	1	1	1
-1.0	-1	-1	-1	-1	-1	-1	-1	-1
-1.1	-2	-1	-1	-2	-1	-1	-1	throw Exception
-1.6	-2	-1	-1	-2	-2	-2	-2	throw Exception
-2.5	-3	-2	-2	-3	-3	-2	-2	throw Exception
-5.5	-6	-5	-5	-6	-6	-5	-6	throw Exception

Internal Precision	This block of information only appears if you have chosen a field display type of **Decimal**, **Currency** or **Number**. This is a number from -24 to 10 indicating how many digits of precision should be maintained for internal computations with this field
Report Precision	This block of information only appears if you have chosen a field display type of **Decimal**, **Currency** or **Number**. This is a number from -24 to 10 indicating how many digits of precision should be displayed on reports or read-only displays of this field
Percentage	This block of information only appears if you have chosen a field display type of **Decimal**. If you set the *Percentage* option, then users must enter the number with a percentage sign. This will be converted internally by ExtraView and stored as a decimal value
Alias Of	A field that is the alias of another field is restricted to fields that you create with a display type of List, Pop-up or Tab. When you create a field with this setting, you will not maintain the list of values for the field, but the field will be created using the list of values from the field you select in the **Alias of** selection list. This is a labor saving device that is used when you require more than one field with the same list of values, but you only want to maintain the list in a single place. For example, you may want to create a list of product releases, and use this list in two forms, one for Release Found and one for Release Fixed. You can place both these fields on a single edit screen, and update the fields independently, but only maintain the field in a single place. A field that is an **Alias of** another field may still be controlled by its own allowed value list. This allows you to create different lists from one base list, where each of the aliased lists is a subset of a master list. This entry only appears when you are creating a new field, or when you are editing a field with a display type of List, Pop-up or Tab. The option is only valid when creating a field of the same types. When creating a new field, you may point it to another list field and the new field will be an alias of the field to which it point. It will be kept in synchronization with the first field. If you are editing an existing entry which is an alias of another field, you can remove the alias entry, from which time the fields will work independently
Total Field on Reports	This only has an effect on numeric display types. When this is selected, the field will be totaled automatically on column reports. This setting is ignored if the field does not have a display type of *Number*.
Enable Interest List	This flag in a data dictionary item enables a notification interest list for specific values of the field. For example, you may want to create an interest list based upon issues that have a critical severity level. If you enable the interest list for *severity level*, then the administrator can edit the *critical* list item in the severity level interest list and maintain the list of user names in the interest list
Auto-complete	This option may be used with popup list fields and with user fields when the behavior setting named USER_LIST_DISPLAY is set to a value of POPUP. Auto-complete, or type-ahead as it is sometimes known, works

by having one or more characters typed into the list and then automatically presenting you with the most likely matches for the value you are composing. At any time you can select a value from those presented to you by clicking on the value, or you may continue typing in characters until you complete the value. As you type, the values presented in the list are refined to further match your entry.

Example of an Auto-complete field

When you set auto-complete to **on** for a field, then you also select the number of characters that the user enters before the auto-complete is activated. There is also control over the maximum number of entries returned in the list and control over the height of the list. This allows fine-tuning when you are dealing with very large lists, ensuring that the user does not retrieve a very large number of records while they are typing characters into the list

Multiple Value	Multiple values can only be enabled for UDF's with a display type of LIST, POPUP or User. If you enable a UDF to have multiple values, you are allowing the user to select any number of items from its list as being valid entries. For example, say you have a list of operating systems against which an issue may be recorded. If you want to note that the issue is found in some of the operating systems, but not all, you can choose the relevant entries from the list. A behavior setting named SORT_SELECTED_VALUES controls how the selected values in multi-valued lists and multi-valued user lists are sorted. When this is set to NO, all multi-valued lists will be displayed in the order set by the field's standard sort sequence. When this value is set to YES, all selected values will be displayed at the beginning of the list, using their sort order, then the non-selected values will be displayed using their sort order. Note that if you place a multple value list UDF on an *add* or *edit* screen layout, and then add a layout cell attribute of FIELD REMOVE NONE, and you then deselect the very last value in a list, the first value in the list will be selected when you submit the form. If this behavior is not desirable, then the recommendation is to also make the field required, which will force the user to select a value before the record is inserted or updated. You can set the character used to signify that a value in a multi-valued field is selected with the behavior setting named MULTI_VALUE_HIGHLIGHT_CHAR. The default value is &# 9654;. This displays as a ? against the selected value. If you want to use a character that is in the basic character set of older computer browsers, it is suggested you use the + character.
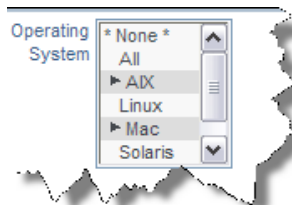
Example of a multi-value list field

Filter Criteria	This option allows the field to be selected on a query filter layout, as a criteria for a search. Only if this item is checked can the field allowed to be placed on a filter screen layout for querying and searching. Note that not all display types can be used as filter criteria. Fields with a display type of Label, Image, Text Area, Log Area and Print Text cannot be used as filter criteria
Is Sortable	If this option is set to Yes, then the field will appear on the Sort Order lists on report composition screens. Typically fields with a display types as follows should be flagged as sortable or not:

Sortable	Not Sortable
Checkbox	Button
Date	Custom
Day	HTML Area
List	Label
Number	Log Area

Pop-up	Print Text
Tab	Text Field
User	Text Area

Used for user attributes
This is only an option when the user's role has permission to SE_USER_ATTRIBUTES and the field being modified is a UDF, with a type of List, Popup or Tab. When set to *Yes*, the field is placed on the User Account Administration screen, under the tab which has the default title of **Attributes**. Custom code must support this operation, typically defining what happens to issues when the user does or does not have specific values set in the lists. Once this option has been set on any field, and the administrator has set a value into the field within the user accounts maintenance screens, then you cannot remove this attribute from field, without first removing all the values in all the users who have this field set

Sort sequence for user attributes
This field only appears when **Used for user attributes** is set to *Yes*. It provides the ability to sort the different fields that are used as user attributes on the User Account Administration screen. The default if this is not set is to sort the fields alphabetically

Encrypt this field
This field only appears for fields with a display type of **Text Field**, and only when adding a new field. This enables encryption for the field, at the database level. The key purpose is to provide an additional level of protection against unauthorized access direct to your database. The feature works in conjunction with a secret key obtained from a keystore. They secret key(s) must be defined before you create fields that utilize encryption. View the page on Encryption Key Management for more information.

Note: You cannot change this property once the field has been created

Reserved Words

Also, note that there are reserved words that cannot be used as UDF names. These are:

ACTION	*NEW_REPORT*	*REPORT_OWNER*	RG_RELATION_FIELD
CALLED_FROM	*NOTIFY*	*REPORT_START*	*SEARCH_ATTACH_SIZE*
CHILD	*OPTION*	*REPORT_STOP*	*SECURITY*
CLASS	*PAGE_LENGTH*	*REPORT_TITLE*	*SELECTED*
CUSTOM_URL	*PAGE_SIZE*	*RG_NAME*	*SELECTEDSO*
CUSTOMER	*PARENT*	*RG_NAME_FIELD*	*SHOW_EXPANDED*
DATE	*RECORD_COUNT*	*RG_REF_ID*	*SOURCE*
FROM_ACTION	*RECORD_START*	RG_REFERENCE_ID_FIELD	*SOURCESO*
FROM_OPTION	*REPORT_AS_OF*	RG_REFERENCE_NAME	*TEMPLATE_EXPANDED*
GROUP	*REPORT_DESC*	RG_RELATE_THIS_TO	*UDF*
INTERFACE	*REPORT_ID*	RG_RELATION	*VALUE*
LANGUAGE			
LAYOUT_SESSION_TAG			

Display Types

Each field in ExtraView, whether it is an inbuilt field, or a user-defined field, has a display type that controls the presentation and behavior of the field, both in input mode on Add and Edit screens, as well as on output for reports. The range of Display types that can be set is as follows:

Button

There are inbuilt fields with a display type of **Button** such as for *View*, *Edit*, *History*, *Delete* and *Select* fields. These are used on reports to provide access to the appropriate functions. These fields appear within the data dictionary under the **Labels** tab. There are a few other inbuilt **Button** fields for internal ExtraView functions.

When a user clicks on a field with a display type of button, it is expected that some action will take place. These actions need to be defined with custom code within the ExtraView environment. To create a custom button, you will create a field within the data dictionary with a display type of **Custom**. Custom code must be defined to perform the action. Functions are available within the custom code environment to detect when a custom button is clicked and from there your code must handle the action.

Checkbox

Check if over limit ☑
Check if over age 25 ☐

This generates a single checkbox on a screen. Apart from being able to set the title of the field, the administrator can also select the values that appear on reports according to whether the box is checked or unchecked. The default is for **Y** to appear if the box is checked, and **N** to appear if it is not checked. These values can be altered to any other titles such as **Yes** and **No**, **On** and **Off** or similar. Note that if a checkbox is never set to being checked or unchecked by a user, or is never specifically set by a default value or a rule, then the checkbox actually has a **null** value, and is never stored in the database. This is an important distinction, as a **null** value is not the same as a **N** value. Querying for a **null** value with * **None** *, will give different results to querying for **N** values.

Currency

This field allows the user to enter numbers that will be displayed with the formatting seen on currency amounts, such as a currency symbol, thousands separators and decimal points. The user may choose from a list of valid currency symbols, as well as set the formatting for negative numbers and set the thousands separator. Note that the formatting information for currency fields is entered into a box that appears in the data dictionary when you select the display type of **Currency** from the selection list. The implementation uses a precision of 38 and scale of 10 (28 digits to the left of the decimal point and 10 digits to the right) when the value is stored in the database. If a user attempts to insert a number larger than this into a currency field, it is not possible to accurately display the number. At this point, ExtraView converts the number to a null value, and enters a message into the application log file.

Custom

This field assumes that ExtraView's user custom code will provide the contents and management of the field, allowing you to extend the functionality of ExtraView with new custom display objects. The field title will be rendered as normal, using the title of the object. The method call to user custom that provides the code is:

ucValue = Z.userCustom.ucRenderEmbeddedObject(dbconn, session, layoutType, dde, ddName, selectedVals, selectedVal, multipleVals, attributes, prefix, le, row, styleVal, doHiddenInput);

The returned value ucValue is a String. For fuller details on this display type, please see the User Custom Guide. However, it is worth noting here that almost any code can be generated into your layouts with this display type.

Date

?	January, 2007	×
«	‹	Today	›	»
Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			
Time:	16 ⬍	:	11 ⬍			
Select date						
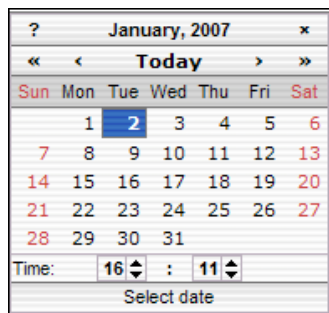
This field type allows dates and their time component to be entered and stored. The date field is automatically provided with a popup button on each add or edit screen. This button provides access to a calendar. Dates are validated against the user's locale. For example, if your locale is US English, the date 1/5/05 is interpreted as January 5th, 2005, while if your locale is GB English, the date 1/5/05 is interpreted as May 1st, 2005. All dates are stored with a time component. When a user's time zone is different from the time zone stored in the behavior setting named DB_TIMEZONE, the date / time a user enters is converted to its equivalent in the DB_TIMEZONE, and stored as such. For storage, all dates are further converted to UTC format. When the same user accesses and displays the date again, it is converted to his personal time zone and redisplayed. If a user is in a time zone different to the user who entered the date / time, and is in a time zone different to the DB_TIMEZONE, the date and time are converted to their personal time zone and displayed accordingly. When any user displays a Date field, the date is displayed with the user's own personal date format. When entering a Date field, the user's locale is used to assist in validating the entry. If the user's date format begins with the day, it is assumed that dates entered will also start with the day. If the user's date format begins with the month, it is assumed that dates entered will also start with the month. This avoids ambiguity with dates such as 3/6/2012 which can be interpreted as March 6th or the 3rd June, depending on where you are located. If your application does not require time with any date you want to store, it is recommended you use the 'Day' display type to define the field.

Note: Microsoft SQL Server has a limitation in that any date before 1753 is invalid and cannot be stored, and calculations based on dates before this are invalid. If your installation is running on SQL Server, this is a limitation that ExtraView cannot work around.

Users who have write permission to date fields on *add* and *edit* screens and whose personal date format is set to exclude time, for example using SHORTDATE, can still enter time into a date field. The popup calendar will supply the time automatically, but the user can override this. This avoids problems between users when some are using a personal date format that contains time, while others use a personal date format that excludes time.

Day
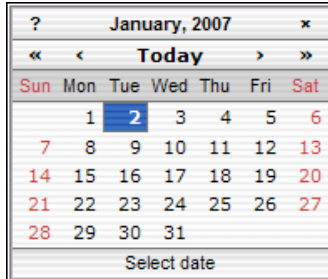
Day field types operate differently than fields with a display type of Date. Firstly, they do not have a time component, and only the date is ever displayed on a screen or report. Secondly, when a field value is input, displayed, or filtered, it is treated the same for all users in all time zones. Day field types still respect the locale and date format of the user, although no time fields will ever be shown for a Day value on reports, admin screens, or add/edit screens. If you require time as part of the field, use the 'Date' display type field. When entering a Day field, the user's locale is used to assist in validating the entry. If the user's date format begins with the day, it is assumed that dates entered will also start with the day. If the user's date format begins with the month, it is assumed that dates entered will also start with the month. This avoids ambiguity with dates such as 3/6/2012 which can be interpreted as March 6th or the 3rd June, depending on where you are located. Using a Day type field within a Business Rule is consistent when it is compared or combined in expressions with other Day type fields. When compared with Date fields, the Date is converted to a Day value using the user's time zone (to which the Date value is, by definition, relative) and the comparison is done with a Day field to Day field comparison or calculation. The date will be presented to each user with the date format they have selected.

Note: Microsoft SQL Server has a limitation in that any date before 1753 is invalid and cannot be stored, and calculations based on dates before this are invalid. If your installation is running on SQL Server, this is a limitation that ExtraView cannot work around.

Decimal

This allows a number with a number of decimal points to be entered and stored. The implementation uses a precision of 38 and scale of 10 (28 digits to the left of the decimal point and 10 digits to the right) when the value is stored in the database. If a user attempts to insert a number larger than this into a decimal field, it is not possible to accurately display the number. At this point, ExtraView converts the number to a null value, and enters a message into the application log file.

Document
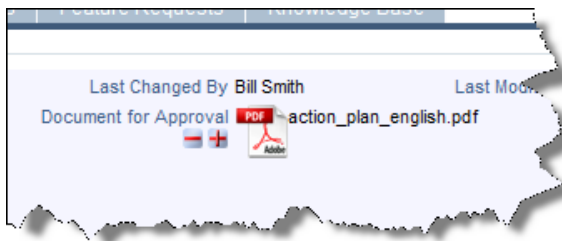
Document fields store a document of any type. For example, you might have a document that is required as part of the submission of an issue, or you may require an approver of a document to provide a signature indicating their approval of a document. Documents are represented on the screen with an icon that shows its type. For example, if the document is a spreadseet, you will see a spreadsheet icon beside its filename. Unlike attachments, documents are subject to workflow, such as being required if a condition is true, or a signature might be required indicating approval of a document that was uploaded. Document fields are only searchable with keyword searching, and only using Quickfind searching. Document fields are not searchable by default, and searching must be activated through the behavior setting named FULL_TEXT_SEARCH_COLUMNS or by updating the searchable columns on the Quickfind settings screen. When uploading a document field, the permission keys named PR_ADD_PROBLEM.ATTACH_DESC and PR_RESOLUTION.ATTACH_DESC are used to control the requiredness of the description to the document field.

HTML Area

This display type offers the ability to enter and edit text with an HTML editor. This field will appear on the *add* and *edit* screens with a toolbar, much as you have with a word processor. On reports, the HTML within the fields is rendered, including embedded objects such as images. You can control which buttons appear on the toolbar, and the style of the toolbar with the behavior settings named EDITOR_BUTTONS and EDITOR_STYLE respectively. The row height of HTML areas throughout ExtraView is controlled with a behavior setting named HTMLAREA_ROW_HEIGHT. Each HTML Area field may store up to 5MB of data.

Screen Example

This is a **field** with rich text. |

Company Name	
User Name	
Address	
Country	

div p
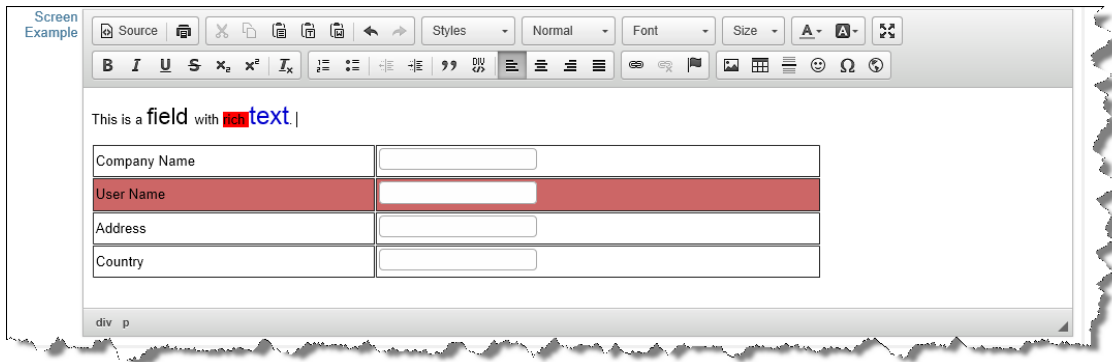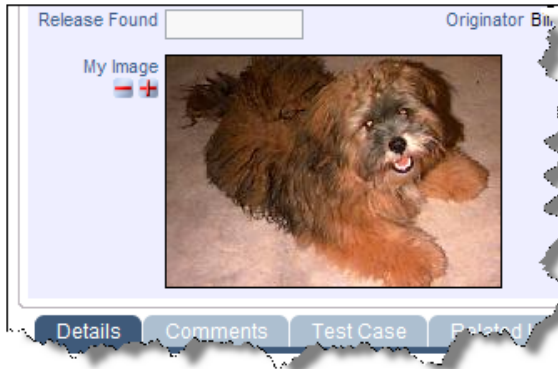
Image

This display type allows images to be stored within issues in the database. These images appear as fields within layouts, and can be added, deleted and updated within issues like all alphanumeric field display types. Image fields may be placed on report layouts and column reports. When an image is uploaded into an issue, a thumbnail is generated and used on the *add*, the *edit* and report layouts. The size of this thumbnail is controlled by the behavior setting named THUMBNAIL_MAX_SIZE. The aspect ratio of the thumbnail will always be preserved, but the maximum number of pixels in the horizontal or vertical direction will not exceed the setting. When the user clicks on the thumbnail image, a window opens with the image being displayed in its full size. A behavior setting named MAX_IMAGE_DIMENSION_PIXELS controls the maximum dimension of any image thay may be uploaded. Images that have a height or width greater than this value will not be stored in the field. When uploading an image field, the permission keys named PR_ADD_PROBLEM.ATTACH_DESC and PR_RESOLUTION.ATTACH_DESC are used to control the requiredness of the description to the image field.

Label

A label is used as a display item only. No data is stored in the database against any label field. Labels are given a value in the data dictionary default value field, and this is displayed on the screen in read only mode. Some inbuilt labels are also used internally within ExtraView, to provide a means of altering a term such as "problem" to be suitable for the use of ExtraView in any customer installation. Labels may be placed on layouts such as the *add* and *edit* layouts. They can be used to provide instructions to users or other information.

List
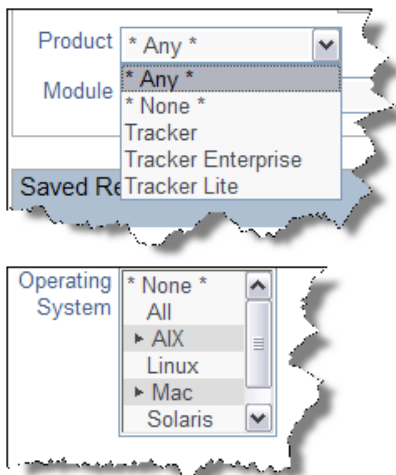
A list of values; when you have a field with a list display type, it will have an entry on the Field List Maintenance section of the administration screen. From there, you can add or delete list items. Lists are often used with allowed values, controlling the absence or presence of any individual item when in data entry mode. If you require two or more fields that are populated with the same list of values, you can create the second and subsequent fields using the "Alias Of" feature. You will maintain the list of values in the first field created, and ExtraView will automatically keep the aliased lists up to date. This is a labor saving device, and ensures that you can keep identical lists in step with each other. At the same time, you may define separate allowed value entries for each of the fields. You may also use the layout cell attribute **Add the * New * entry to list fields** to allow the user to create new values for a list, without requiring the user to have administrative privileges. For scalability, fields with the **List** display type may be changed to the display type of **Pop-up**. Note that when you make this change, any business rules in force are not refreshed immediately, so any rules using the field you altered may not work until the rules refresh themselves (an automatic process done every few hours). If you want the rules to work immediately, just go to the rules screen, and press the **Update** button. An attribute of list fields is that they may be set to accept a single entry, or mutiple values within the list may be selected. Multi-valued list fields have an indicator to the left of the field value when

that value is selected. In addition, the value is highlighted. Reports will
display all the selected values when the field is shown.

Log Area   This field functions as a log of successive text entries to a field. Without access to the security permission key named PR_RESOLUTION.EDIT_LOGAREA_FIELDS, the user cannot edit or delete previous entries but can only add additional entries. The user's name and a time stamp are shown against each entry. When the field can be updated, two buttons, to shrink and enlarge the text area appear adjacent to the field. There is a behavior setting named LOG_AREA_DISPLAY_CHARS in the Display Settings area of Administration. This can be used to truncate the size of the old entries displayed on the edit screen. When the field is truncated, "more…" is displayed at the end, allowing the user to refresh the screen with the entire entry. Each entry into a Log Area field may store up to 5MB of data.

When there is more than one entry within a log area fields, sort icons will appear by the user's name and timestamp of the first entry. The user can use these to resort the comments in different directions. Use the LOG_AREA_INITIAL_SORT behavior setting to determine the direction of the initial presentation.

Log area fields may be entered on forms and presented on existing entries and reports as rich text, when the prompt **Present as HTML** is set to **Yes** in the data dictionary definition for the field. This option is not available on log area fields which are defined for repeating rows.

It is not recommended that you place log area fields on repeating record layouts. Each entry in a log area field is preceded by a header that shows a combination of the user's name, the user's company name, the timestamp of the comment and a sequence number for the comment. See the text for the behavior setting named LOG_AREA_TEMPLATE to adjust the presentation of log area fields.

Note: If a user does not have read permission to the security permission key named PR_RESOLUTION.ASSIGNED_TO, then they will not see the name of the person who created the entry to the log area. This is useful if you want to allow customers or guests to see the comments made, but to not know who made the entry.

You can make the size of the log area field auto-size to the amount of text in the field, by adding a layout cell attribute of *onkeyup=autoSize(this);*. This attribute will only apply to the layout within which it is set. With this attribute, the number of rows that the field occupies will increase automatically as the user enters additional text.

Number   This is a field that accepts and stores only numbers. Numbers may have decimal points, but they must not have other formatting. The internal data type used for the NUMBER data type is double, allowing precision of approximately 16 decimal digits (0 to 9007199254740992). You cannot enter numbers larger than this, without losing precision and accuracy.

Pop-up   Similar to the display type of List, except a separate popup window opens with list of possible values. The user may either type an entry directly into the field, or select a value from the list in the popup window. This field is useful if the list is long (say more than 100 entries), as it is easier to search a pop-up box. If there are more entries in the list than the number defined by the behavior setting named POPUP_LIST_SIZE, the popup list is embellished with a search box, and a list of the first characters of the data entry values (usually A – Z), to make it easy to search the list. Similar to fields with a display type of List, you can create Pop-up list fields that are aliased to a master list, reducing the amount of maintenance needed. Pop-up fields have an attribute that allows type-ahead, or auto-complete. You can set from 1 to 5 characters that are entered before triggering the presentation of the closest matches to the characters you have typed into the field.

Print Text   This is similar to the Text Area type field, but on display, a fixed-width font is used for the field. This can be used if the field may routinely be used to hold diagrams drawn with characters such as +------+ on the keyboard and you want to preserve the spatial accuracy of the diagram. Print text fields can store up to 128k of text at one time and while they are in edit mode, two buttons appear adjacent to the field, allowing you to expand or shrink the area for

text entry. Text output fields to reports with a type of Print Text will preserve exactly the spacing that the field was created with, including spaces and returns in the data. This preservation of the input may cause reports such as the detailed report to display wider than normal. You can make the size of the log area field auto-size to the amount of text in the field, by adding a layout cell attribute of *onkeyup=autoSize(this);*. This attribute will only apply to the layout within which it is set. With this attribute, the number of rows that the field occupies will increase automatically as the user enters additional text. Each Print Text field may store up to 5MB of data.

Radio Buttons
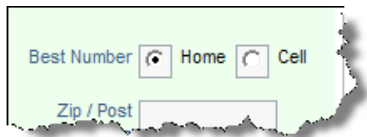
Horizontal Radio Button
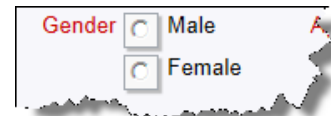
Vertical Radio Button

Radio buttons will use a list of values to create a set of buttons on the screen. The user can select only one value from the list when they are filling in an *add* or *edit* screen. You must create at least two values for the field to function properly. It is usual to select one value as a default value. This will be the selected value when the form is first displayed. Horizontal radio buttons are rendered across the screen on both *add* and *edit* forms. Vertical radio buttons are rendered down the screen on both *add* and *edit* forms. On query screens, the radio button will act like a single-valued list field, and the results on reports will look just like results from a single-valued list field.

Like all other field types that render a list-like object, the default rendering of a radio button will include a value option of *** None *** in the list. However, a global attribute is automatically added to a newly created radio button, to **Remove the * None *** Entry from the field. You may delete this attribute if you do want a *** None *** entry within the field when it's displayed.

Within a Quickedit session, radio buttons are rendered as list fields, as the amount of space occupied by a radio button with many values may cause navigation problems within the Quickedit session. Also, within a Quickedit session, a value of *** None *** is added to the list, even if there is a layout cell attribute with a value of **Remove None**. This ensures that the first value in the list is not selected automatically when it has no value. Radio buttons are treated as list fields internally within ExtraView, but there are very few occasions when you want the default *** None *** value to be displayed.

Radio buttons have two ways in which they may rendered on the screen. Further, it is possible to set a "toggle" option, allowing the user to deselect a radio button, as opposed to only allowing them to select values.
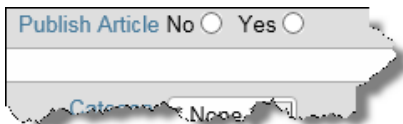
Standard style radio button

Box style radio button

Note that when you display radio buttons on the ExtraView Mobile app, radio buttons are always displayed with the box style. Standard radio buttons are most often too small to select with a finger.

Tab

This field behaves like fields with a display type of List, except that the list of values that is displayed as a set of tabs across the screen. Typically, this is used to provide a high-level selection on a screen, where the subsequent fields depend on the tab selected. You should not use this display type if the supporting list has more entries than fit across the screen.
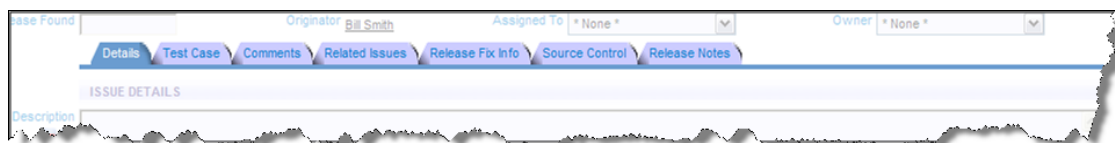
Example of a tab field

When you create a new Tab field in the data dictionary, it is strongly recommended that you add a global attribute of REMOVE NONE, so that the *** None *** value does not appear on the row of tabs that are generated. If you want the Tab field to default to a specific value, you should set a default value for that tab.

Text Area

This is multi-line text field. Text area fields can store up to 128k of text. When the field is in edit mode, two buttons appear adjacent to the field, allowing you to expand or shrink the area for text entry. Note that text area fields may not contain tab characters. You can make the size of the log area field auto-size to the amount of text in the field, by adding a layout cell attribute of onkeyup=autoSize(this);. This attribute will only apply to the layout within which it is set. With this attribute, the number of rows that the field occupies will increase automatically as the user enters additional text. A behavior setting named TEXTAREA_ROW_HEIGHT allows the administrator to set the default height of all text area fields. Each Text Area field may store up to 5MB of data.

Text Field

This is a single-line text field. Up to 255 characters may be stored in a Text Field. Note that you can control the width of the text box into which data is typed within each layout on which the Text field is used, using the Layout Cell Attribute named Size.

Encryption

Text fields may be encrypted. To achieve this, you must set up a keystore with the encryption key(s) and decide which fields are to be encrypted within the data dictionary, on a field-by-field bsais.

Telephone Number Formatting of a Text Field

There is a built-in function that can be applied to a text field to apply standard formatting for US-style telephone numbers. To enable this feature on a text field, you create either a data dictionary global attribute or a layout cell attribute with the following HTML modifier: onkeyup=formatAsPhoneNum2(this) In addition, set a layout cell attribute of SIZE = 15 to ensure the phone number is of the correct number of characters. If you create the attribute as a global attribute within the data dictionary, there is no need to add the layout cell attribute to each layout that uses the function. There is a deprecated telephone format using an HTML modifier function of formatAsPhoneNum(). This should no longer be used as it does not work within all browsers. If you require the formatting of a telephone number in a different way, or if you require the formatting of any text field in an efficient way, it is recommended that you write a user JavaScript function and call this from an HTML modifier on the field.

Time Interval

Time interval fields are inbuilt fields only and are used to display the formatted results of measuring time in a Service Level Agreement (SLA). These fields are automatically generated and placed in the data dictionary when an SLA is created. The administrator may choose the format in which to display the time measurement on an SLA report. All the normal attributes and permissions for fields apply to time interval fields.

User

The inbuilt fields *assigned_to, originator, owner* and *contact* are User fields. User Lists displaying these fields will display the *first_name* and *last_name* or the user ID, depending on what is set in the USERNAME_DISPLAY property as the behavior setting. You can define a User Defined Field with the display type User. This will generate a list populated with a list of active users in the system. If the behavior setting named USER_LIST_DISPLAY has a setting of POPUP, then user fields have an attribute that allows type-ahead, or auto-complete. You can set from 1 to 5 characters that are entered before triggering the presentation of the closest matches to the characters you have typed into the field.

Note: If your installation is configured to use an LDAP connection with the behavior setting named LDAP_USER_LOOKUP having a value of YES, when auto-complete is configured, only users that are currently stored within ExtraView are returned with auto-complete, as opposed to all potential users in the LDAP server. This prevents a potential performance issue, returning a huge number of records from the LDAP server to ExtraView, whenever a user presses a single key.
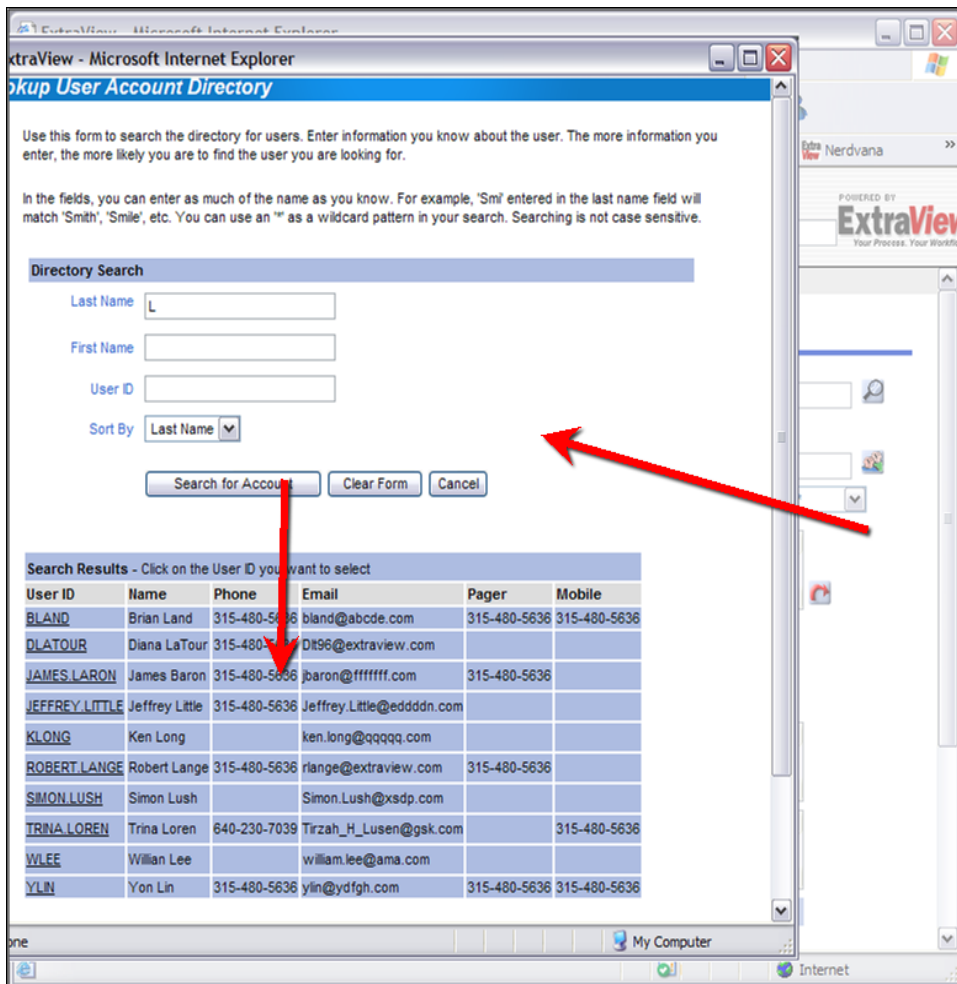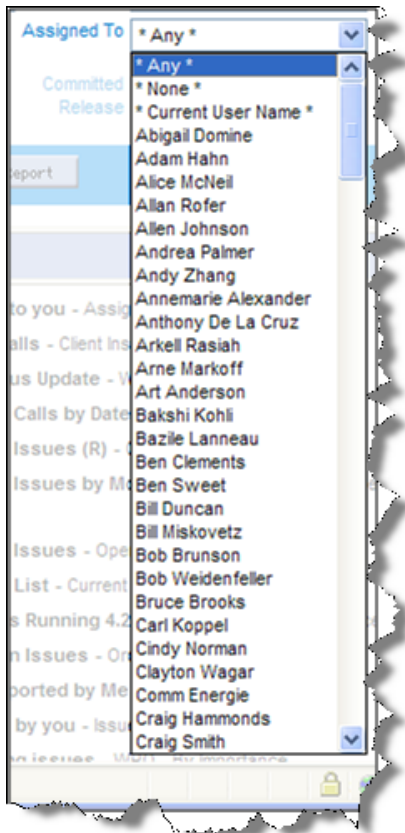
Altering the Display Type of a Field

The data dictionary allows the administrator to make changes to the display type of fields. While these are allowable, they may have consequences if the fields are used within any business rule. For example, you may alter the display type of a field from **List** to **Pop-up**. This is a legitimate change and the rules you have written about the field will still be valid. However, for efficiency, the display information about the field is cached within ExtraView, and changing its display type will render this information out-of-date. ExtraView refreshes the rules and re-caches this information several times a day, but there will be periods when the cache has out-of-date information. If this proves to be a problem, you can simply enter the rules file and press the **Update** button to refresh the information immediately.

Creating & Updating Fields

Consult the pages on Built-In & User Defined Fields and Display Types in conjunction with this page for complete details on how to create and update fields. Access to the Data Dictionary is from the System Configuration tab. The initial data dictionary screen is similar to the following:

Data Dictionary screen

If your current user role does not have any read or write access permission to a field, it will not appear in the list of fields displayed. Normally this is only an issue when there are multiple administrative roles defined, and your role only has permission to a subset of the available fields.

If you have an ExtraView Standard or Professional license, as opposed to the Enterprise license, you are restricted in how many user defined fields you may add to your installation. For these license types, the screen will display how many fields you have created, and will inhibit you from adding additional fields when you reach the limits. For Standard licenses you may add 25 user defined fields, for Professional licenses, the limit is 150 fields. Enterprise licenses may add an unlimited number of fields.

Creating Fields

From the data dictionary, under the **Fields** tab, click the **Add** button. A screen similar to the following appears:

Add a Data Dictionary Entry screen

Follow the instructions on the screen to create your field. For a complete list of instructions and a definition of each of the fields on the screen, click here.

Updating Fields

Click the **Edit** button next to the field that you want to modify. Pressing **Update** within the edit screen will save the changes. Notice that fields with a display type of List have a button which allows you to access their values to add, edit and delete these. Also note that you can edit a field from the Design Center, by right clicking on the field and choosing the Edit option. The tabs within the data dictionary offer access to the field attributes as follows:

Note the **Edit** button for each occurence where the field is found on a layout. Clicking this takes you to the layout opened within the Design Center.

- **Field Properties** - all the basic properties of the field can be altered within this tab. Note that only the properties appropriate for the display type of the field are displayed and available to alter.

Field properties

- **Global Attributes** - The most important use of global attributes is to create field properties that will be inherited and used when the field is placed within all layouts. For example, you might set a style or size for the field as a global attribute, which becomes the default wherever the field is used on an *add* or *edit* layout. This property is inherited by, but may be overridden on any one of these layouts by providing a local layout cell attribute. You often want a field to have the same properties and attributes wherever it is used, and this mechanism provides a means to achieve this, so you do not need to set up layout cell attributes on all the layouts where you use it..

  It is recommended that you do not configure attributes such as VISIBLE IF and REQUIRED IF at a global level.

Global Attributes

When you add a new optional attribute, you will see a screen similar to the following. Information on these attributes is found on the page layout cell attributes.
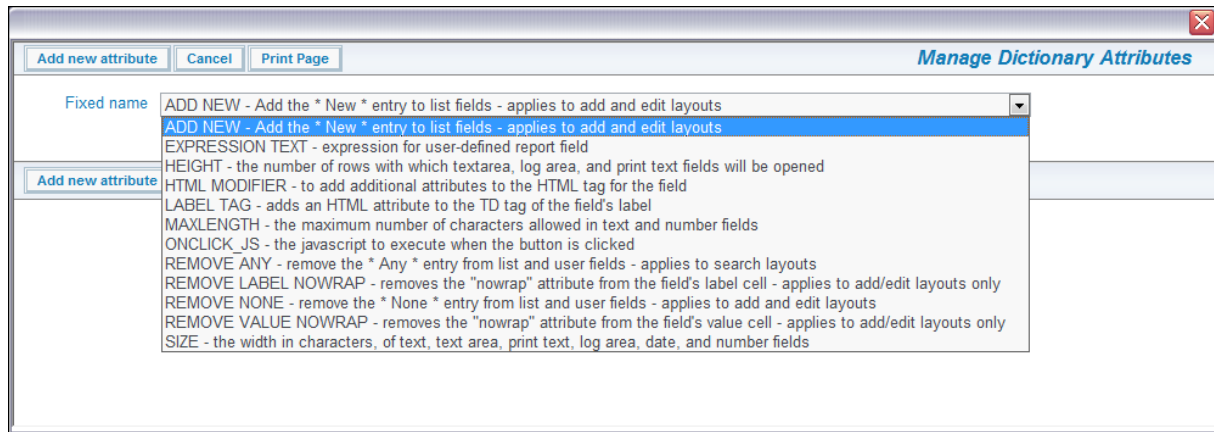
Adding a New Optional Attribute

- **Database** - The Database tab only appears if you are the ADMIN user. It is highly unlikely that you will need to access this tab, and when you change anything on this tab, it should be under the direction of the ExtraView Support team.
- **Permissions** - This tab displays the security permissions for the field, for each role to which you have permission. You can update the permissions for the fied here. For more information on security permissions, visit the page on security permissions.
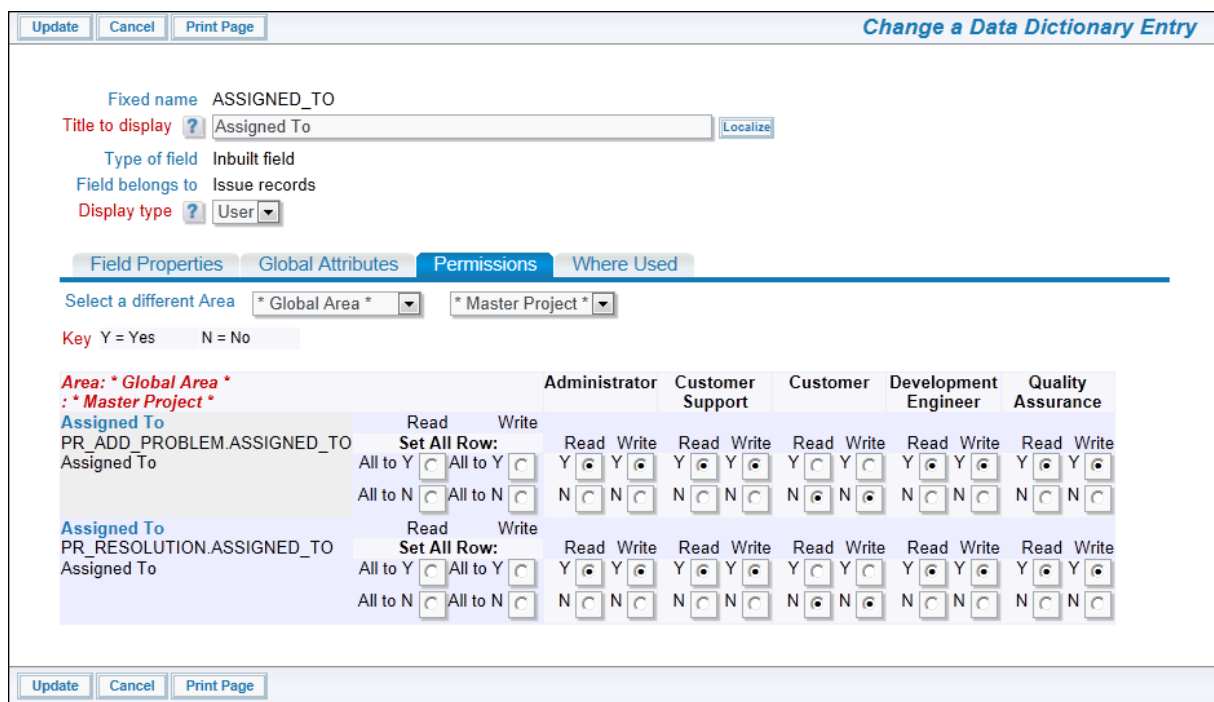
Field permissions

Note that you are viewing the permissions for the object, at the highest level to which you have permissions. If you do not have permissions to view / update the global area / master project level, you will not see these. If you have permission to view / update the object's permissions at the global area / master project level, this will be the top level display of permission radio buttons. You can view and alter the permissions for the level of the currently displayed Business Area and Project by using the Show Dependent Permission Keys, Hide Dependent Permission Keys buttons. You can also switch to a different Business Area and Project's keys by using the select lists at the top of the window

- **Where Used** - This tab allows you to find out all the places where a field is used within ExtraView. For example, you might want to delete a field, but want to know where it is being used before you do so. The first screen within this section shows a list of all the high level places where a field is used, such as **Layouts**. You can drill down on this to see the exact layouts where the field is used.

Where Used

Drilling Down in Where Used

Note the **Edit** button with each occurence of the field on a layout. Clicking this will take the user directly to the Design Center where they can view and modify the layout.

Lists

List administration is the administrative section where you can manipulate all the lists that comprise the metadata values within ExtraView. For example, this is where you maintain lists of business areas, products, modules, and many other items. This includes all the lists that you create as User Defined Fields.

For convenience, the list of items on this screen includes Privacy Groups and User Roles. These lists can be viewed and modified both from this administrative section as well as within the Users section.

The metadata field lists that appear on this screen include all fields with a data dictionary display type of List and Tab.

Lists values may be dependent upon values in other lists. This is termed Allowed Values, and is covered in more depth in the Field Administration section of this guide.

Interest Lists are also maintained for field values within List maintenance. For example, if you want to maintain an interest list for a specific product, or an interest list for Priority 1 issues, then this is completed through the add and edit functions of each list.

Managing Lists

Lists are managed in the same way, from a variety of different places in ExtraView. For example, there are lists of field values, lists of behavior settings, lists of fields in the data dictionary, interest lists and many others.

When you first access a screen with a list it will look similar to this:

Data dictionary list

If you do not have either read or write permission to a field, it will not appear in the above list.

You can manage the list by using the Add button to add a new entry, or by clicking the Edit button at the left hand side of each entry in the list. Some lists such as behavior settings allow you to edit a value, by clicking on the value. If there is a delete capability, edit the list entry and delete the value from there.

Note the **Show Filters** button on the above screen. This is provided to offer a search capability on long lists. When you click on this button, you will see a screen like this:

Data dictionary list with filters

Set the filter column to one of the columns displayed in the list, and then click on a letter to see all the list entries that have the displayed column's first character.

You can enter a search expression, including wildcards, to retrieve the subset of records you want to work with. For the longer lists, you can also use the Export button to create a CSV (Comma Separated Value) file with the list values.

Lists behave differently, according to how many entries are in the list, and the value of the behavior setting named ADMIN_LIST_SIZE, in order to provide a balance between ease of use and scalability, when the number of records retrieved for an administration list exceeds the value of ADMIN_LIST_SIZE.

When the number of entries in a list is less than the value of ADMIN_LIST_SIZE, the list is immediately displayed in its entirety. When the number exceeds the value, a search screen is displayed, with the first character of each entry of the list being available to click on as a shortcut to looking only at these records. Thirdly, if you set a negative number, as opposed to a positive number, the list acts similarly to the second option, but with an abbreviated interface.

Sorting Lists

You can sort lists by clicking on any of the column headings that have a ■ symbol in the heading. When you click on a column heading that is sorted in an ascending order and shows a ▼ then the sort symbol alters to show ▲ and the list is sorted in descending order using this column.

Sorting List Values

You can sort a field list within the add, edit and query screens for each list field by allocating a sort sequence to its entry within list administration. The sort sequence is a number or string of letters. Valid sort sequences would be:

23	2	a	2a	open

If no sort sequences are provided, the list is sorted alphabetically, using the title of the field. If you provide a sort sequence, then the entries to the list are sorted in their ASCII sort order sequence. It is recommended that you establish a sort sequence for a list, allowing room to insert new values in the future. This will give the most flexibility. For example, if you have a list that starts with five values, but may grow, use a sequence such as:

010	020	030	040	050

This will allow you to insert new values between each of the entries, if needed in the future.

Enabling and Disabling List Values

Within all user defined lists, there is the opportunity to disable a value. Once the value has been disabled, it will not appear in any screen in a selectable manner. If an existing record already has the value set, then the user may leave the value in place or they may select a new, enabled value.

Note that checkbox fields are stored internally as lists that are constrained to just two values, typically *Yes* and *No*, *On* and *Off* or something similar. You may not disable a value within a checkbox field.

Field List Values

List values are maintained from one of several places - the **Manage List Values** administration menu, the **List** button by each list type field in the **Data Dictionary**, and by using your mouse to right-click on a list type field within the Design Center. This screenshot shows the **Manage List Values** screen as seen from the **Manage List Values** and the **Data Dictionary**. From within the **Design Center** you access the maintenance screen for each individual field directly:

Managing field lists

Note that once you have one or more values entered against any list type field by adding or updating issues, you are not able to delete the list value. If you want to remove a value from a list after data has been recorded against the value, most lists allow you to disable the value, so it will not appear in the add or edit screens. This allows reporting to show the historic data, even though the value is not currently being used. If a user edits an issue which has a disabled value in a list, they will see a message similar to this:
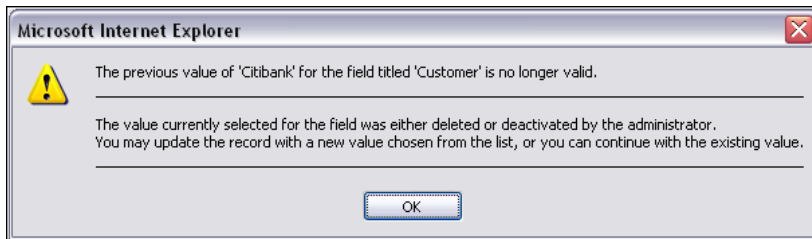
Warning when viewing a disabled value

The user may either update the record with the out-of-date value, or choose a new valid entry from the list. If the out-of-date value is removed from the list and the record updated, no user will be able to choose that value again. The behavior setting named IGNORE_DEACTIVATED_USER_FIELDS may contain a delimited list of fields with a display type of USER. Users will not be warned if they edit an issue which has a field in this list with a deactivated user. If a USER field does not appear in this list, the user will always be warned if the field has a deactivated user, when they edit the issue. There are a few lists which are managed differently:

* Business Area and Project lists. These are used to set up and maintain the business areas and projects within your system. Values within these lists may not be disabled. Security permission keys exist to turn Business Areas on and off for different user roles
* Values within the Category, Priority, Product Line, Status and User Role lists may not be disabled. Note that there are security permission fields that can be used to disable and enable the visibility of the Status field values
* The Product and the Module fields work together, and the Module field administration will ask you to select a valid Product before you maintain any modules. Module field values may not be disabled.

To add an entry to the list, press the **Add** button. To edit an entry, press the **Edit** button by the value you want to edit.

Editing a list entry

Basic UDF Details	These always appear, and you must provide a Title for the field. It's optional to provide a Sort Sequence and an Owner, and you may enable or disable the field value. If the behavior setting named COMPANY_NAME_LIST_UDF has the current field as its value, there is the additional prompt, **Populate this list with nn new list values**. This allows you to synchronize the list with the user's company's names.
Interest List Details	If the field may be used as an Interest List, this section appears. If more sophisticated processing of an interest list for the value is required, then the administrator should use the **Interest List** menu under the **Notification** tab of **Administration**.
Migration Details	It is occasionally useful to inhibit the migration of some list values between instances when performing XML exports / imports. This section allows the administrator to flag specific values so they will not be migrated between instances. Note that this section also displays how a specific list value has been created, for example through the user interface of this utility, or perhaps as a value created by an OBJECT business rule.
Allowed Value Details	This section allows you to define which parents a field value belongs to in allowed value relationships.

Loading List Values from Files

This feature only works to load values into user defined field lists. It does not work with inbuilt fields such as **STATUS, CATEGORY, RESOLUTION** and user field lists. This is not a real limitation as these inbuilt field lists tend to be small lists. Each user defined field list has a button **Import Field Values**. When this is pressed, the administrator is able to load a file of values into the field.

Importing a file of list values

The format of the file to be uploaded is very simple. It is simply a file with a single value per row. The list of values may be ordered and the order retained when it is imported, or it may be unordered, in which case ExtraView will sort the file alphanumerically. When the administrator presses this button, the following dialog appears:

Specifying the file to upload

When the file is uploaded, you will see a sample of the file:

Sample of uploaded file

Note the prompt that allows you to retain the order of the field values in your import file, or allows the list to be sorted alphanumerically.

List Entries with Allowed Values

Editing lists that are the child in an allowed value relationship is a little different. As seen in the screenshot below, the list of all the potential parents is displayed, each with a checkbox. Also, if Business Areas and Projects are enabled, you must select the Business Area and Project for which you want to set the allowed values (see the next section).

Click on each of the parent values that this child is allowed to be related to. The example shows an allowed value relationship between two fields, where the field named *Building* is the parent and the field named *Floor Number* is the child.

Allowed Value relationships are maintained within their child list

Note that this screen allows you to modify the allowed value children for a single parent, on a single screen. This is ideal for making alterations to existing relationships. If you want to add a series of allowed value relationships for different parents, you may find it more convenient to use the administration screen under Allowed Value Types. Both methods will result in the same changes, but sometimes it is more convenient to take a "top-down" rather than "bottom-up" approach.

Business Areas and Allowed Value Relationships

It is important to note that when Business Areas and Projects are enabled, the allowed values that you set are only for the selected Business Area and Project. This is controlled by additional prompts on the maintenance screen as shown above. You must set up the allowed value relationship for each Business Area and Project.

Note: If the Business Area field is the parent field in the allowed value relationship then you should be aware of how the interaction with the fact that allowed values are dependent upon Business Area works. The implication is that you can only define the child in the relationship to either belong, or not belong to the currently selected Business Area. There is no meaning to allowing the child to be selected with other Business Areas. Therefore, you will only see the currently selected Business Area with a checkbox in the list of possible allowed values.

Note: You can set up a set of default values for a relationship. These will be used, unless overridden by a specific set up within any Business Area and Project.

End-User List Management

End users may add new values to lists under certain circumstances, where the administrator has enabled the feature with security permissions, and has turned on the feature on specific layouts.

To an end user, the ability to add a new value to a list appears like this:

Adding a new value to a list

When the end user selects the * **New** * entry in the list, a window pops up, asking for the details of the list. The appearance of this popup varies a little, according to whether the list is an inbuilt list field (such as **product_name**), a child list such as **module_id** or **release_found** or **release_fixed**, or is a user-defined field list. The inbuilt fields require the addition of the field's fixed name, plus its title. Dependent child fields require the parent field to be selected, and the relevant fixed name and/or title. User defined fields only require the title to be input. This is an example of a popup:

Adding a new value from a list field

To configure a field to use this feature, take the following steps. Note that only a small number of inbuilt fields may be used in this way, but any user defined field list may be configured.

Field	Security Permission Key
PRODUCT_NAME	CF_PRODUCT
MODULE_ID	CF_MODULE
RELEASE_FOUND, RELEASE_FIXED	CF_PRODUCT_RELEASE
All UDF's with a display type of List, with the exception of multi-valued lists	CF_UDF_LIST

* For the user role or user roles that you want to be able to add new values to a list, give the security permission key in the above table write access
* The * **New** * entry is added to the list by adding a layout cell attribute to the field on the appropriate *add* and/or *edit* layout. The layout cell attribute is named **ADD_NEW - Add the * New * entry to list fields - applies to add and edit layouts**.
* The feature works with UDF list and popup fields on repeating rows as well as on standard list fields
* The feature works with multi-valued list and popup fields on standard list fields only. It does not work with multi-valued list fields on repeating row fields
* The feature does not work with fields being used on reports with the Quickedit mode
* The feature does not work with other types of list fields such as tab and radio button.

Aliased Lists

Within the data dictionary you can create list type fields which are the alias of other list type fields. ExtraView will keep the values of the two (or more) lists synchronized. However, you can only edit values within the original list. If a field has other fields that are aliased to it, you will see a message when editing the list, similar to:

This field is also the alias for 'Customers', 'OEM Customers'. Modifying the values in this list will modify all the values in all the lists.

If the field is the alias of another field, you are not able to edit its values. There will be a message similar to:

This field is an alias of the field named 'Customer'. You may not edit the list values for this field. Please go to the alias list to make changes to the entries for this list.

You can create an alias list for fields with a display type of List, Popup and Tab.

If you delete a list which is the original of a list with an alias, then the aliased list becomes an independent, stand-alone list. Of course, you may only delete a list when there are no values with data stored against it.

Lists with Enabled Interest Lists

When an interest list for a field is enabled, the administrator has the option to create an interest list entry for the value they create, when they are adding a new value to the list. Under these circumstances, the screen to add an item to the interest list also contains two additional fields, **Create a global interest list for this value** and **Add users to this interest list**. This is shown in the following screenshot.

Creating an interest list entry while adding a new value to a list

Lists with Special Properties

Business Areas

Business Areas are created as a list, like all other lists. From an administrator's perspective, a Business Area is created like any other list. Internally, many actions are taken to prepare ExtraView for the new business process you have just added. In summary, these are –

- If DISALLOW_AREA_0_DATA and DISALLOW_PROJECT_0_DATA are set to YES, then two projects are created automatically. The first is named by appending *Project Defaults* to the title of the new business area. The second is named by appending *Project Data* to the title of the new business area. The *Project Defaults* is the project with the PROJECT_ID of 0. This is the project from which other projects inherit their characteristics. The *Project Data* is the project in which the issue data will be stored, assuming you do not create other projects within the business area. You may alter the titles of these projects for your own purposes. **It is not normal practice to set these two settings to anything other than a value of YES. These settings are mainly used for backwards compatibility with sites created prior to ExtraView version 5.**
- Security permission keys are created for each user role for the new business area. This allows you to enable or disable the business area for each user role within ExtraView. These security permission keys are named with the following convention when you look at the Grant Security Privileges screen – AREA.id (area_title) where id is the internal ID of the business area and area_title is the title of the business area. The effect of the security permission keys for user roles is to enable / disable the ability for a user within a specific role to be able to view and / or select the business area on all the *add*, *edit* and *query* functions. You can lock any user role out of any business area with this facility.
- The layout editor will make the new area available to create new layouts. Until you create a new layout within the new area, the layouts from the Global Area will be inherited.
- The new area will be available within the status rules administration screen, to allow you to set status rules for the new area. Until you create status rules for the new area, the rules will be inherited from the Global Area.
- You can define multiple roles with access to administration. However, each of these roles can be permitted to see only a specific combination of business areas.

- Note: Even if a user role does not have permission to see a business area, that role may still be able to retrieve records from that area. For example, a query that asks to retrieve all the "open" issues irrespective of business area will do just that, including records from areas where no permission exists to see the area. You can prevent this behavior by setting a layout cell attribute on query layouts with the FIELD REMOVE NONE attribute. This will force the user to select one or more areas to which they have access as part of their query.

Business Area administration screen

Business Areas and Projects Relationship

Business Areas are linked to Projects, and vice versa. The relationship is that many projects can belong to a single business area.

ExtraView Corporation does not recommend that the global business area (i.e. the business area with the internal ID of 0) is used to store data. It should be used primarily to store metadata about the installation, such as the security permissions to be used as defaults throughout the system, and the layouts that are common across the entire system. Taking this step will simplify the installation of ExtraView when it is to be used for multiple tracking purposes.

The same principle holds true for the global project within each business area, and we recommend that this not be used to store data.

However, this may be cumbersome if you only intend to use ExtraView for a single tracking function and you have the option to alter this behavior. To accommodate the differences, there are two behavior settings in the Workflow Settings menu of Administration.

Setting Name	Default Value	Purpose
DISALLOW_AREA_0_DATA	YES	When this setting is NO, issue data may be entered into AREA 0. This is provided for backwards compatibility for installations created prior to version 4.2. Installations created with version 4.2 and greater should not allow issue data to be placed in AREA 0. With installations from 4.2 onwards, this should be set to YES
DISALLOW_PROJECT_0_DATA	YES	When this setting is NO, issue data may be entered into PROJECT 0. This is provided for backwards compatibility for installations created prior to version 4.2. Installations created with version 4.2 and greater should not allow issue data to be placed in PROJECT 0. With installations from 4.2 onwards, this should be set to YES

The default business area and project lists in the list modification screens do not include entries for the global area (0), if DISALLOW_AREA_0_DATA is YES or project (0), if DISALLOW_PROJECT_0_DATA is YES. In addition, when adding or editing an area, a warning is issued if DISALLOW_PROJECT_0_DATA is YES and the area added or modified has no projects beyond the global project.

Project Lists

The PROJECT field is the child of the AREA field. This relationship is more fully explained in the page on Business Area lists. The default project is used as the default project for all layouts, fields and values within the entire project, and each of these objects may be inherited from the default project to other projects within the business area. Each business area will have a minimum of two projects, the default project and a data project. The default project only stores the default metadata for the entire set of projects within the business area, and never stores any issue data. The remaining projects may store metadata that overrides the metadata within the default project as well as issue data for that project.

By convention, the default project always has a value for the field PROJECT_ID of zero. When a business area is created, the first two projects are named XXXX_DEFAULTS and XXXX_DATA where XXXX is the name of the business area. When you are modifying the data for a project, there are a number of behavior setting values you can override. The behavior settings are default-wide settings; if you provide entries within the default project, the values become the values for the entire business area. If you provide an entry within any project other than the default project, then the value is only used for that project. The behavior settings that can be modified in that way are used for email notification. The list is:

- AD_HOC_EMAIL_FROM_ADDRESS
- AD_HOC_EMAIL_FROM_SENDER
- EMAIL_ADMINISTRATOR_NAME
- EMAIL_ADMINISTRATOR_USER_ID
- EMAIL_FROM_USER_NAME
- EMAIL_FROM_USER_ID
- SUPPRESS_STANDARD_EMAIL_LIST

Status List

The status list drives workflow, and is explained fully in the section within this guide entitled Workflow Setup. The list has the following special properties in addition to driving the workflow:

- Security permission keys are created for each status list entry. These allow you to control the visibility of the status entry in an absolute fashion, for the list entry within each user role
- There is a behavior setting named STATUS_CLOSED_NAME. This should be set to the status name for the value that means "closed" in your installation. This is usually CLOSED, but you may alter this as you need. You should not alter this value on a production system, as you will lose the historic tracking related to the dates that issues were closed. Set this value at the time of implementation only.
- The status list will be populated on *add* and *edit* screens according to the rules set within the workflow setup section. Fundamentally, the workflow can be different or the same for any combination of business area, project, and user role or product.
- Although ExtraView allows duplicate titles in lists it is strongly advised that you do not create status values with duplicate values. If you do so, ExtraView will give a warning.

Deleting a Status Name and Value

Like all field values, you must delete or remove all dependent data before you can remove the status name and value. The following is the suggested order of removing a status name and its value:

- First you must ensure that no issues exist with the status value. With an empty database, nothing needs to be done. If values exist, use the mass update feature within reporting to select all records with the value you want to remove, and then alter the value to a new value you have created, or to one of the status values you are keeping
- You must next remove all the status change rules that use the value you want to remove. With this version of software, you must remove these individually. In future versions this will be simplified
- Check that the value you are removing is not the behavior setting named STATUS_CLOSED_NAME
- Check that the value you are removing is not the default value for the STATUS field within the data dictionary
- Finally, remove the status value from the Status List maintenance screen

Product Lists

Product Line Lists

Note:This field is deprecated and should only be used within existing applications. Newly configured applications should substitute the product line list functionality by creating a new user defined list field, then creating an allowed value relationship were this field is the parent, and the **product_name** field is the child field. Product lines (the **product_line** field) have a special relationship with products (the **product_name** field). Any single product may belong to any number of product lines.

Adding a product to one or more product lines

When adding or editing a product line, you will see a list of all products, each with a checkbox. To add a product to a product line, simply check the box. When you place both the product line and product fields on an add screen or edit screen, they interact as follows:

Condition	Behavior
No allowed values on the *product_name* field	When a product line is selected, the screen refreshes and the list of products will reflect the values of products in the list for the product line
product_name is the child in an allowed value relationship	When a product line or the parent field in the relationship is chosen, the screen refreshes and the list of products displayed will be those that both belong to the product line, and are a child in the allowed value relationship

Product Name Lists

Product name lists (**PRODUCT_NAME**) have an additional attribute that gives the ability to deactivate a product. When you deactivate a product, the product will no longer appear within the select list on the Add and the Edit screens. If the user edits an issue with an inactive product, then they will receive a warning that it is no longer a valid option. The user may update the issue with the inactive product, or they can select a new active product.

Deactivating a product

Module Name Lists

Use of the built-in MODULE_ID field is discouraged for new configurations. This field and its relationship with the built-in PRODUCT_NAME field will be deprecated in a future release. Support for existing use of the field will continue at this time. You should configure a UDF in place of the MODULE_ID field.

Module names are managed a little differently to other lists. This is because there is a built-in relationship between **PRODUCT_NAME** and **MODULE_ID**. ExtraView does not use an Allowed Value relationship to manage this.

To add or modify a *Module*, click **Module Names** on the Lists tab. The following screen appears:

Module Names screen

Click the **Add** icon to create a new Module Name. Since *Module* depends on *Product*, you will have to choose an applicable *Product* from the drop-down list to associate with your new *Module*. Fill in the appropriate fields and click the **Update** button.

Adding a Module Name

To modify an already existing *Module Name*, return to the Administration menu, click the Manage Field List Data tab, and select *Module Name*, as in step 4. Select the *Product* that pertains to the *Module* that you want to edit.

If you have set the display type for the MODULE_ID field to be popup, then searching for a value is slightly different than for other fields. The search mechanism utilizes the **name** value of this field, not the value of the **title**. In most instances, the name and title of the field values for modules are the same, but in the cases where they are not identical, the popup list will display **name_value(title_value)**, allowing you to distinguish between the name and the title.

The owner of the module will automatically receive email notification when the module is selected on an issue. Also, there are two behavior settings that affect the way notification happens with the *module* field.

Setting	Purpose
EMAIL_MODULE_OWNER_ALWAYS	Email the module owner irrespective of whether they are assigned to an issue. Valid values are YES and NO
LINK_MODULE_USER	Links the module owner field to the specified user field. This will automatically set the value of the user field specified to the value stored in the *Module Owner* field This can have a value of ASSIGNED_TO, CONTACT or OWNER

Editing an Existing Module

Click the **Edit** button next to the *Module* that you wish to change. On the screen that appears, you can edit the values, and then **Update** the record.

Release Found & Release Fixed Lists

Use of the built-in RELEASE_FOUND and RELEASE_FIXED fields are discouraged for new configurations. These field and its relationship with the built-in PRODUCT_NAME field will be deprecated in a future release. Support for existing use of the fields will continue at this time. You should configure UDF's in place of the RELEASE_FOUND and RELEASE_FIXED fields

.

Release Found and Release Fixed Maintenance Screen

These lists have several special properties:

- They point to the two inbuilt fields RELEASE_FOUND and RELEASE_FIXED
- These fields belong to repeating row records. They should not be used on layouts other than repeating row records. If you want to use a field or fields for the functionality of RELEASE_FOUND or RELEASE_FIXED on layouts such as an add or edit layout, then define UDF's and use these
- These fields use PRODUCT_NAME as their parent value, and therefore separate lists can be defined for each product in the system
- Both lists adopt the same values and are maintained in a single place within the List maintenance screen
- Releases may be made inactive. Once inactive, they will not appear within lists as you add or update an issue. However, if you edit an issue where the release found or release fixed is inactive, you will see a warning, which you can choose to ignore
- The sort sequence of the Release Found and Release Fixed lists can be sorted in an ascending or descending order. This is controlled with the behavior setting named RELEASE_SORT_ORDER on the Workflow Settings administration menu
- If you have set the display type for the RELEASE_FOUND and RELEASE_FIXED fields to be popup, then searching for a value is slightly different than for other fields. The search mechanism utilizes the **name**value of these fields, not the value of the **title**. In most instances, the name and title of the field values for RELEASE_FOUND and RELEASE_FIXED are the same, but in the cases where they are not identical, the popup list will display **name_value(title_value)**, allowing you to distinguish between the name and the title.

Adding a new entry to the release list

Company Name Lists

It is often useful to be able to maintain a reference list of Company Names. A User Defined Field list can be created and linked to the User Account Maintenance screens. This may prevent the duplication of multiple versions of a single company name. For example, you may otherwise enter a company name as Superior Corp, Superior Corporation, Superior Corp., etc.

To facilitate this:

- Create a user defined field which is to retain the list of company names
- Place the name of this user defined field in the behavior setting named COMPANY_NAME_LIST_UDF
- When you add a new user to your site, you will then see a list of all current company names, and you may then select from a previously entered name or you may create a new one.

Use of the COMPANY_NAME_LIST user defined field

When you edit the list defined by COMPANY_NAME_LIST_UDF, it has a special property as shown in the screenshot below:

Updating the COMPANY_NAME_LIST field

This indicates that when you add the new value to the list, that you can synchronize the list values with the list of users. Normally you should select this checkbox and perform the synchronization.

Relationship Group List

It is often useful to be able to maintain a reference list of Relationship Groups. A User Defined Field list can be created and linked to the Relationship Group Maintenance screen. To facilitate this:

- Create a user defined field which is to retain the list of Relationship Groups
- Place the name of this user defined field in the behavior setting named RELATIONSHIP_GROUP_LIST_UDF
- When you add or update issues to a relationship group, you will then see a list of all current relationship groups, and you may then select from a previously entered name or you may create a new one.

Special Purpose Fields

Most fields in the data dictionary refer to a field that may be placed on the *add* layout, the *edit* layout, a *search* layout or a report layout. Examples are inbuilt fields such as PRODUCT_NAME and ID, or UDF's such as COMMENTS and DESCRIPTION.

However there are a number of fields whose purpose requires some additional explanation.

ID and ALT_ID Fields

Each item added to the ExtraView database is automatically given an ID. The ID is a sequence which is incremented by one for each issue that is inserted into the database. The field is always read-only to users and developers. It may be used as a filter on reports. The ID is the primary key to all items. The sequence is applied without reference to which Business Area or Project to which the item belongs. For some purposes it is useful to have a unique sequence per Business Area or per Project, with the items within each Business Area or Project being given sequential numbers.

Further, it can be advantageous to provide formatting to this identifier to easily distinguish items in different Business Areas/Projects. This can be achieved through the use of the ALT_ID field. The ID field is still maintained automatically within ExtraView, but the mechanism described here can be set up to provide a unique sequence within the ALT_ID field. You should only set up this mechanism on a new database as existing items are not back-populated with the ALT_ID values. To set up separate sequences for each Business Area:

* Within the data dictionary entry for the ALT_ID field, place a pattern in the default value of ALT_ID
* You should not use the characters ‹ and › within ALT_ID titles

Entering the ALT_ID pattern

This pattern is created as follows:

- **$$AREA.SEQ$$**. This is the unique sequence for the area within which the item is being created. You should always include this
- **$$AREA.SHORT_TITLE$$**. This is an optional set of characters entered into the definition for the Business Area. For example, you may provide a short title in the Business Area maintenance screen of CR to define a change request and a short title of FR to indicate a feature request. You enter the short title into the **Abbreviated title for ALT_ID** field

Entering an abbreviated title for the ALT_ID

- ○ Any constant string. For example, you may include punctuation characters such as : and –
- ○ Replaceable variables of the form **$$*FIELD_NAME*$$**. These allow the use of other fields on the input form to become part of the ALT_ID
- ○ You must not use a space character within the pattern. This ensures that keyword searching is kept simple within installations that have enabled ALT_ID
- ○ Set a starting number for the sequence at the prompt **Starting sequence number for ALT_ID**
- You must set the behavior setting **ITEM_ID_DISPLAY** to a value of ALT_ID
- Place the ALT_ID field onto all appropriate layouts as opposed to using the ID field. Typically this will be the *add, edit, detailed report, Quicklist* and other similar layouts. You may still set a layout cell attribute of FIELD/LAYOUT VISIBLE IF ID is not null to hide the ALT_ID before the item is saved.
- If you clone an issue, then the new ALT_ID is generated from the default pattern if it is writeable, and the default value for ALT_ID is non-empty.

To use this feature for Projects, within the definition for the PROJECT field, use the above instructions with **$$PROJECT.SEQ$$** as opposed to **$$AREA.SEQ$$**.

Examples:

ALT_ID Default Value	Example Output
$$AREA.SHORT_TITLE$$:$$AREA.SEQ$$	FR:12345
$$AREA.SEQ$$($$AREA.SHORT_TITLE$$)	12345(CR)

Fields

Field Name	Title	Definition
ALT_ID_START_SINCE	Start Issue ID	In conjunction with ALT_ID_STOP_SINCE, this field is used to establish a search range for issues on any query filter layout. Both this and ALT_ID_STOP_SINCE should be created on the layout. The user can enter the start of a range to search for in this field. If it's left blank, but the ALT_ID_STOP_SINCE field has a value, then all issues with values less than ALT_ID_STOP_SINCE will be returned
ALT_ID_STOP_SINCE	Stop Issue ID	In conjunction with ALT_ID_START_SINCE, this field is used to establish a search range for issues on any query filter layout. Both this and ALT_ID_START_SINCE should be created on the layout. The user can enter the end of a range to search for in this field. If it's left blank, but the ALT_ID_START_SINCE field has a value, then all issues with values greater than ALT_ID_START_SINCE will be returned

Text Type Field Special Properties

URL's and Links

A special end-user property of text fields is that if they enter a URL into the field, then when the field is rendered on a report, or the field is rendered in a read-only mode, the URL becomes a hyperlink to the URL. For example, if you enter **http://www.mycompany.com** or **www.mycompany.com** into a text field, then it will appear as **http://www.mycompany.com** or **www.mycompany.com** on a report, and if you click the text, a new window will open at the address of the URL. This behavior is extended one step further. Assuming that the title for the ID field in your instance is **ID #**, then entering **ID # 12345** into a text field,

will provide a link on reports and read-only versions of the field, to the detailed report view of the issue with the ID of 12345. If the title to the ID field is **Report Number**, then the end user would enter **Report Number 12345** for the same effect.

Rendering HTML within a field

It is possible to render HTML within a field on a report which has a display type of either text area, print text or log area. This is separate from the capability offered with fields with a display type of HTML Area, which is the recommended way of handling fields that will routinely hold HTML. The normal behavior with text area, print text and log area fields is that ExtraView will "escape" the sent being sent to the report in the browser. For example, HTML being sent to a field of these types will be shown as HTML and not rendered within the browser. This is the normal mode of working, as ExtraView is often used to track defects or bugs that refer to HTML code. There are some circumstances when you may want to actually view the code rendered as HTML within the text field. In order to do this, place the following text within the text field as you create –

<!-- generated valid html - don't escape! -->

All code following this, until the end of the field will be rendered in the browser. It is the user's responsibility to ensure the code placed following the above text is valid HTML and will render appropriately. One interesting side effect of this feature is that you can legitimately place a complete program, along with buttons, forms and JavaScript within a field!

The field can be populated with code within user custom routines. For example, a field on an *add* or *edit* screen can contain the complete results of a query in report format.

Attachments

When you are adding attachments to an issue from both the *add* and *edit* screens, you use the PR_ADD_PROBLEM security permission key. This is not immediately obvious with regard to the *edit* screen.

Also note that the ATTACH_FILE_DESC permission key has special properties with regard to fields with a display type of document and image.

Field Name	Title	Definition
ATTACHMENT	Attachments	This field is used on reports to display all the fields that comprise the attachment record. All fields that may be placed on the attachment record are displayed, assuming they have read permission.
ATTACHMENT_ID	Attachment ID	The internal ID for the attachment. Used within the ATTACHMENT field and the ATTACHMENT_HISTORY layout.
ATTACH_CHANGE_TYPE	Change Type	The type of attachment change. Used on the ATTACHMENT_HISTORY layout.
ATTACH_CONTENT_TYPE	Attachment Content Type	The mime type of the attachment. Used within the ATTACHMENT field and the ATTACHMENT_HISTORY layout.
ATTACH_CREATED_BY_USER	Attachment Created By	The person who created the attachment . Used within the ATTACHMENT field and the ATTACHMENT_HISTORY layout.
ATTACH_DATE_CREATED	Attachment Date Created	The date the attachment was created. Used within the ATTACHMENT field and the ATTACHMENT_HISTORY layout.
ATTACH_FILE_DESC	Attachment File Description	The description of the attachment. Used within the ATTACHMENT field, the ATTACHMENT_HISTORY layout and on fields with a display type of IMAGE and DOCUMENT. This field has some special properties.

- If PR_ADD_PROBLEM.ATTACH_FILE_DESC is writable, then the description for all upload objects is required on the *add* screen
- If PR_ADD_PROBLEM.ATTACH_FILE_DESC is writable, then the description for all upload objects is permitted, but not required on the *add* screen
- If PR_ADD_PROBLEM.ATTACH_FILE_DESC is neither writable nor readable, then the description for all upload objects is not shown on the *add* screen
- If PR_RESOLUTION.ATTACH_FILE_DESC is writable, then the description for all upload objects is required on the *edit* screen

- If PR_RESOLUTION.ATTACH_FILE_DESC is writable, then the description for all upload objects is permitted, but not required on the *edit* screen
- If PR_RESOLUTION.ATTACH_FILE_DESC is neither writable nor readable, then the description for all upload objects is not shown on the *edit* screen.

ATTACH_FILE_NAME	Attachment File Name	The file name of the attachment. Used within the ATTACHMENT field and the ATTACHMENT_HISTORY layout.
ATTACH_FILE_SIZE	Attachment File Size	The file size of the attachment . Used within the ATTACHMENT field and the ATTACHMENT_HISTORY layout.
ATTACH_PATH	Attachment Path	The original path in which the attachment was stored . Used within the ATTACHMENT field and the ATTACHMENT_HISTORY layout.
ATTACH_HIST_TIMESTAMP	Attachment History Timestamp	The timestamp of attachment history. Used on the ATTACHMENT_HISTORY layout.
ATTACH_LAST_DATE_UPDATED	Last Date Updated	The last date the attachment was updated. Used on the ATTACHMENT_HISTORY layout.
ATTACH_LAST_UPDT_COMPANY	Company	The last date the attachment was updated. Used on the ATTACHMENT_HISTORY layout.
ATTACH_SELECT	Select?	An optional field. When the security permissions for this field are turned on, this causes a column on the attachment layout to be made visible, with a checkbox by each attached file added to the issue. This works in conjunction with another field, EMAIL_SELECTED. This field appears in the notification layout. When a user checks the EMAIL_SELECTED field, then each of the file attachments checked in the list of attachments is sent out as an attachment to the notification. In this way, file attachments can optionally be sent to users connected with an issue, when the issue is either added or updated. This field is also subject to control with the ADD and COPY business rules, as well as being available within user custom code.
ATTACH_THUMBNAIL	Thumbnail	Preview a thumbnail image of the attachment. Used within the ATTACHMENT field and the ATTACHMENT_HISTORY layout.
NUMBER_OF_ATTACHMENTS	Number of Attachments	This field may be placed on reports, or used as a report filter and provides a count of the number of attachments on an issue. For example, you may use this as an advanced filter to show all the issues that have attachments, when used as **Number of Attachments > 0.**

Behavior Settings

Setting Name	Default Value	Description
ALLOWED_ATTACH_SEARCH_FILE_EXT	application/pdf,application/msword, application/vnd.msexcel,text/csv, text/html,text/plain,text/rtf	This is a list of the allowed mime types for attachment files to be searched for keywords, when the user checks the **Search Attachments** box. If the file in the list to be searched does not have one of these mime types, it is skipped. This is used to skip searching of files such as image files, to speed the search process
ATTACHMENT_CONTROL_FIELDS	NO	This setting is only used in conjunction with the Java user custom exit named ucAllowAttachmentOperation. This setting enables the functionality within this method. When the value is NO the user custom exit is not called. When the value is YES the code within the method is called and acted upon

ATTACHMENT_REPOSITORY_DMAX	999	The maximum number of files or directories that will be created under one node of the external directory structure. The default for this value is 999. It is not likely that this value needs to be altered
ATTACHMENT_REPOSITORY_OPT	INTERNAL	This setting controls whether attachments are stored internally within the database, externally on the file system, or in some combination of the two methods. If the value of this setting is INTERNAL, then all attachments are stored internally within the database. If the value of this setting is EXTERNAL, then all the attachments are stored on the external file system. Alternatively, you may provide a comma separated list of file extensions and then all files with these extensions will be stored externally while all others will be stored in the database. For example, if you set the value of this setting to "avi,png,gif,jpg" then files of these types will be stored externally. This strategy leaves the files stored internally in the database as searchable by keywords, while the image and video files are stored externally. The default for this value is INTERNAL. Also, ensure that the setting ATTACHMENT_REPOSITORY_ROOT is set correctly before storing attachments outside the database
ATTACHMENT_REPOSITORY_ROOT		The name of the directory on the file system where file attachments will be stored. Before any attachments are stored externally, you must also provide a valid setting for the setting ATTACHMENT_REPOSITORY_OPT. You must also ensure that the path is valid from the application server(s) that are running ExtraView, and that you have all the permissions to read and write to the storage. You must also ensure that you set up a separate backup method for this external storage as backing up your database will no longer backup the file attachments
ATTACH_SELECT_CHECKBOX	UNCHECKED	This setting controls behavior of the Attachment Select box (ATTACH_SELECT) on the Add and Edit layouts. Valid values are UNCHECKED, CHECKED and CHECKED ON ADD. A value of CHECKED or UNCHECKED will set the checkbox for an attachment on entry to the screen or when a new attachment is added. A value of CHECKED ON ADD will cause the checkbox to be checked only when the attachment is first added, otherwise it will be unchecked
BLOCK_ATTACHMENT_EXTENSIONS		If a user uploads an attachment, or a file into a document field or a file into an image field, with an extension that is in this list, the file will not be stored on the server, and the user will receive a message that informs them that they are not allowed to store files

		of that type. The default is that all files of all types are stored. The list of file extensions is comma delimited. For example, you might set a value of *exe, js, com* to prohibit end users from uploading files of these types.
COPY_ATTACHMENT_ON_CLONE	YES	When this setting is YES, attachments will be copied to the new issue when an issue is cloned
DEFAULT_ATTACHMENT_CHARSET	UTF-8	The default character encoding for files being uploaded to ExtraView. This value is used to select the initial value presented to the administrator when creating a new user
SEARCH_ATTACH_THRESHOLD	100000000	The size of attachments to be searched, before the user is alerted of performance cost. If the size of attachments to be searched exceeds this value, then the user is shown a dialog box and asked to confirm whether they want to proceed with the search
THUMBNAIL_MAX_SIZE	150	This is the maximum number of pixels in either the horizontal or vertical direction to which thumbnail images will be generated. The aspect ratio of the original image will be retained. Thumbnail images are generated for file attachments and for fields with a display type of image. If you change this value, existing thumbnail images will remain at their original size and new thumbnails will be generated at the changed value

FILTER CHILD VALUES Field

The FILTER_CHILD_VALUES field is a special field that controls the ability to return, or not return, all matching repeating rows from a query. When this field is placed on a search query layout, a checkbox will appear to the end user when they are using the Standard Query screen. When this is not checked, a query that generates repeating row records will return all the repeating rows within an issue. When it is checked, the query will only return repeating row records that match the remaining filters in the query.

If the user has selected Advanced Query mode, then the option appears a a checkbox at the top of the filter selection list.

Before this field can be placed on a search layout, the behavior setting named FILTER_CHILD_VALUES must be set to have a value of USER.

STATUS Field

The STATUS Field

The STATUS field has several properties that separate its functionality from other fields, including other inbuilt fields. As its name implies, it is used to track the *status* of issues as they transition through the workflow you configure. The handling of the STATUS field has these properties –

- The STATUS field is used on issues to manage workflow. There is a complementary field available for repeating rows named RELEASE_STATUS. Both fields share the same set of values and the same workflow. The values for both lists are defined within the STATUS field list
- You may set up workflow around the transitions of the states of an issue between the STATUS values. This workflow may be set up role-by-role or product-by-product. It may also be set up globally and inherited by different business areas and projects, or it can be set up individually within any business area and project combination

- The STATUS field is unique in that each value (such as *Open* or *Closed*) is given two security permission keys, one for adding and one for updating issues. If the user does not have write permission to a STATUS add value, then the value will not be displayed in the STATUS list. If the user does not have permission to the key when updating an issue, they will still see the value in the list, but they may not select that value from the list and update the issue. They may still edit and update the issue with other STATUS values to which they do have permission. They simply cannot update the issue to a value of the STATUS field to which they do not have permission. This functionality is desired as a user may have permission to update an issue, but they may not have permission to update to an individual STATUS value. When in query screens, the user will see all STATUS values, irrespective of whether they have read or write permission. If you require to hide STATUS values from users, consider using an allowed value relationship, where STATUS is the child, and use this capability to control the visibility of individual STATUS values.

STATUS_TRANSITION Field

This is defined within the data dictionary as a UDF with a display type of **Custom**. This field provides an alternative to the STATUS field, offering a visual way of highlighting the status of an issue and showing exactly which statuses are valid to transition the issue to, obeying the workflow and status change rules. It looks like this:

The STATUS_TRANSITION field

The current status is highlighted. The user can click on any of the statuses shown, in order to transition the issue to that status. For this to operate, you should retain the STATUS field on the same layout, but you may hide this from the user, using a layout cell attribute of FIELD STYLE. For example, setting a value of **display:none** will hide the STATUS field list. Setting an alternate title of a space character will hide the label to the STATUS field on this form. Only the STATUS_TRANSITION field will be displayed in an editable form - actually it's a clickable field.

If the STATUS field is displayed on the same screen as opposed to being hidden, then both the STATUS and STATUS_TRANSITION fields will work in tandem.

You may alter the width and height of the field by setting a width and height (both measured in pixels) into the default value of the field in the data dictionary. Use a delimiter of a semi-colon between the values. For example, a default value of **1000;100** will set the size of the field to 1,000 pixels wide and 100 pixels high. Note that you must be precise with the syntax of the default value as no error checking is performed. If you do not provide a default value, a width of 900 pixels and a height of 80 pixels will be assumed.

KEYWORD Field

This field is used on search layouts to provide a text input box for the purpose of enabling keyword searches through the ExtraView database. This field has the usual security permission key associated with the field, named PR_RESOLUTION.KEYWORD. However, there is a second security permission key, named PR_RESOLUTION.SEARCH_ATTACHMENTS. This also applies to the KEYWORD data dictionary field. If a user role has permission to read this key, then a checkbox appears directly beneath the KEYWORD field. This controls the ability to search attachments as part of the query.

Expression Fields

Expression fields are used to calculate, and display the results of the calculation, on a column report, a Detailed report or a Quicklist report. For example, you might use a numeric expression to calculate the sum of two or more fields within an issue. The Default value of the expression field may store an expression that will be evaluated as the default when the field is placed on a report. See the section in this guide for Reporting and Querying for a discussion of valid expressions.

Expression fields with a type of Duration are used as filters in queries, where they compute the value associated with a **date** or **day** field. For example, you might use an expression to filter all the issues with a *date created* of *last month*.

This list shows the built-in expressions provided with ExtraView. Adminstrators may create their own expressions. It is recommended that you do not alter the value of the built-in *duration* expressions. You may add additional expressions of any type.

Data dictionary field name	Title	Type
EVDTF_LAST_FISCAL_QUARTER	Last fiscal	DURATION

	quarter	
EVDTF_LAST_FISCAL_YEAR	Last fiscal year	DURATION
EVDTF_LAST_FQ_TO_DATE	Last fiscal quarter to date	DURATION
EVDTF_LAST_FY_TO_DATE	Last fiscal year to date	DURATION
EVDTF_LAST_MONTH	Last month	DURATION
EVDTF_LAST_MONTH_TO_DATE	Last month to date	DURATION
EVDTF_LAST_WEEK	Last week	DURATION
EVDTF_LAST_WEEK_TO_DATE	Last week to date	DURATION
EVDTF_LAST_12_MONTHS	Last 12 months	DURATION
EVDTF_NEXT_4_WEEKS	Next 4 weeks	DURATION
EVDTF_NEXT_FISCAL_QUARTER	Next fiscal quarter	DURATION
EVDTF_NEXT_FISCAL_YEAR	Next fiscal year	DURATION
EVDTF_NEXT_MONTH	Next month	DURATION
EVDTF_NEXT_WEEK	Next week	DURATION
EVDTF_THIS_FISCAL_QUARTER	This fiscal quarter	DURATION
EVDTF_THIS_FISCAL_YEAR	This fiscal year	DURATION
EVDTF_THIS_FQ_TO_DATE	This fiscal quarter to date	DURATION
EVDTF_THIS_FY_TO_DATE	This fiscal year to date	DURATION
EVDTF_THIS_MONTH	This month	DURATION
EVDTF_THIS_MONTH_TO_DATE	This month to date	DURATION
EVDTF_THIS_WEEK	This week	DURATION
EVDTF_THIS_WEEK_TO_DATE	This week to date	DURATION
EVDTF_TODAY	Today	DURATION
EVDTF_YESTERDAY	Yesterday	DURATION
EXP_CURRENCY	Expression - Currency	CURRENCY
EXP_DATE	Expression - Date	DATE
EXP_DAY	Expression - Day	DAY
EXP_DECIMAL	Expression - Decimal	DECIMAL
EXP_NUMBER	Expression - Number	NUMBER
EXP_TEXTFIELD	Expression - Text Field	TEXTFIELD

PROMO Field

This field is included within the sign on screen of ExtraView. Use the Title of the field to store HTML, including JavaScript. The HTML is displayed on the Sign On page above the user name and password. You may include the JavaScript within the HTML, or call a user defined JavaScript function. The title field within the data dictionary is limited to 255 characters, so you may need to call a user defined JavaScript function for anything more than a simple entry.

Date & Date Range Fields

Date Fields

The following date fields in the data dictionary provide a means to display information about an issue, calculated from when the issue was initially created.

Field Name	Title	Definition
DAYS_OPEN	Days Open	The number of days that an issue has been in a status that is not closed. The closed status is defined by the setting of the behavior setting named STATUS_CLOSED_NAME. The result is rounded to the nearest day.
WEEKS_OPEN	Weeks Open	The number of weeks that an issue has been in a status that is not closed. The closed status is defined by the setting of the behavior setting named STATUS_CLOSED_NAME. The result is rounded to the nearest week
MONTHS_OPEN	Months Open	The number of days that an issue has been in a status that is not closed. The closed status is defined by the setting of the behavior setting named STATUS_CLOSED_NAME. The result is rounded to the nearest month.
SYSDATE	Current Date	This is a Special Variable Field. When this is encountered within a date field within a query, in the form $$SYSDATE$$, ExtraView will substitute the current date, complete with the current time. This allows users to compose and save queries that have a range to or from the present date.
SYSDAY	Current Date	This is a Special Variable Field. When this is encountered within a date field within a query, in the form $$SYSDAY$$, ExtraView will substitute the current date, without time. The time is set to 00:00. This allows users to compose and save queries that have a range to or from the present date, as of midnight.

Date Range Fields

The date fields within the ExtraView database have an extra field defined in the data dictionary that works in conjunction with the date. This additional field works as a filter on ExtraView report screens, allowing reports to be defined that will select results based on a number of days since the date.

The date fields and the additional field are:

Data dictionary main field	Data dictionary additional field	Comment
DATE_CREATED	DATE_CREATED_SINCE	Days since the issue was created
TIMESTAMP	TIMESTAMP_SINCE	Days since the issue was last updated
DATE_LAST_STATUS_CHANGE	DATE_LAST_STATUS_CHANGE_SINCE	Days since the issue's last change of status
DATE_CLOSED	DATE_CLOSED_SINCE	Days since the issue was closed

User Fields

These fields are additional to the inbuilt user fields of OWNER, ORIGINATOR, CONTACT and ASSIGNED_TO.

Field Name	Title	Definition

| LAST_CHANGE_USER | Changed by | This is the user ID of the user who last updated the issue. |
| USER | *Current User Name* | This field adopts the value of $$USER$$ when used. ExtraView interprets this value as the user ID of the user who is signed on to the current session. |

User Fields with Different Display Format

The display format of user fields is globally set using the behavior setting named USER_LIST_DISPLAY. This is set to either a value of **LIST** or **POPUP** to determine how all user fields on the user input screens will appear.

You can set an Optional Attribute in the data dictionary for any field with a display type of user. This Optional Attribute is also named USER_LIST_DISPLAY. Setting this to either **LIST** or **POPUP** will override the behavior setting, giving you control over the way user's input values into each field with a display type of user.

Note: You should not try to set the attribute of USER_LIST_DISPLAY as a Layout Cell Attribute within the Data Dictionary. This is not supported. You must set the attribute as a global attribute for the field, in the data dictionary.
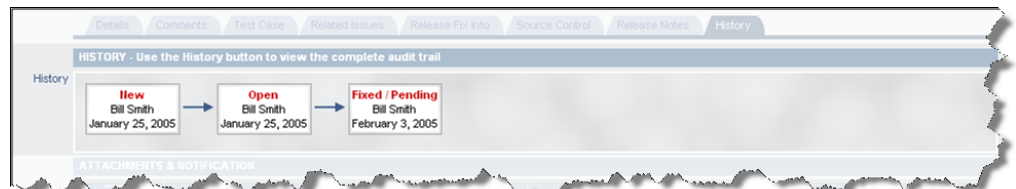
Email Fields

Field Name	Title	Definition
MAILING_LIST	Mailing List	This field is used in conjunction with its security permissions to define which user roles have visibility to the mailing list on the *add* and *edit* screens
CC_EMAIL	CC Email	This field is used in conjunction with its security permissions to determine which user roles have the ability to see and use the CC email list on the *add* and *edit* screens
GENERATE_EMAIL	Generate Email	In conjunction with security permissions, this field is used to control the visibility of the Generate Email checkbox on the *add* and *edit* screens. Without permission, this checkbox will not be visible
CUSTOM_EMAIL	Custom Email	In conjunction with security permissions, this field is used to control the visibility of the Email button on the action bar within the *edit* screen. With permission, the user role will be able to send custom emails
EMAIL_ADDRESS	Email Address	This field may be placed upon layouts. It serves a special purpose. When a user accesses the custom email function from the *edit* screen, to send an ad-hoc email, or an email created from a pre-defined template, this field will be used to automatically populate the email address to which the mail is to be sent. This simplifies communication to users who, for example, enter an email address when reporting an issue. The value stored in this field automatically gives a return address.
EMAIL_SELECTED	Include selected attachments	This field works in conjunction with the attachment field named ATTACH_SELECT. Both fields need to be turned on with permissions for this feature. When enabled, there is an additional checkbox within the notification area. When a user checks this box, and the user has checked one or more file attachments that has the **Select?** checkbox also checked, then the selected attachments will be sent along with the email notification sent when the issue is added or updated.

History Fields

- [Add new comment](#)

Field Name	Title	Definition
ATTACHMENT	Attachments	This field provides visibility of attachment history within the history of an issue. There is an accompanying permission key named PR_RESOLUTION.ATTACHMENT_HISTORY. This field provides a display of a record of which attachments were added or deleted when the update to the issue was made. However, if you are using the behavior setting named ABBREVIATED_HISTORY, then the attachment history is still shown, as part of each history entry

HIST_RANGE_END	Historic data filter	This field is used as a report filter to show the results as of a specific point in time. If this is selected, the results returned by a query are not the results as of the current time. This allows the user to go back in time and determine what issues looked like as of the date/time entered. This is also used within the *evhist* CLI command to determine the end date of transactions being generated
HIST_RANGE_START	Historic data filter (start)	This field is used within the *evhist* CLI command to determine the start date of transactions being generated.
MINI_HISTORY	History	This is a UDF with a Custom display type that can be placed on layouts (typically edit screen layouts and detailed reports. This field generates an area that is about the width of your screen, that displays, by default, the status change dates and the person responsible for the change. If you are creating this field in your own environment, then make certain that you set **Allow selection on reports** to Yes in the data dictionary, and that you provide read access to the field for the roles that are allowed to see the information. There is a behavior setting named MINI_HISTORY_FIELDS that is used to define the names of the fields that are displayed within the mini-history area. By default, these fields are STATUS, TIMESTAMP and ASSIGNED_TO. Not every field may be displayed within the mini-history area. The valid fields are: SEVERITY_LEVEL, PRIORITY, STATUS, PRODUCT_NAME, OWNER, TIMESTAMP, ASSIGNED_TO, PRIVACY, LAST_CHANGE_USER, ALT_ID, AREA, PROJECT, CATEGORY, RESOLUTION, PRODUCT_LINE, DATE_LAST_STATUS_CHANGE, CONTACT, ORIGINATOR, and HIST_TIMESTAMP. Note that the STATUS field is always displayed. You may alter the width and height of the field by setting a width and height (both measured in pixels) into the default value of the field in the data dictionary. Use a delimiter of a semi-colon between the values. For example, a default value of **1000;100** will set the size of the field to 1,000 pixels wide and 100 pixels high. Note that you must be precise with the syntax of the default value as no error checking is performed. If you do not provide a default value, a width of 900 pixels and a height of 80 pixels will be assumed. If the field is placed on a detailed report, then the standard width of the detailed report is used, and the height of the field will automatically expand as required, so that no scrolling takes place. This allows the user to print the detailed report and see all the data. An example is:

The MINI_HISTORY field

NOTIFICATION_HISTORY	Notification History	This field is placed on History layouts to provide a record of the users who were sent notification at each update of the issue. There is an accompanying permission key named PR_RESOLUTION.NOTIFICATION_HISTORY. This field provides a display of a record of the users who were sent a notification upon the update to the issue. However, if you are using the behavior setting named ABBREVIATED_HISTORY, then the record of email norifications are still shown, as part of each history entry
PRODUCT_NAME_HIST	Historical Product Reference	This field refers to the name of a product (PRODUCT_NAME) within an issue's historic audit trail
RELEASE_FOUND_HIST	Historical Release Reference	History is maintained on repeating records within ExtraView. This field refers to that history
RELATIONSHIP_GROUP_TXNS		This field provides visibility of related issue transactions within the history of an issue. There is an accompanying permission key named PR_RESOLUTION.RELATIONSHIP_GROUP_TXNS. This field provides a display of a record of changes to related issues to the current issue. However, if

you are using the behavior setting named ABBREVIATED_HISTORY, then the related issue transactions are still shown, as part of each history entry

| STATUS_HIST | Historical Status | Full history is maintained on all status changes to issues. This field refers to the status history |

Relationship Group Fields

Field Name	Title	Definition
RELATIONSHIP_GROUP_ID	Relationship Group	The internal ID of a relationship group. The only significant change an administrator should make to this field is to alter the title "Relationship Group"
RELATIONSHIP_GROUP_REMOVE_BTN	Remove ?	When this is placed on a related issue layout, it allows the user to remove the issue from the relationship group
RELATIONSHIP_GROUP_OWNER	Relationship Group Owner	This field is used to specify the user ID of the owner of a relationship group. The only significant change an administrator should make to this field is to alter the title "Relationship Group Owner"
RELATIONSHIP_GROUP_TITLE	Relationship Group Title	This field is used to specify the title of a relationship group. The only significant change an administrator should make to this field is to alter the title "Relationship Group Title"
RELATIONSHIP_GROUP_TYPE	Relationship Group Type	This field is used to specify the type of relationship group. This field is for ExtraView internal use only

Built In Repeating Row Fields

Field Name	Title	Definition
RELEASE	Release	The key field used to provide repeating records with a title on screen. It's security permissions control the presence or absence of the repeating record on add and edit screens
RELEASE_ASSIGNED_TO	Release Assigned To	Used to provide a title for Release Assigned To. It's security permissions control access to the value.
RELEASE_FIXED	Release Fixed	Used to provide a title for Release Fixed. It's security permissions control access to the value.
RELEASE_FOUND	Release Found	Used to provide a title for Release Found. Its security permissions control access to the value.
RELEASE_FOUND_HIST	Historical Release Reference	History is maintained on repeating records within ExtraView. This field refers to that history
RELEASE_OWNER	Release Owner	Used to provide a title for Release Owner. It's security permissions control access to the value.
RELEASE_PRIORITY	Release priority	Used to provide a title for Release Priority. It's security permissions control access to the value.
RELEASE_PRODUCT	Release Product	Used to provide a title for Release Product. It's security permissions control access to the value.
RELEASE_RESOLUTION	Release resolution	Used to provide a title for Release Resolution. It's security permissions control access to the value.
RELEASE_SEVERITY	Release Severity	Used to provide a title for Release Severity. It's security permissions control access to the value.
RELEASE_STATUS	Release Status	Used to provide a title for Release Status. It's security permissions control access to the value.

Button Fields

Inbuilt Button Fields

Field Name	Title	Definition
DELETE_BUTTON	Delete Button	The item delete button. Access to the security permissions for this button allow users to delete issues
EDIT_BUTTON	Edit Button	The drill down edit button used on reports and within email
HISTORY_BUTTON	History Button	The button that accesses history from the edit screen or reports
QUICKEDIT_BUTTON	Quickedit Button	The button that accesses the Quickedit mode from column reports and Quicklists
VIEW_BUTTON	View Button	The drill down button that allows you to see the detailed report for an issue
SAVE_BUTTON	Save Button	This button is used on attachment layouts, allowing users to save rather than view attachments

User Defined Button Fields

This allows you to create and place a button with any label of your choosing on an *add* or *edit* screen.

Define a field with a display type of Custom in the data dictionary, with its name begining with the characters **BUTTON_** or ending with the characters **_BTN**. Placing this field on an *add* or *edit* screen layout results in a button being generated on the screen, when you place the field on a layout.

The following are used to define the button:

Data dictionary field	Purpose
Name	This defines the name of the field. For example, a button named BUTTON_GENERATE is valid.
Display Type	Must be **Custom**
Title	Typically this is a space character
Help Text	This will become the text on the button

To provide the action for the button, place a layout cell attribute of type FIELD HTML MODIFIER within the layout upon which you place the button. There are three primary ways you can provide an action for this button:

- If you want to open a new window where you can add a different issue, your FIELD HTML MODIFIER may look like this:

  onclick=window.open("evSignon?
  p_action=doAddDisplay&p_option=Display&p_close_win=true&ev_menu=off&p_area=7&p_project=9")

  The parameters for the Area and Project use their respective IDs. Note that the URL does not need to reside within ExtraView and your action might open up a web address anywhere. See the API Guide for more information.

- Your button might look to execute business rules. As an example, you might want to add a new issue directly into the database, using values from the current issue. To accommodate this functionality, there is an inbuilt JavaScript function that will execute your rules when the button is created. To configure this, follow these steps:
  - The FIELD HTML MODIFIER that acts upon the pressing of the button should be:

    onclick='javascript:;submitChange(this.name)';

  - Create your business rules. This example shows how you might add an issue into a different business area, while copying over some of the field values from the current issue. It is based upon your button field name being BUTTON_CREATE_TEST_CASE:

    if (BUTTON_CREATE_TEST_CASE.{changed}) {
      ADD :

```
                    AREA = 'Test Case Management';
                    PROJECT = 'Test Cases';
                    TEST_CASE_STATUS = 'New';
                    PROCEDURE = TEST_INSTRUCTIONS;
                    MODULE_ID = MODULE_ID;
                    ASSIGNED_TO = '{null}';
                    CREATED_FROM_BUG_ID = ID;
                    COMMENTS = DESCRIPTION;
               }
```

- You can provide an action for the button by implementing custom code within the method named ucRenderEmbeddedObject. The standard distribution of ExtraView that ships with the user custom class named CustomCodeBase.java contains this functionality. In this case, you would write a user custom JavaScript function that submits a URL to the server to execute the server-side code within CustomCodeBase.java.

Notepad Field

A Notepad field is deliberately rendered at the a fixed position within *add* and *edit* screens, to provide a non-scrollable area into which the user may enter notes about the current issue. It is usually rendered at the top of the screen. The Notepad field will remain stationary as the remainder of the form is scrolled within the browser window (or panel within a workspace). This screenshot shows how the Notepad field is first rendered on an *add* screen:

This screenshot shows how the Notepad field remains stationary, when the window is scrolled:

To setup the Notepad field, take these steps:

- On the first row of an *add* or *edit* layout, place the built-in field named EV_NOTEPAD_FOR_LAYOUT. This is a custom field which renders within a fixed area at the top of the layout
- Provide read and write permission for the fields EV_NOTEPAD_FOR_LAYOUT and EV_NOTEPAD for the roles which are used to access the field.
- You may tailor the presentation of the Notepad field with the following layout cell attributes, all placed on the EV_NOTEPAD_FOR_LAYOUT field:

SIZE	This attribute is the width, in characters, of the Notepad field
HEIGHT	This attribute is the number of rows that will be rendered with the Notepad field. A scrollbar appears if you exceed this number of rows
LEGEND:*title*	The default title to the Notes section is the word **Notes**. You can change this by entering an HTML MODIFIER where *title* is the text you want to use to replace the word **Notes**
BGCOLOR:*color*	This is entered as an HTML MODIFIER attribute, and allows you to modify the background border color of the Notepad field on the screen. *color* is a standard HTML color, entered either as a name or as a hexadecimal value
TOP:nn	This is entered as an HTML MODIFIER attribute, and is the number of pixels from the top of the *add* or *edit screen* that the Notepad field is rendered. The default is 30 pixels, allowing room for the menubar buttons. This number is automatically adjusted for the user's text size and for workspaces. It is unusual to need to add this HTML MODIFIER as a layout cell attribute, but it may be used to position the Notepad field at the bottom or side of the screen
BODY_OFFSET:nn	This is entered as an HTML MODIFIER attribute, and is the number of pixels from the top of the *add* or *edit screen* that the first row of the layout following the Notepad field is rendered. The default is 145 pixels, and is automatically adjusted for the user's text size and for workspaces. The most likely need to add this attribute is if you also add a HEIGHT attribute to the Notepad field or you want to eliminate the offset to position the Notepad field at the bottom or side of the screen.

- The EV_NOTEPAD user defined field is used to store the text that the user enters into the field. You can use this a normal user defined field with a display type of text area, but you cannot place it on the same layout as the EV_NOTEPAD_FOR_LAYOUT field
- You can use the EV_NOTEPAD field, which has a default title of **Notepad**, within reports.

Ranking Fields

Ranking fields allow the end user to order a logical collection of issues. The collection of issues is defined by a filter on the ranking field and can therefore contain any subset of issues within your database. For example, you might have a ranking that contains all the issues in a product release within a product, within a specific business area. An end user may then order the issues within the collection, using the ranking field to depict the ranking of an individual issue. Rank fields have the following attributes:

- A rank field is defined in the data dictionary as a field with a display type of number
- You set an attribute on the field to define its use as a rank field. This ensures the field will behave with its special characteristics
- Rank fields can be updated within an *add* or *edit* screen or within a column report. However the ability to manipulate the ranking of issues on a column report is recommended within a workspace, where the user can drag-and-drop an issue to change its position in the rank. Outside a workspace, an end user can use Quickedit to update the ranking of any field, but the underlying report will need to be manually refreshed to see the updated rankings
- Any one issue may belong to different rankings, and have a different position in the ranked order for each of these different ranks. Each of these different rankings is specified with a different ranking field
- Although rank fields may be placed on *add* and *edit* layouts, they are not placed within the audit trail (history) of the issues. The principal reason is that when a user is using a report to rank their issues, they may perform many, many re-rankings within a short space of time. These interim rankings have little value within the audit trail, and if, for example, a user places a top-ranked issue at the bottom of 1,000 issues within the ranking report, 1,000 updates to history would need to be made. This would lead to a very unresponsive and slow user interface.

Creating a Ranking Field

1. Within the data dictionary, create a field with these attributes:
   1. A valid **Business Area**, or * Global Area *
   2. A valid **Project** or the * Master Project *
   3. **Field Belongs** to must be *Issue records*
   4. Any valid **Fixed name**
   5. Any valid **Title**
   6. A **Display type** of *number*
   7. Set **Use as a Rank Field** to *Yes*
   8. Set **Allow selection on reports** to *Yes*
   9. Set **Filter criteria** to *Yes*
   10. Set **Is Sortable** to *Yes*
   11. **Add** the field
2. The new field you created has a **List** button as part of its definition, as shown in this screenshot. This button accesses the screen where you set the specific filters for the ranking field

3. Click the **List** button and then click the **Add** button to add a new entry for the ranking field:
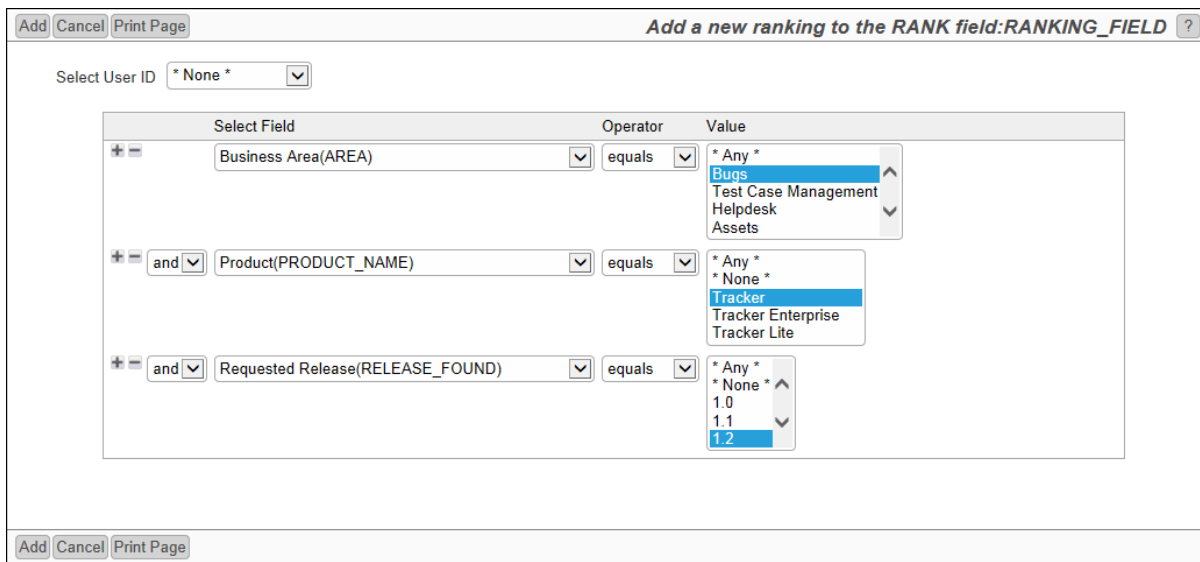
The **Select User ID** can be used to create a ranking for a specific user to use. The filters are used to select the scope of the ranking. Only issues that fall within the collection of issues defined by these filters will be ranked. Only fields with display types that are enumerated lists can be used as filters. For example, list, multi-valued lists, radio button, user and tab fields may be used as filters, but you cannot use numeric type fields

4. **Add** the ranking filters to the database.

Using a Ranking Field

The ranking field can be placed on *add*, *edit* layouts or reports. The use of a ranking field within an *add* or *edit* layout is somewhat limited in that you only have a view of that issues, and not the complete collection of issues within the ranking. Consideration should be made to making the ranking field read-only, through the use of the **read only if** layout cell attribute.

Their primary use is within workspaces, where the user might want to rank a large number of issues at one time. This is documented on the page Ranking Reports.

Timer Field

The function of this field is to create an onscreen timer that is used to measure how much time is spent processing an issue. The timer may be initialized on an *add* or *edit* layout in either the **on** or **off** state using the default value of the TIMERFIELD. Assuming the timer is **on**, then time is accumulated in the field with each successive edit session. The user will see the timer updating each second that it is turned on.
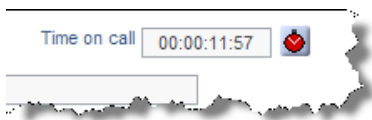
Timer field on a layout

At this time, only one timer field is supported in an installation within a business area. To configure the timer, you must have the following fields defined in the data dictionary:

Field	Configuration
TIMERFIELD	Display type – Custom Allow selection on reports – No Default value – On or Off Help text – The time spent working on this issue This field may be placed on an *add* or an *edit* layout and presents the timer to the user, with a button that is used to turn the timer on and off. Being a custom field, it is not possible to report on this field, therefore a second field is defined which is used to provide reporting.
TIME_ON_CALL	Display type – Text Field Allow selection on reports – Yes Filter Criteria – Yes Is Sortable - Yes Help text – The time spent working on this issue This field is maintained by ExtraView custom code and takes its value from the TIMERFIELD when an issue is inserted or updated.

Tag Cloud Fields

A **Tag Cloud** field is a graphical representation of the values in a list field. The values in the **Tag Cloud** field are clickable to allow the user to select or de-select them; selected values are highlighted with the color configured in the HIGHLIGHT_COLOR behavior setting. The **Tag Cloud** field is configured in the following manner:

- A UDF with a display type of List. This is termed the **source list field**. This contains the list of values in the tag cloud
- A UDF with a display type of Custom. The name must start with the same name as the **source list field** and end in **_TAG_CLOUD**
- Optionally, the Default value field in the Data Dictionary entry for the custom field can be configured with the following parameters for rendering the field:
  - height=NNN (default 100)
  - width=NNN (default 900)
  - columns=NNN (default 10)

Multiple parameters must be separated by commas. E.g, **height=90,width=800,columns=7** overrides all the default values. You need not spcify all the all parameters. E.g, **columns=5** renders list values in five columns with the default height and width.

On the *Add* or *Edit Screen* layouts, both the **source list field** and the **Tag Cloud** fields must be included, where the **source list field** must precede the **Tag Cloud** field in order on the layout, i.e, the Tag Cloud field must be on a row and column higher than the **source list field**. The **Tag Cloud** field includes support only for the REMOVE_NONE layout cell attribute, which removes the ***None*** value from the list of values rendered; all other layout cell attributes configured on the **Tag Cloud** field are ignored.

Layout Editor Tip: In order to hide the source List field and make only the Tag Cloud field visible to users, configure the source List field with the following layout cell attributes:

- FIELD ALTERNATIVE TITLE = " " (a blank space; this will replace the original title of the source List field)
- FIELD STYLE = display:none (this will cause the values drop-down box to be hidden)

This will cause the **source list field** to be rendered on the screen in a hidden mode, but still be accessible to internal routines that are used to communicate between the source list and Tag Cloud fields. Other layout cell attributes configured on the source List field, such as FIELD REQUIRED, will continue to be enforced. Note that using the FIELD/LAYOUT VISIBLE IF layout cell attribute to hide the source List field will disable the Tag Cloud field functionality.

The Security Permissions configured for the Tag Field affect how the values are rendered: if readonly, then values are displayed as plain text without hyperlinks; if read/write, then values are displayed as hyperlinked text.

Behavior Settings

Behavior settings control many aspects of ExtraView behavior. Behavior settings tend to be set up once with your installation and then retain that value for ExtraView behave the correct way for your purpose.

The behavior settings are grouped together in like categories, and each category forms one of the following pages. All settings can be modified by either clicking on the *Value* field, or by clicking the *Edit* button at the left of the field name.

The Behavior Setting utility offers the ability to filter the settings list by selecting which column on which you want to perform the filtering operation, then clicking on a letter. The letter is the first character of the selected filter column. You may alternatively enter a search expression and use the **Go** button to search for the results. Wildcards are allowed in the search.

Note the buttons that represent each category. You can use any button to filter the page so that you only see that category of setting.

API Settings

This section of behavior settings deals with API settings. The available settings are:

System Controls menu- API Settings	Typical Value	Description
ALLOW_ANONYMOUS_API_ACCESS	NO	When this is set to YES, users can make calls to the API, without having a valid user name and password. Typically used when you are integrating ExtraView within your own website, and you do not want to give your external users (usually your customers) a user ID within ExtraView. Use

this in conjunction with ANONYMOUS_API_USER_ID. Valid values are YES and NO

ANONYMOUS_API_USER_ID	guest	If you have ALLOW_ANONYMOUS_API_ACCESS set to YES, this is the user ID that is set within issues as the ORIGINATOR
ALLOW_CLI_UPDATE_ORIGINATOR	YES	YES will allow a user to update the ORIGINATOR field through the CLI
CLI_EDIT_MULTI_VALUE_FIELDS	NO	Indicates whether you can edit multiple value UDFs through the CLI
DEFAULT_TEXT_REPORT_DELIMITER		Single character to place between data fields on text reports. When you output results to a text file, or through the API / CLI, this character is used to delimit the individual fields
MAX_REPORT_TEMP_FILE_HOURS	2880	This is the number of hours for which temporary report files used by the API will be kept on the file system of the server. Following this number of hours, the temporary files are removed by ExtraView.
MULTI_RELEASE_XML	NO	To enable multiple release to be output to XML through the API. Valid values are YES and NO

Company Information Settings

This section of behavior settings deals with Company Information settings. The available settings are:

System Control menu- Company Information Settings	Typical Value	Description
COMPANY_NAME		The company's name
COMPANY_ADDRESS1		The company's address
COMPANY_ADDRESS2		Second line of your company's address
COMPANY_CITY		The company's city
COMPANY_STATE		The company's state
COMPANY_ZIP		The company's postal or zip code
COMPANY_PHONE		The company's phone number
COMPANY_EMAIL		The company's general email address
CONFIDENTIALITY_MESSAGE		This setting allows the administrator to create a system-wide confidentiality statement that appears at the bottom of most screens and reports. This statement may be up to 4,000 characters in length and may contain valid HTML
LICENSE_INSTALL_EXPIRATION_DAT		This setting defines the date after which the license key may no longer be installed
LICENSE_UPGRADE_VERSION		This setting contains the VVV.SSS version of code beyond which the site cannot be upgraded by the administrator, unless a new activation key is obtained from ExtraView
LIMITED_USER_UPDATABLE_FIELDS	CUSTOMER_COMMENTS, ATTACHMENT, PRIORITY	This setting contains the comma-separated field names of fields that may be modified by a limited user

Display Settings

This section of behavior settings deals with Display settings. The available settings are:

Workflow menu- Display Settings	Typical Value	Description

ADMIN_LIST_SIZE	1000	This number will determine how many records will be displayed on any administrative list. If the number of rows to be displayed exceeds this behavior setting value, then the records are not immediately displayed. Instead, there will be a drilldown capability, with the administrator being able to search the list, or to use the first character of the entry they are searching for to see a limited number of entries. Note that if a negative number is used, then the functionality remains the same, except the search capability is compressed on the screen, to a single line
ADMIN_LIST_MIN	50	Below this number, any admiistration list will not display the button which allows the set of filters to be displayed. This prevents the display of a button on moderate-sized lists where no filtering is typically needed
ALLOW_HELP	YES	When this is set to YES, tooltip help is displayed when placing the mouse over a field label on the add and edit screen and clicking on the field label will result in a pop-up window with help on the field. When the value is set to NO, no tooltip or pop-up window is generated
AUTO_SCROLL_TO_ CLICKED_LAYOUT	NO	If this value is set to YES, then when a user clicks on a field within an embedded layout which causes a screen refresh, the screen is automatically scrolled to the top of the embedded layout. This can be useful if you have relatively long add and edit screens, as it takes the user to the same area of the screen on which they were working, as opposed to seeing the top of the form on each refresh. If this value is NO, the screen refresh leaves the user at the top of the form.
BG_ALT_COLOR	#DEF0F8	Alternative background color for tables. Used as a complementary color to BG_COLOR
BG_COLOR	#dddddd	Background color for tables. Used as a complementary color to BG_ALT_COLOR
BORDER_COLOR	#C7C9C7	Border colors on the Search / Report page
CACHE_AREA_PROJECT	YES	Specify YES to allow the caching of area/project dropdown lists in templates built internally from layouts. Specify NO to force a dynamic refresh of the area/project list for each screen refresh in *add* or *edit* mode. This should always be YES unless there is some dynamic list modification by USER_CUSTOM code.
CALENDAR_STYLE	Fancyblue	Use this setting to set the color and style of the popup calendar used to select dates. Valid entries are aqua, bluexp, fancyblue, forest, green, greengrass, maroon, win2k, winter, winxp, wood and yellow
CLICK_LOCKDOWN_ TIMEOUT_SECS	0	The number of seconds to wait after an *add* or *edit* screen begins loading to unlock the page for mouse clicks and input. The minimum, and the default setting is that ExtraView will wait for up to 20 seconds for an Ajax server response. This option should be set higher if you have a number of users who have very low bandwidth access to ExtraView and as a result, the *add* and *edit* screens take a very long time to load. This setting prevents the user seeing errors if they attempt to set a value in a field which causes a screen refresh, but the initial screen has yet to complete loading.
DATE_FORMAT	DD-MON-YYYY hh24:mi	Default format for displaying date fields (do not include seconds)

DB_TIMEZONE	PST	The time zone of the database server and the reference from which all local times are calculated for each user. This value should be set on the initial installation of ExtraView, and it should not be altered from that time. Altering this behavior setting to a different time zone will cause all timestamps in the system to be displayed with a different value than with which they were created. Normally, this time zone should be the same time zone of the system clock of the server on which the installation resides. Note that you must restart the ExtraView server for this setting to take effect
DEFAULT_DATE_FORMAT	MEDIUMDATE	The system default date format for presenting user's data. This may be one of the following, with the result of the example shown:

SHORTDATE	11/28/03
MEDIUMDATE	Nov 28, 2003
LONGDATE	November 28, 2003
FULLDATE	Friday, November 28, 2003
SHORTDATETIME	11/28/03 7:20 AM
MEDIUMDATETIME	Nov 28, 2003 7:20:08 AM
LONGDATETIME	November 28, 2003 7:20:08 AM WST
FULLDATETIME	Friday, November 28, 2003 7:20:08 AM WST

In addition, this may be set to a custom format. For example, you might set DEFAULT_DATE_FORMAT to a value such as *yyyy-MM-dd hh:mm:ss*, in which case this becomes the default date format used to present dates to the user. See the Appendix on date formats for an explanation of valid formats.

DEFAULT_CHART_FONT	Arial, Helvetica, sans-serif	The font used for the body of almost all pages. It is suggested that you provide three font names, in the order of preference, as different browsers, on different platforms may only have access to some of the fonts in the list
DEFAULT_PDF_FONT	Arial	The default font for selection as the text font for PDF's. It is essential that this font exists on the server and contains all the characters you are likely to display in Adobe PDF documents. This is particularly important if you are working in a double-byte language such as Japanese.
DISPLAY_SIGNON_PROD_INFO	YES	If this is set to a value of YES, ExtraView will display marketing information sourced directly from ExtraView's servers
DISPLAY_SIGNON_URL_1	http://prodinfo.extraview.com/site/content/product-information-1	This is the URL of the right-hand frame sent to the sign on page if DISPLAY_SIGNON_PROD_INFO is set to YES. Care should be taken to ensure you enter a valid absolute URL into this field
DISPLAY_SIGNON_URL_2	http://prodinfo.extraview.com/site/content/product-information-2	This is the URL of the left-hand frame sent to the sign on page if DISPLAY_SIGNON_PROD_INFO is set to YES. Care should be taken to ensure you enter a valid absolute URL into this field
EDITOR_TOOLBAR	STANDARD	This sets the toolbar for the HTML editor used in

		HTML Area fields and other screens. The possible values for this setting are BASIC, STANDARD, FULL or CUSTOM. CUSTOM toolbars must be supported with JavaScript code in the UserJavaScript.js file
ENABLE_SPELLCHECK	YES	This setting enables and disables ExtraView's internal spellchecking software. Many browsers offer a spell-checking feature, but this may be used in place of these. See the Appendix entitled Spell Checker for more details.
FIXED_WIDTH_FONT	'Lucida Console', Courier, monospace	Font used for the display of Print Text user defined fields and used for read-only description fields and old log area fields, if REPORT_WITH_FIXED_WIDTH_FONT is YES
FOLD_TEXT_POSITION	100	This specifies the character position at which to fold text lines in read-only fields with a display type of text area, print text and log area. You should not specify a number lower than 65. If you specify a high number, such as 99999, the text input will never be folded. Note that FOLD_WORD_POSITION has precedence over this setting
FOLD_WORD_POSITION	100	This specifies the character position at which to break up a long word in fields of Text Area, Print Text, and Log Area fields. You should not specify a number lower than 65. If you specify a high number, such as 99999, long words will never be broken up. This can be used in conjunction with FOLD_TEXT_POSITION. Set FOLD_WORD_POSITION to a high number if you make routine use of long URL's in your text fields. This will ensure the URL's are not broken onto more than one line, and that the user can click on the URL with success
HIDE_MOBILE_SIGNON_SCREEN	NO	This setting may have the value of YES, NO or SIGNON_BUTTON. When the value is NO, the User ID and Password fields are always visible. When the setting is YES, the input fields for the User ID and Password are never viewable and the client is reliant on one or more of the behavior settings named MOBILE_DIRECT_ACCESS_1 through MOBILE_DIRECT_ACCESS_5 providing direct access to screens within ExtraView. When the value is SIGNON_BUTTON, there is a button with the title **Sign On** and when this is clicked, the User ID and Password fields become visible.
HIGHLIGHT_COLOR	#FF0000	Highlight color for a table cell when you use a layout element attribute of FIELD HIGHLIGHT IF
HIGHLIGHT_COLOR_ADD	#FF0000	The color to indicate an added value between updates of a record, in history and email notifications
HIGHLIGHT_COLOR_DELETE	#CCCCC	The color to indicate a value deleted in an update of a record, in history and email notifications
HIGHLIGHT_COLOR_UPDATE	#ff0000	The color to indicate an updated value, in history and email notifications
HIGHLIGHT_TIMESTAMP	YES	If set to YES, the TIMESTAMP field will always be highlighted on email and history report. Valid values are YES and NO

HIGHLIGHT_VALUE_STYLE		A CSS style that is applied to the value of a field on an add or an edit layout when the layout cell attribute named FIELD VALUE TAG is applied to the cell.
HTMLAREA_ROW_HEIGHT	10	The number of rows to display on the *add* or *edit* screen for fields with a display type of HTML Area.
IMG_HOME		This is a relative path within the locales directory for your language where all the buttons and images used within the main application frame reside. For English this is normally locales/en_US. If you want to create your own image set, copy an existing directory to a new folder in the path locales/en_US/images, e.g. locales/en_US/images/my_images. Then set this behavior setting to a value of ../images/my_images/
HIGHLIGHT_VALUE_STYLE		A CSS style that is applied to the value of a field on an *add* or an *edit* layout when the layout cell attribute named FIELD VALUE TAG is applied to the cell
LABEL_COLOR	#0000FF	Color of field labels on the *add, edit* and *query* screens
LABEL_WRAP_POSITION	15	Character position after which to wrap label text in the *add* and *edit* issue screens and the search screen. Note that if a layout element attribute for a field of FIELD ALTERNATIVE TITLE is set, then this is ignored and the administrator will set his own formatting in the field
LOG_AREA_BREAK_LINE	NO	If this value is YES and ExtraView is rendering the LOG_AREA field as HTML rather than text, then text up until a or a </p> tag will be rendered, followed by a link with the word "more". Upon pressing the "more" link, the entire field will be rendered. Note that this can override the functionality offered by the setting named LOG_AREA_DISPLAY_CHARS. If you want this setting to override LOG_AREA_DISPLAY_CHARS, then set that value to 32000
LOG_AREA_DISPLAY_CHARS	250	Maximum character length of log area fields before they are truncated on the *edit* issue screens. Once truncated, the word "More" appears and the user can drill down on this word to see the remainder of the text. Do not set to less than 80 or more than 32000
LOG_AREA_INITIAL_SORT	DESC	This setting controls the initial order of the log area entries. Valid values are ASC for ascending times and DESC for descending times
LOG_AREA_TEMPLATE	$$LA_DATE$$ $$LA_USER$$	This setting controls the appearance of the header on fields with a display type of Log Area. You may use any text within the header, plus the following variables: • $$LA_DATE$$ - the date the field was updated • $$LA_USER$$ - the name of the user performing the update • $$LA_COMPANY$$ - the company name of the user making the update • $$LA_SEQNUM$$ - a sequence number for the comment within an issue, starting with 1
MANDATORY_FIELD_POST		HTML tag or characters to place after mandatory field labels. The default is to use the HTML end bold tag, but you can substitute and valid HTML or characters
MANDATORY_FIELD_PRE	<b>	HTML tag or characters to place before mandatory fields labels. The default is to use the HTML bold tag, but you can substitute any valid HTML or characters

MAX_IMAGE_DIMENSION_PIXELS	4000	This setting defines the maximum number of pixels in either the horizontal or vertical direction that an image may be, to load it into a field with a display type of Image.
MENU_DIRECTION	VERTICAL	The main navigation bar on the screen can be HORIZONTAL or VERTICAL. Note that you must sign off from ExtraView and sign on again for this setting to take effect
MENU_BUTTON_POSITION	LEFT	This setting allows the swapping of the screen title and the menu buttons at the top and bottom of each screen. If the setting is LEFT then the buttons appear at the left of the screen.
MENU_SIZE	105	Width or height of the navigation bar, in pixels, according to whether MENU_DIRECTION is VERTICAL or HORIZONTAL. Note that you must sign off from ExtraView and sign on again for this setting to take effect
MOBILE_DIRECT_ACCESS_1 to MOBILE_DIRECT_ACCESS_5		This setting defines the title for a button, and a URL within the ExtraView installation to which the user is directed when they press the button. The syntax of the entry in the setting is button_title:URL. When this is defined, a button will be displayed on the sign on screen of the mobile client and when this is clicked, the user will next see the screen defined by the URL
MORE_HISTORY_NUMBER_ROWS	25	Defines the number of rows to be shown before the MORE... message in a related issue display on a history report
MOUSEOVER_COLOR	#E0E8F3	The color of rows on reports when the user places their mouse over the row
MULTI_VALUE_HIGHLIGHT_CHAR	▶	This value is the character that will be used to highlight the selected values in a multi-valued UDF list field. If the value is left blank it will display a + next to selected values. The value ▶ will display as a ▶ character. Use a single character, or a string that your browser will interpret as a single Unicode character. If this character does not display in your user's browsers, select an ASCII character such as +
NAV_BAR_DRILLDOWN_ BOX_STYLE	position:relative; left:-30px	The CSS style to be applied to the table containing the drilldown box on the navigation bar. This is only used when the MENU_DIRECTION is set to HORIZONTAL with horizontal style navigation bars and is used to alter the position of the drilldown box for different styles of navigation bar. Most frequently, you can position the drilldown box in an absolute position on the navigation bar, but you can also use effects such as altering the background-color to change the presentation of the control.
NAV_BAR_GO_BUTTON	NO	This places a Go button onto the navigation bar by the drilldown box. Valid values are YES and NO. After changing this setting, you must sign off from ExtraView and sign on again for this setting to take effect.
NAV_BAR_LOGO_STYLE	padding:20px 0px 0px 25px	This style is used to position the CompanyLogo.gif on the navigation bar, when the MENU_DIRECTION is set to HORIZONTAL
NAV_BAR_STYLE	text-align:right	This allows you to apply a CSS style to the navigation bar buttons. For example you can use the style to left or right align all the buttons on the navigation bar

POPUP_LIST_SIZE	100	The number of entries on a pop-up list, before the list is accessed through a list of the characters A .. Z as opposed to a list of the entries themselves
POPUP_WIN_STYLE	silverxp	This setting provides different styles for the decoration of popup windows. You may select "default", "minimal", "osx", "plain", "silverxp" or "winxp" as valid styles. The value must be in lower case.
REFRESH_LIST_MAX_SIZE	200	A fast refresh using JavaScript will occur in lists that are less than this size and the refresh JavaScript option has been selected for an allowed value relationship. This setting allows the administrator to make a trade off for their users, as to the time to load metadata into a browser and perform a fast refresh when selections are made, compared to the time it takes to refresh the metadata lists from the server
SHOW_FOOTER	YES	If the value of this setting is YES, then the footer with action buttons is displayed on all screens. If the value is NO, the footer is suppressed.
REPORT_HEADER_ FOOTER_COLOR	#ADBFD0	This is the background color of the headers and footers on reports
REPORT_SUMMARY_ TOTAL_COLOR	#ADBFD0	The background color of the total rows on summary reports
SCREEN_HEADER_ FOOTER_BG_COLOR	#E1EFEE	The background color of the screen titles and footers
SHOW_PRINT_BUTTON	YES	If this setting has a value of YES then a button allowing the user to print the current screen will be displayed on all menubars.
SIGN_ON_SCREEN_LOGO	../images/CompanyLogo.gif	This is the path to the logo placed on the sign on screen. This setting allows you to use a different logo on the sign on screen than is used on the navigation bar. You can point this image to any file your server can access
SIGN_ON_SCREEN_LOGO_STYLE	position: absolute; left:10px;top:10px	The style to be applied to the sign on screen logo. The default position is the top left-hand corner of the screen
SUPPORT_LINK	\ExtraView Support Site\	This HTML statement is used for the link at the end of the copyright statement on each screen. Normally used to direct your users to a specific place for support. You should use the JavaScript function *window.open* to provide a link, as opposed to using the HTML attribute *href*. This provides a more general method to link to URLs that are outside of the domain where your ExtraView server is installed. The example value shows how to open a web page. You can also open a new email by using the parameter mailto:xxx@yyy.com?subject= Please%20provide%20assistance.
TABS_PER_ROW	10	Limits the number of tabs on a form such as the *add* or *edit* screen, in a single row when being displayed on the add or edit issue screens. If more values than this number need to be displayed, they are placed on a separate row
TAB_FONT_OFF_COLOR	#444444	Color of the font on non-selected tabs that are generated with the tab display type and within the administration area
TAB_FONT_ON_COLOR	#FFFFFF	Color of the font on the selected tab. This is used for

		fields that have a display type of tab and within the administration area
TAB_OFF_COLOR	CCCCFF	Off color is the unselected tabs color that are generated with the tab display type and within the administration area
TAB_ON_COLOR	6666FF	On color is the selected tabs color. This is used for fields that have a display type of tab and within the administration area
TAB_SEPARATED_EXPORT	NO	When you are exporting information from the administration list utilities, the default is that the data will be exported in CSV (comma separated value) format. Most browsers will place this information in a Microsoft Excel spreadsheet by default. There are some occasions when you may want to export the list information in TSV (tab separated value) format, and setting this value to YES achieves this. If you work predominantly with double-byte character sets (e.g. you may be working in Japanese) TSV may be the preferable format. Note that you can often configure your browser to work with TSV files. Consult the documentation for your browser on how to achieve this if you require this functionality.
TEXTAREA_ROW_HEIGHT	4	The number of initial rows of data to display on the *add* and *edit* screens for fields of display type Text Area, Print Text, Log Area
THUMBNAIL_MAX_SIZE	150	A behavior setting named THUMBNAIL_MAX_SIZE controls the size of the thumbnail. This is the maximum number of pixels in either the horizontal or vertical direction to which thumbnail images will be generated. The aspect ratio of the original image will be retained. Thumbnail images are generated for file attachments and for fields with a display type of image. If you change this value, existing thumbnail images will remain at their original size and new thumbnails will be generated at the changed value.
USE_ALLOWED_VALUE_ SORT_ORDER	NO	If set to YES, the child allowed values list will be sorted by the sort_seq in the allowed values table instead of the meta data sort_seq
VALUE_TAG_DEFAULT		This provides a point where a default style for all value fields on add and edit screens may be inserted. The insert is an HTML attribute that sits within the table cell that holds the value for fields on these screens. For example, you may insert a CSS style that is applied to all the values on the form. You may override the default in this setting on a field by field basis with the layout element attribute FIELD VALUE TAG. The default value in a new instance of ExtraView is that this setting has no value.
WINDOW_BG_COLOR	#ffffff	Window background color

Email Settings

This section of behavior settings deals with Email settings. The available settings are:

Email Notification Menu- Email Settings	Typical Value	Description
AD_HOC_EMAIL_FROM_ADDRESS		Return address for all emails sent using templates or ad hoc text entry. These are emails sent by using the Email button on the *edit* screen
AD_HOC_EMAIL_FROM_SENDER		The User ID of the sender of all template and text emails sent from the Email button on the *edit* screen. The user's email address will be taken from his account information
CHECK_EMAIL_ADDRESS_FORMAT	YES	If you set this value to YES, then the email address entered on the user accounts screen will be checked to ensure it has a valid format with a fully qualified domain. If set to NO, the format of the email address is not checked
CONTACT_ADMINISTRATOR	NO	When this setting is YES, a prompt appears on the sign on page, with a default message of "Forgotten Password?". When this link is pressed, a mailto the user defined in the setting named EMAIL_ADMINISTRATOR_USER_ID is started, allowing the user to send a message. When the setting is NO, the prompt does not appear. Note that this link will only appear if the site has EMAIL_NOTIFICATION set to YES, and a valid directory set for EMAIL_DIRECTORY
EMAIL_ADMINISTRATOR_NAME	ExtraView Administrator	This is the title to the email address for the ExtraView administrator. Emails that are automatically generated by ExtraView are originated with this name. Examples are emails sent upon the self registration of a user, or an unauthorized access attempt. This setting is used alongside the EMAIL_ADMINISTRATOR_USER_ID
EMAIL_ADMINISTRATOR_USER_ID	admin@my_company.com	This is the email address from which emails originating within ExtraView are sent. This is usually the administrator's email address or an alias for the administrator. This setting is used alongside the EMAIL_ADMINISTRATOR_NAME
EMAIL_ALLOW_UNQUAL_ADDRESSES	NO	If your company's email server allows unqualified email addresses (i.e. email addresses without the @company.com part of the address), then set this to YES and users will not need to use the complete email address.
EMAIL_BCC_ARCHIVE		The email address where a copy of template and ad hoc emails are sent as Blind Carbon Copy (bcc)
EMAIL_CHARSET	UTF-8	The default character set that is used by each user when email notification is sent to them by the ExtraView server
EMAIL_CONTACT_ADMINISTRATOR	admin@my_company.com	The email address for the administrator who will be contacted via the sign on page when a user clicks the link "Forgotten Password?"
EMAIL_CUSTOMER_BOX	UNCHECKED	This setting can have a value of CHECKED or UNCHECKED. When this is CHECKED, the checkbox on the *add* and *edit* screens that controls whether the LIMITED_USER_ROLE users (usually customers) is checked by default.
EMAIL_DIRECTORY	../mailbox	Email Directory for outgoing messages to be stored. This is the location on your server where the batchmail process looks for outgoing emails. You

may use an absolute path, or a relative path. Relative paths are relative to the WEB-INF directory

EMAIL_FROM_USER_ID	support@yourcompany.dom	Return address for all email originating in ExtraView. This allows the recipients of emails to reply to the email, and know there is a valid destination
EMAIL_FROM_USER_NAME		Alias for the real user name that email originates from. This is inserted into the header information of the outgoing notification
EMAIL_MODULE_OWNER_ALWAYS	YES	Email module owner irrespective of whether they are assigned to an issue. Valid values are YES and NO
EMAIL_NOTIFICATION	YES	Turn email notification of changes on and off. Valid values are YES and NO. This is the master control for enabling and disabling email. When this is set to NO, email addresses are not required when creating or editing users. When the value is set to NO, no email is sent and no warnings are generated to users.
EMAIL_NOTIFY_USERS_ALWAYS	YES	This behavior setting controls whether ExtraView will always send a notification to a user when an issue is updated, or whether notification is only sent when one or more fields on the layout that is used to communicate with the user are changed. Set this value to YES if you always want users to receive notification. Note that a change does not mean any change to the issue being updated, but means a change to a field that is on their email layout must occur. Also, note that if more than one update to a record occurs within a minute, and the user's timestamp does not include seconds, then they will only receive the first update made within a minute. The user can set his personal date format to include seconds if he wants to be certain that he will receive multiple updates within a single minute.
EMAIL_STYLESHEET	* {font-family: Arial,Helvetica,sans-serif; font-size:10pt} .text {font-size : 10pt} .report_text {font-size : 10pt}	The style for HTML email. This is included within the body of the email notification sent, so that the user does not need access to a server when reading their email. An entry in EMAIL_STYLESHEET is pure CSS and it gets embedded into an HTML <style>...</style> tag, so the user does not need to be online when they receive the email. You can use any valid CSS in here, either as the global style or to embellish the text or report_text styles.
EMAIL_SUBJECT_TEMPLATE	ExtraView Notification [$$ID$$]: $$STATUS$$ - $$SHORT_DESCR$$	Format for subject line of emails. The tokens between the $$ signs will be replaced with the actual value from the current record
EMAIL_TABLE_WIDTH	660	This setting controls the width of the output within email clients, for standard system generated notifications. The normal value is 660. If you want to generate standard notifications that are not fixed in width, set a value of 0.
EVMAIL_DELIMITER_TEXT		When generating outbound email in HTML format, ExtraView surrounds the body of the email with this text within invisible tags. If a user replies to this mail notification, thus generating an update to the issue, the evmail utility will suppress all the text between the two tags, thus eliminating the outgoing email from being added into the issue as part of the update
GENERATE_EMAIL_BOX	CHECKED	The default value for the Generate Email Box on the *add* and *edit* screens. Valid values are CHECKED

and UNCHECKED

MAX_SIZE_EMAIL_ATTACHMENTS	0	This is the maximum size, in bytes, of attachments that are sent out with standard email notifications. If the size of attachments exceeds this number of bytes, then the issue is not updated, the notification is not sent and the user will receive a warning. The user may remove attachments from the notification using the **Select?** checkbox, and then attempt to update the issue again. If the value is 0, the size of attachments is not sent. This setting is useful if you are working within an email environment which has a low setting for processing the size of emails and rejects emails over a specific size.
SET_EMAIL_ENCRYPTION	NO	Show encryption option in ad hoc email screen. Valid values are YES and NO
SUPPRESS_STANDARD_EMAIL_LIST	NO	If this is set to a value of YES, then no email notification is made by using ExtraView's inbuilt logic. Instead rules defined in the Email Rules script are used to generate notifications to users. An exception to this is that email generated using email templates will still send email to the users attached to the template.

Environment Settings

This section of behavior settings deals with Environment settings. The available settings are:

Advanced menu- Environment Settings	Typical Value	Description
ALLOW_DEBUG_URL	YES	Allows user to set up debug level of the application server log. Valid values are YES and NO. If turned off, then the user cannot alter the debug level of the log with a URL. The form of the URL is: http://server.extraview_domain.com/evj/ExtraView? DEBUG=6 The default level of debug messages is 6. Valid values are in the range 1 to 12
ATTACHMENT_PERMANENT_DELETE	YES	When this setting is set to YES, any file attachment is permanently deleted from the server when the user deletes an attachment. When the setting is NO, the attachment is retained on the server, but is not visible on add / edit screens or on reports. However, the attachment may be viewed from the Attachment History screen for audit trail purposes.
ATTACHMENT_REPOSITORY_DMAX	999	The maximum number of files or directories that will be created under one node of the external directory structure. The default for this value is 999. It is not likely that this value needs to be altered.
ATTACHMENT_REPOSITORY_OPT	INTERNAL	This setting controls whether attachments are stored internally within the database, externally on the file system, or in some combination of the two methods. If the value of this setting is INTERNAL, then all attachments are stored internally within the database. If the value of this setting is EXTERNAL, then all the attachments are stored on the external file system. Alternatively, you may provide a comma separated list of file

extensions and then all files with these extensions will be stored externally while all other files will be stored in the database. For example, if you set the value of this setting to "avi, png, gif, jpg" then files of these types will be stored externally. This strategy leaves the files stored internally in the database as searchable by keywords, while the image and video files are stored externally. The default for this value is INTERNAL. Also, ensure that the setting ATTACHMENT_REPOSITORY_ROOT is set correctly before storing attachments outside the database.

ATTACHMENT_REPOSITORY_ROOT	The name of the directory on the file system where file attachments will be stored. Before any attachments are stored externally, you must also provide a valid setting for the setting ATTACHMENT_REPOSITORY_OPT. You must also ensure that the path is valid from the application server(s) that are running ExtraView, and that you have all the permissions to read and write to the storage. You must also ensure that you set up a separate backup method for this external storage as backing up your database will no longer backup the file attachments.
BLOCK_ATTACHMENT_EXTENSIONS	If a user uploads an attachment, or a file into a document field or a file into an image field, with an extension that is in this list, the file will not be stored on the server, and the user will receive a message that informs them that they are not allowed to store files of that type. The default is that all files of all types are stored. The list of file extensions is comma delimited

CACHE_ENABLE_FIX_FOR_IE	NO	There is a bug in Microsoft Internet Explorer which prevents documents being uploaded to a server where there is non-secure back-end server instance in a network with secure proxies. To counteract this, set the value of this behavior setting to YES when you have a network configured in this manner. This is highly unusual and most instances should not need to reset the default value of NO for this setting.
CSS_HOME		Only used if you want to override the inbuilt CSS stylesheets for the given image set you access, specified in the behavior setting named IMG_HOME. This is normally not used, as ExtraView will look for the CSS stylesheets as a directory named stylesheets/ within the IMG_HOME path and this is the best location for your stylesheets. If you provide a path for this setting, then you must provide three stylesheets in the directory, named small.css, medium.css and large.css
DEFAULT_ATTACHMENT_CHARSET	UTF-8	The default character encoding for files being uploaded to ExtraView. This value is used to select the initial value presented to the administrator when creating a new user.
DEFAULT_LANGUAGE	en	Default language for the installation. This is typically EN for English
DEFAULT_REGION	us	Default region for the installation. This is typically US for the United States
DEFAULT_VARIANT		Default variant for the installation. This is not typically used
DOMAIN		Cookie domain
ENABLE_AREAS	YES	This setting controls whether the installation works with multiple business areas or a single business area. Values can be YES or NO
ENABLE_PROJECTS	YES	This setting controls whether you can use multiple projects within each business area. Valid values are YES and NO. Note than if you use YES, then ENABLE_AREAS must also be set to YES

ENABLE_QUICKFIND	NO	This setting controls whether QuickFind text search is used for keyword searches. Valid values are YES and NO. Note that if you use YES the database must have text indexes created to perform the searches.
HELP_HOME		The default for this is a null entry, in which case ExtraView points to ../WEB-INF/locales/en_US/help for the default locale. Localized help can be placed underneath other locales/xx_XX directories, in which case ExtraView will find these automatically, assuming the locale has been created. HELP_HOME can also be set to an absolute URL. If the value begins with "http://", "https://" or "file://", then the value is assumed to be a url pointing to a resource below which is a "help" directory for the help files.
HTTP_CHARSET	UTF-8	The name of the default character set used within the user's browser. It is recommended that this be "UTF-8"
LOCALIZE_TITLES	NO	Used to turn off and on the localization buttons within administration. This is used when you are using ExtraView's multiple languages on the user interface. When this option is set to YES, a button with the title of "Localize" will appear beside all metadata titles and values that can be localized into different languages. Valid values are YES and NO
LUCENE_INDEX_LOCATION		The name of the directory on the file system where Lucene full text indexes will be stored. Before any indexes are built, you must also enable full text searching by setting ENABLE_QUICKFIND. You must also ensure that the path is valid from the application server(s) that are running ExtraView, and that you have all the permissions to read and write to the storage. You must also ensure that you set up a separate backup method for this external storage as backing up your database will not backup the full text indexes
MAX_LAYOUTS_IN_CACHE	1000	This is the maximum number of layouts that will remain cached in memory once they have been built upon their first access. This layout caching improves the performance of ExtraView significantly. If this number is increased, attention should be paid to the amount of free memory available in the application server. If this number is too high compared to the amount of memory available, then the system may run out of memory
MESSAGE_WORKER_COUNT	1	The behavior setting MESSAGE_WORKER_COUNT controls the number of Message Workers to be allocated for message and title access. This must be set to an integer from 1 to n. Each Message Worker requires a single Connection to be permanently allocated to that Worker. Therefore, setting this value to a high number (near the ConnectionPool's maximum number of connections, for example), will starve other threads of available connections. Normally 1 is the recommended value. The maximum number should be the maximum number of connections defined in your Configuration.properties file
MULTI_LOCALE	NO	ExtraView will behave as a single locale system, using the language specified in the behavior setting named DEFAULT_LANGUAGE when this value is NO. When it is set to YES, then the administrator may add additional language locales to the system, and provide localized messages and metadata for each locale
REAL_IP_HEADER		The name of a request header to use to retrieve a client's IP address. This is useful if the application server works behind a proxy or load balancer, and the request comes from that IP, and a header was added with the clients original IP address. If

this behavior setting is configured, and the header exists in the request, and it has a reasonable value, then it is used as the IP address. Many proxy servers, load balancers and some application servers do not set the standard HTTP header properly. This setting may be used when your application server works behind a proxy server or load balancer, and users are seeing unexpected session expiries.

RESPONSE_COMPRESSION_THRESHOLD	0	This setting provides optional compression for the responses to Ajax calls from the user's screen. A setting of 0 does not provide any compression. The value is the number of kilobytes of data that are to be sent to the browser within the Ajax response. When the amount of data to be sent is above this threshold, the data will be compressed on the server, and then expanded by the client browser. There is a tradeoff that is highly dependent on the speed of the network between your users and the server. The higher the volume of data to be sent, the longer the transmission time, but the client browser then needs time to decompress the data. If you believe your installation will benefit from this compression, we suggest starting with a value of around 50. If your network is slow, a smaller number may provide better performance.
SERVICE_STATISTICS	NO	YES writes additional statistics to the application server log, providing detailed timings of the execution of each service and how much data was generated and sent to the client browser. NO turns this facility off
SITE_URL		The full URL of the site, e.g. http://extraview.company_name.com. This setting is optional. If it is not provided, ExtraView assigns two values internally, one to be used from within the company's network, and one to be used externally. When the ExtraView application server initializes, the server will look at the value in this behavior setting. If the incoming request to start the server originated from a SSL session, the two values set internally will similar to: Internal URL: http://extraview.company_name.com/evj External URL: https://extraview.company_name.com/evj The values set can be seen in the application server log in the startup section. If the incoming request to start the server originated from a standard HTTP session, the two URL's will be identical. There are some circumstances where the administrator will need to set a different URL, and this is done by placing the value directly into the SITE_URL value. The most common reason for requiring this is when you see a notification email from ExtraView, where the Edit button is missing and its URL is malformed. The setting is also sensitive to your web server configuration and any redirection provided there. If you have set up redirection in your web server, these URL's may be different. If SSL is being used within your environment and the URL to access ExtraView therefore begins with https://, then the SITE_URL must be set to https://extraview.companyname.com/evj in order for drill downs from email and other remote applications to work correctly. With redirection, it is probable that you may not use a form such as http://extraview.company_name.com/evj/ExtraView to sign on. It is most probable that you will use http://extraview.company_name.com/evj. If you provide an incorrect value for this setting, it is highly likely that users will not be able to sign on to ExtraView.
USER_CUSTOM_CLASSNAME	com.extraview. usercustom. CustomCodeBase	Log timing data for every user custom method call. Valid values are YES and NO
USER_CUSTOM_ENABLE_METRICS	NO	Log timing data for every user custom method call. Valid

values are YES and NO

USER_CUSTOM_JAVASCRIPT	This setting provides the capability for custom JavaScript files to be loaded into the *Add* and *Edit* screens. To load multiple files, separate the entries with a semi-colon. The path(s) you enter are relative to the directory named **javascript** within your ExtraView instance. No checking is done by ExtraView for the existence of the file, so it is important to enter the correct path(s) and filename(s)

LDAP and SSO Behavior Settings

This section of behavior settings deals with LDAP and SSO settings. The available settings are:

Behavior Setting	Value	Description
ALLOW_SSO_AT_SIGN	NO	If set to YES, SSO user id's will be generated from the SSO primary key in its entirety. If set to NO, the User ID within the header is treated as a potential email address, and the characters preceding @ are assumed to be the User ID within ExtraView. For example, if the User ID within the header is john_smith@mycompany.com, then the resultant User ID is assumed to be john_smith
CUSTOM_AUTHENTICATION	NO, YES, HYBRID, LDAP or LDAP-HYBRID	This setting can have one of five values, NO, YES, HYBRID, LDAP or LDAP-HYBRID. When the value is NO, ExtraView uses its standard inbuilt authentication. If it set to YES, custom authentication (with user custom code) will be used rather than using the inbuilt user authentication scheme. If the value of HYBRID is selected, then ExtraView attempts to perform the custom authentication first. If this is unsuccessful, then the standard inbuilt authentication method is called. If it set to LDAP, LDAP authentication will be used rather than the inbuilt user authentication scheme. If the value of LDAP-HYBRID is selected, then ExtraView attempts to perform the LDAP authentication first. If this is unsuccessful, then the standard inbuilt authentication method is called. When the CUSTOM_AUTHENTICATION behavior setting is set to YES and a user's account is updated with the **Expire password** checkbox being checked, then if the custom authentication routine returns true indicating that the authentication was successful, the Change Password form is presented to the user during the login process. When the CUSTOM_AUTHENTICATION behavior setting is set to HYBRID and the user's account is set to expired (same as above), then it does not matter what the custom authentication routine returns as long as the credentials are valid, and the Change Password form is also presented for the user to change their password.
LDAP_ALLOWED_STALE_INTERVAL	240	The minimum number of minutes between LDAP upsert operations on a single user. This stops ExtraView accessing the LDAP server with every operation that requires user information in order to optimize performance. After this interval, a fresh check of the LDAP server is made

LDAP_DEFAULT_AREA	<AREA_ID>	The default area_id to be set when adding a new user by retrieving their details from the LDAP server
LDAP_DEFAULT_PROJECT	<PROJECT_ID>	The default project_id to be set when adding a new user by retrieving their details from the LDAP server
LDAP_HOST	ldaps://<hostname>:<port>	The URL to the LDAP server, e.g. ldap://blah.com:389
LDAP_MANAGER	<DN_FOR_USER_LOOKUP>	The "Security Principal" or user accessing LDAP
LDAP_PSWRD	<LDAP_MGR_PASSWORD>	The password to the LDAP server. Note that this does not appear in clear text, but as a row of asterisks.
LDAP_ROOT	<SEARCH_BASE_DN>	The root directory of the LDAP server or search base, e.g. ou=blahWorker, o=blah.com
LDAP_SEARCH_FITLER		LDAP filters may be defined in the behavior setting or may be defined in the Configuration.properties file. The behavior setting takes precedence over the Configuration.properties entry. LDAP filters are defined in RFC 2254. As an example, if you wanted to add a filter to only retrieve records with mycompany within the email address, you could set this as the filter: (mail=*mycompany*) The parentheses are essential.
LDAP_UPSERT	YES	This setting controls upserting LDAP information to ExtraView. THe possible values are YES or NO. When this value is NO, LDAP will not be used to add or update ExtraView user information. A valid LDAP user that is not already an ExtraView user will not be able to log in to ExtraView. When this setting is set to YES, the LDAP Background Task must be configured and must be running. The timer in LDAP_ALLOWED_STALE_INTERVAL is used to refresh the LDAP cached information within ExtraView. The information within ExtraView for a user is also refreshed when the user signs on.
LDAP_UPSERT_DEFAULT_USER_ROLE	<EXTRAVIEW_ROLE>	If this setting contains a valid role, then this is the role a user is given when the LDAP upsert takes place. If this setting does not contain a value, then the role defined in the behavior setting named LIMITED_USER_ROLE is used instead.
LDAP_USER_LOOKUP	YES or NO	When this behavior setting is set to YES, whenever a user performs an operation to lookup the details of another user, ExtraView will ask the LDAP server for the information. At the same time this is done, the information for the user within ExtraView, will be synchronized with the information within the LDAP record
SSO_DEFAULT_AREA	0	This setting is used to identify the Business Area ID for a new user being created within ExtraView, when SSO headers authenticate the new user, and it is the first time the user has signed on. In this case, the user is created automatically and this Business Area ID is associated with the user.
SSO_DEFAULT_PROJECT	0	This setting is used to identify the Project ID within the Business Area specified in SSO_DEFAULT_AREA, for a new user being

created within ExtraView, when SSO headers authenticate the new user, and it is the first time the user has signed on. In this case, the user is created automatically, and this Project ID is associated with the user. Set this to zero if you do not want to associate the user with a specific Project, but you have set SSO_DEFAULT_AREA to a value

SSO_SIGNOFF_REDIRECT_URL		This is a URL to which the user is directed, after signing off from ExtraView
SSO_STATE	NO	Enable Single Sign On in this instance (YES/NO)

Report & Query Settings

This section of behavior settings deals with the Report and Query settings. The available settings are:

Layout & Display menu - Report and Query Settings	Typical Value	Description
ABBREVIATED_HOME_PAGE	NO	A value of NO will display the sign on message and up to three user-defined reports on the Home Page. A value of YES will only display the sign on message
ALLOW_SEARCH_ DEACTIVATED_USERS	YES	The behavior setting controls whether you will allow users to include deactivated users when querying and the setting of USER_LIST_DISPLAY is set to POPUP. If the value is YES, then an additional option appears in the popup search screen. This option allows you to enable querying for deactivated users. Users can search for deactivated users when USER_LIST_DISPLAY is NO in the same manner as searching for other deactivated values
ALLOW_SEARCH_TEXT_UDFS	YES	When this setting is YES, then performing a keyword search will include the searching of UDF's with a display type of Text. This may have a performance impact on the speed of searches. Valid values are YES and NO
ALLOW_UNLIMITED_SEARCH	YES	Allow or disallow unlimited rows to be returned on queries when searching. Valid values are YES and NO. In large databases, the system administrator typically wants to stop users running reports that consume a large amount of resources. This is used in conjunction with LIMIT_QUERY_ROWS
ALLOWED_ATTACH_ SEARCH_FILE_EXT	txt,html,doc,htm,application/ base64. . .	The allowed attachment files of the types in this list to be searched for keywords, when the user checks the "Search Attachments?" box. If the file in the list to be searched does not have one of these extensions, it is skipped. This is used to skip searching of files such as image files, to speed the search process
DEFAULT_SORT_ORDER	ID:DESC	The default sort order for reports
DISPLAY_ALL_FIELDS	YES	When this value is set to YES, all fields to which the user has read permission will be generated into the all the query field lists from which a user can select for column reports, at all levels, for hierarchy reports. If the value is set to NO, then only the fields that exist in the edit screen of the

		Business Area and Project defined for the hierarchy level will be displayed. This keeps the number of fields displayed to a minimum, and most probably displays all the fields in which a user will have an interest in using as filters
DRILLDOWN_ATTRIBUTE	ID	Data dictionary entry name for search criterion. Typically, this is the ID of the issue, but it may become ALT_ID or another field according to how your system is configured
EXCEL_CELL_CHAR_LIMIT	31000	This specifies the number of characters exported to a single Excel cell. If the length of the text in a log area field exceeds this number, it will be truncated. Excel supports a maximum of 32,000 characters in a single cell, therefore this number must be less. Only log area fields can exceed this limit of Excel.
EXCEL_PRESERVE_LEADING_ZEROS	NO	EXCEL_PRESERVE_LEADING_ZEROS allows the spreadsheet to retain leading zeroes output. For example, if the setting is NO, then when 0001234 is output, the Excel cell will display 1234. If the setting is YES, then the Excel cell will display text as 0001234.
FILL_IN_REPEATING_RECORDS	YES	In text and Microsoft Excel reports, this determines whether to pad parent data in repeating rows with blanks. If NO is selected then the parent data will be repeated for each repeating row child values. Values are YES and NO
FULL_TEXT_SEARCH_COLUMNS	ITEM.SHORT_DESCR ITEM_TEXT.TEXT ATTACHMENT.FILE_DESC ATTACHMENT.FILE_NAME ITEM_UDF.VALUE ATTACHMENT_ CONTENT.CONTENT_BLOB	This is a comma-delimited list of the database columns that will be indexed and used to perform high speed keyword searches for users. If a column is in this list, and there is no suitable index for that column in the underlying database, search queries using standard SQL will be used when the user searches that column. There is typically no need for the administrator to modify the list in the value of this behavior setting as all the columns used to store text within issues are on the list.
GRANULAR_REPORT_DEFINITIONS	NO	By default, the report editors allow the end user to output all the report definitions with a single mouse click. When this setting has a value of YES, then the end user can elect to output any combination of the Report Definition, Report Filters and Custom Filters, though individual checkboxes.
HIGHLIGHT_LAST_CHANGE_USER	YES	If set to YES, the LAST_CHANGE_USER field will always be highlighted on email notifications and history report. Valid values are YES and NO
HISTORY_DISPLAY	ABBREVIATED, SIDE_BY_SIDE, LAYOUT	This setting allows the administrator to set which methods for the presentation of history are available to the end user. You may set one, two or three of the values ABBREVIATED, SIDE_BY_SIDE and LAYOUT, separated by commas. For each of these values, a button will appear on the menubar of the history screen, and the user can switch between the alternatives. The first value in the list is the default value when the user first enters history for an issue

HOME_PAGE_REFRESH_SECONDS	900	The frequency, in seconds, of the Home Page automatically refreshing itself. A value of 0 means the Home Page will not refresh. This allows a user to see data that is continuously being updated, as new issues arrive and are updated, altering the information on his Home Page reports
INSERT_REPORT_HEADERS	YES	This setting controls whether header and footer information is inserted into reports with a destination of Microsoft Excel or Text. Values may be either YES or NO. Note that headers and footers are always generated for reports output to the browser and Microsoft Word.
ITEM_TABLE_CARDINALITY		The optimal order for indexed queries.
KEYWORD_SEARCH_FIELD_SECURITY	NO	Determines whether field security is checked on keyword search for text fields. Use YES to check, NO to skip
LIMIT_GROUP_BY_ITERATIONS	40	This setting will limit the number of iterations produced by an Group By Container Report. The report will stop running after the limit has been reached. The last row of the table of contents will show an error message informing the user that the limit has been reached. The default is 40.
LIMIT_HOMEPAGE_QUERY_ROWS	20	Maximum number of rows that will be returned by any column report from homepage. This is used in conjunction with ALLOW_UNLIMITED_SEARCH to provide the limit of rows returned by a query.
LIMIT_QUERY_ROWS	10000	Maximum number of rows that will be returned by any search query. This is used in conjunction with ALLOW_UNLIMITED_SEARCH to provide a limit to the number of rows returned by a query. Note that a user whose current role is ADMIN, bypasses this check
LIMIT_SUMMARY_QUERY_ROWS	1000	Maximum number of rows that will be returned by a summary report. This is an exception to LIMIT_QUERY_ROWS which is the limiter for most other report types
LIMIT_WORD_DETAILED_RECORDS	10000	The maximum number of detailed records on a MS Word report. Use this if you encounter the bug in Microsoft Word (pre-2007) where it freezes when downloading or loading large HTML reports. This is an exception to LIMIT_QUERY_ROWS which is the limiter for most other report types
LIMIT_WORD_RECORDS	25000	The maximum number of records returned from a search to a MS Word report. This is used to limit the amount of HTML data being sent to Word (pre-2007), if you encounter the bug where Word freezes when loading a large amount of HTML data. This is an exception to LIMIT_QUERY_ROWS which is the limiter for most other report types
MASS_DELETE_TEMPLATE	VIEW_BUTTON, ID, AREA, PROJECT, SHORT_DESCR	This provides a list of the fields displayed on the review screens when the user clicks the Mass Delete button from Quicklists and Column reports
MAX_CELLS_ON_TASKBOARD	500	The maximum number of cells that will be

rendered when producing a Taskboard report. Browsers may run out of resources and consequently hang, on client computers if you attempt to render a huge taskboard with too many cells. This setting provides the means to set sensible limits on the size of taskboards

MAX_DRAG_NUM_PROMPT	5	This is the maximum number of issues which may be dragged and then dropped onto a report, before a warning is provided to the end user, indicating how many issues are being moved, and asking them to confirm that it is OK to proceed. The update proceeds without any warning to the user when the number of issues being dropped is equal to or less than this number
MENUBAR_SEARCH_TARGET_WIN	main	When you drill down to an issue from the navigation bar, a value of MAIN will place the results in the main window alongside the navigation bar. A value of __BLANK will open a new window for the display
MINIMUM_KEYWORD_SEARCH_CHARS	5	This setting controls the number of characters that a user must enter into a keyword search box that appears on report output. No refreshing of the results appear until the user has entered this minimum number of characters
MINIMUM_SEARCH_FIELDS	0, 0	This setting is used to force a user to select a number of filters before their queries will be executed. If you specify a single number, then this specifies the number of filters in addition to the KEYWORD filter that must be provided. If you provide two numbers, separated by a comma, then the first number specifies the number of filters in addition to the KEYWORD filter that must be provided and the second number specifies the number of filters that must be provided when no KEYWORD filter is provided. The default is 0,0. For large databases, a setting of 2 or 3 for each number typically provides sufficient control, to ensure that only a small section of the database is searched at one time, and that users do not attempt to download millions of records or perform complex queries across millions of records.
MS_OFFICE_CHARSET	UTF-16LE	The default character set for reports sent to Microsoft Office Products. The default value is UTF-16LE, which is the appropriate value for English language versions, and most foreign language versions of MS Office
MULTIPLE_VALUE_REPORTING	RELEASE	Must be RELEASE, MODULE or NONE - Note this is not on the GUI and is set to RELEASE as this is the only valid option at this time
QL_REPORT_LAYOUT_ AREA_PROJECT		This setting defines the business area and project to be used when creating a Quicklist report. If defined, the value of QL_REPORT_LAYOUT_AREA_PROJECT has two comma-separated values - Area_Title, Project_Title, where Area_Title and Project_Title must exist. These values are used to select the layout for the Quicklistif the business area and project can not be resolved uniquely in the

search filters for the Quicklist report. This happens if there are no business area/project filters selected and more than one exist, or there is more than one business area and project pair combinations selected. If QL_REPORT_LAYOUT_AREA_PROJECT is not defined, the current business area and project for the current user is used to select the layout.

QUERY_TIMEOUT_SECONDS	0	The maximum number of seconds that any SQL query is allowed to run in the database before it is aborted. This prevents any one query from taking all the database resources, and in the worst case, stops a query that may never end. A value of 0 allows a query to run to completion, regardless of how long it may take
RECORDS_PER_PAGE	20, 100, 500	The number of records per display page that a user can select on reports. These values will populate the list box on the Search / Report screens. If ALLOWED_UNLIMITED_SEARCH is YES, then Unlimited is also appended to the list of values
RECURRENCE_ISSUES	RECURRENCE_ISSUES	This value is used by ExtraView to relate issues together by the recurrence generator in the calendar report. It is highly suggested you do not edit this field. Editing this field after a recurrence has been generated will result in the loss of that recurrence
REPORT_DTL_ITEM_DATA_LAYOUT	NO	When this is set to YES Detailed Reports will be rendered with the layout for the each issue's area and project and the current user's role. If this is set to NO, the layout used will be that of the user's current area and project, and their current role
REPORT_FILTER_BY_ CURRENT_ROLE	NO	When this is set to YES, the user will only see reports on the Query / Report page that are valid for their current role. Typically this means they will see their personal reports, public reports and reports that are visible to their current role. If this is set to NO, then they will also see reports that are valid for other roles that they are eligible to see.
REPORT_IN_NEW_WINDOW	NO	If set to NO the Quicklist and Detailed Report will be opened in the main ExtraView window. If set to YES, the Quicklist and Detailed Report will be opened in a new window.
REPORT_LABELS_POSITION	TOP	Position of the labels on reports, relative to the data. Valid values are LEFT and TOP. When the value of LEFT is set, the detailed report uses the layout defined for the purpose. If you set the value to TOP, then the detailed report will be displayed in a single column, with the labels to the left and the values presented top to bottom. The order of fields is taken from the detailed report layout, from left to right, top to bottom.
REPORT_REL_ISSUES_EXCEL_TEXT	NO	When this behavior setting is set to YES, reports that are output to Microsoft Excel or to text, will include the RELATED_ISSUE_DISPLAY layout for related issues to the records being output. When this is set to NO, related issues will be suppressed on Microsoft Excel and text output.

		Note that browser output and Microsoft Word output will always contain the RELATED_ISSUE_DISPLAY for related issues if it has been defined.
REPORT_SUPPRESS_BLANK_LINES	YES	If YES, the detailed report will suppress any row where the data values are all null. Valid values are YES and NO. This is used to shorten the length of reports that may contain a significant amount of blank cells
REPORT_TABLE_WIDTH	100%	This setting controls the width of displayed information and tables on all the query screens and the home page report widths. The most common setting is to use 100% as the value, but the setting can be an absolute number of pixels for the width of objects on most screens presented to end users
REPORT_WITH_FIXED_WIDTH_FONT	NO	When this is set to YES, fields with a display type of "text area" and "log area" will display their contents with the FIXED_WIDTH_FONT. When this field is set to NO, these fields will display their contents using the DEFAULT_FONT. Note that this setting also applies to the display of log area fields on the *edit* screen.
RESTRICT_ROLE_BASED_REPORTS		Any report created for *** All Roles *** is visible to everyone that means RESTRICT_ROLE_BASED_REPORTS behavior setting is ignored if a report is created for All Roles. If the setting of RESTRICT_ROLE_BASED_REPORTS is left blank, all users will see all reports that match the role(s) they may adopt, if the role has been given access to role-based reports. If the setting of RESTRICT_ROLE_BASED_REPORTS is set to COMPANY_NAME, users will see all reports that match the role(s) they have, and that have been created by a user with the same company name as their user account, if the role has been given access to role-based reports. If the setting of RESTRICT_ROLE_BASED_REPORTS is set to COMPANY_NAME_OR_ADMIN, users will see all reports that match the role(s) they have, and that have been created by a user with the same company name as their user account or that were created by the ADMIN user, if the role has been given access to role-based reports. This setting does not affect reports saved for public reports - all public reports are available to all users, if their role has permission to that type of public report via the security permission keys.
RID_PAGING_DEFAULTS	0,0	This behavior setting determines whether the results returned within a related issue display are paged, with **Next** and **Previous** buttons to move between pages. The page size and the minimum number of rows where related issue displays are rendered in pages are separated by commas. A page size of 0 is unlimited, resulting in no paging.

Security & Session Settings

This section of behavior settings deals with the Security and Session settings. The available settings are:

Systems Control menu- Security and Session Settings	Typical Value	Description
ALLOW_CHANGE_LOCALE_AT_SIGNON	NO	If this setting is set to YES, then there is an additional prompt on the sign on screen, where users may elect to alter their language setting as they sign on. This only has meaning in a multi-locale system and where users are regularly changing which language they use
ALLOW_PASSWORD_AUTOCOMPLETE	NO	This setting allows or disallows your browser to cache the user's password and to auto-complete this for the user. Note that setting this to YES may be a security risk if users access ExtraView from public computers. Also note that this setting allows the browser's **Back** button to be used to go back and retrieve the original sign on page with the valid User ID and password. It is not recommended that you turn this feature on if you are not completely certain of who can access your installation
ALLOW_SECURITY_PERMISSIONS	NO	When you create a new user defined field it will be given read and write permissions for all user roles if the value of the setting is YES. If the value of this setting is NO then only the ADMIN role will be given read and write permission. The latter is a more secure mode
APP_HOME		Only used if value is not the default for the system. This is the path to the ExtraView Java servlet
AUTO_SIGNOFF_ON_USER_EXIT	NO	YES or NO to sign off user if the user closes the last ExtraView window, or navigates to another site. If NO, the session cookies remain and the user can press the browser Back button to go back to their ExtraView session. If YES, the user's session is terminated when they navigate to another site or close the ExtraView window
CACHE_COHERENCY_POLL_TIME		The number of seconds between updating cached metadata information within the server. This setting provides a compromise between between performance and up-to-date information being available across all nodes of a multi-server environment, and across different sessions within a single application server environment. The longer the cache interval, the higher the performance of the server, but the increased likelihood that a user will access stale information within the server cache. The default if you do not enter a value is 5 seconds. If you change metadata often on an installation with clustered application servers, the recommendation is to keep this value at 5 seconds so that any metadata change is propagated promptly to all the application servers and sessions. If you change metadata rarely, or the metadata is only changed on a staging server and then propagated to a production server, you can consider extending the time
CLIENT_IP_ADDRESS_CHECK	YES	YES or NO to check or ignore that the client workstation maintains a constant IP address during a session. Usually YES, but may need to be set to NO if your server is accessed via a proxy server
DEFAULT_TIMEZONE	America/Los_Angeles	This time zone will be used as the default for all new users who are created. Please make sure the time zone is spelled correctly. You can consult the Appendix on Time Zones for a list of valid time zone

ENHANCED_SECURITY	NO	This setting provides password access to the User Accounts Maintenance screen for enhances security
KEEP_FAMILY_SESSIONS_TIMEOUT	NO	If this behavior setting is NO, then each window that opens a new session within ExtraView will set up its own measurement against SESSION_EXPIRE_TIME_HOURS. If this is set to YES, then all sessions in all windows will share a single SESSION_EXPIRE_TIME_HOURS. The NO value provides for more efficient memory utilization on the server, but may lead to unexpected session timeouts of ExtraView for individual users, when they operate with a significant number of open windows
KEEPALIVE_INTERVAL_SECS	300	This is the number of seconds between polling operations from the *add* or *edit* screen. This timer is used to send a message from the user's browser to the server, to allow the server to keep track of users with open *add* or *edit* sessions. If the user closes their browser or navigates away from an *add* or *edit* page, then the server will be able to recognize this event.
LICENSE_USAGE_WARNING	90	This should be a decimal number. It will be interpreted as a percentage. When the number of concurrent licenses consumed exceeds this amount, ExtraView will send an email to the administrator to warn them of this event. If the number of concurrent licenses is 10 or less, then the email is sent only when a user is denied access because of no availability of licenses. If the setting is zero, then this warning is not generated
MAX_SIGNON_ATTEMPTS	3	The maximum number of consecutive failed sign on attempts allowed by an individual user before their account is disabled. The number of failures is measured in the period defined in SIGNON_PERIOD_MINUTES. Note that all users who have the administrator role defined by the setting ADMIN_OVERRIDE_ROLE are notified by email when a user's account is disabled. Note that if you are using custom authentication, then MAX_SIGNON_ATTEMPTS is not operative.
NOSPILL_SESSION_COUNT	500	Two numbers control the number of sessions that can reside in memory at any one time: • SPILL_SESSION_COUNT • NOSPILL_SESSION_COUNT A session is created in memory in response to a login by a user or as a "sub-session" of some previously created session. These sessions require memory resources to maintain the context of individual users "conversations" with ExtraView. When the number of sessions gets too large, they threaten the continued ability of the system to operate by consuming too much resource such as memory; therefore, there is a mechanism to "spill" old sessions to disk and retrieve them only when they are needed. When the number of sessions in memory reaches or exceeds the NOSPILL_SESSION_COUNT, the "oldest" sessions are written to the database and removed from memory. In this context, the "oldest" session is the one that has not been touched by some user action for the longest period. The writing of the session data to disk occurs as a background task and does not directly affect the creation or utilization of sessions that are in memory. Responsiveness is affected only when a disk-resident session is invoked by some user activity that requires a short delay to "de-serialize" the session and reconstitute it

in memory from the database. When the number of sessions in memory reaches the SPILL_SESSION_COUNT, the in-memory session cache is "full" and new session requests must wait for older sessions to be written to the database. In this case, there is a direct effect on responsiveness while the user waits for old sessions to be written and in-memory session slots to be made available. Often, this delay is not noticeable. Another way of looking at it is that the system is in one of three states at any one time: (N_SESSIONS is the number of sessions in memory)

1. Non-spill state: N_SESSIONS < NOSPILL_SESSION_COUNT In this state, no sessions are written to the database and new sessions are created directly in memory.
2. Spill state: SPILL_SESSION_COUNT > N_SESSIONS >= NOSPILL_SESSION_COUNT In this state, a background task is writing the oldest in-memory sessions to disk, and new sessions are created directly in memory without any delays.
3. Full state: SPILL_SESSION_COUNT <= N_SESSIONS In this state, a new session creation request will be delayed until at least one old session is written to disk to free a "slot" for the new session.

Note that NOSPILL_SESSION_COUNT < SPILL_SESSION_COUNT in all cases. If the behavior settings do not adhere to this invariant, the default values (480 / 500) will be used. Note also that these behavior settings must be set when ExtraView is started, and they are not inspected after startup. If they are changed, a restart of the ExtraView application server is necessary for them to take effect.

PASSWORD_EXPIRE_TIME_DAYS	0	This is the number of days which a user's password will last, before it automatically is expired. Every time this number of days elapses, the user will be prompted to create a new password when they next sign on. If this setting has a value of zero, passwords will never expire. Note that if a user is created when the value is zero, and this setting is then changed from zero to any number, then existing users passwords will never expire. The administrator will then need to set a new date for the expiry in the users accounts.
PASSWORD_REUSE_DAYS	0	This is the number of days that must go by before a password can be reused, after it is changed. The default value of zero disables checking for reused passwords.
PASSWORD_RULES	0,0,0,0	PASSWORD_RULES has four numbers separated by comma.

The first number is the minimum number of characters required to compose a valid password. The default is 0, which indicates that passwords may be of any length.

The second number is the minimum number of numeric characters. Numeric characters have values between 0 and 9. The default is 0 which indicates that passwords need not contain any numeric characters.

The third number defines the minimum number of characters in upper case. Upper case characters have values in the range A to Z. (All passwords are case sensitive.)

The forth number is the minimum number of non-alphabetic and non-numeric characters. These are characters that are not in the ranges of 0 - 9 and a - z and A - Z, but are valid characters to store in a password. Double-byte characters are not allowed, but characters such as -, =. +. !, @, #, $, and % are allowed.

REAUTH_REDIRECT_PARAM		After the re-authentication with a status signature, this is the URL and parameter that ExtraView will go to
REAUTH_URL		When performing a re-authentication with status signature rules, this is the URL that will perform the re-authentication
SECURITY_CACHE_MINUTES	30	Number of minutes before refresh of session security cache. If you alter security permission settings and do not wish to wait for this cycle to complete to automatically refresh the settings, you can sign off and sign on again
SESSION_EXPIRE_TIME_HOURS	24	Maximum session idle time in hours, the session data for a user is kept. After the period set in USER_EXPIRE_TIME_HOURS and before the value of this setting, ExtraView will attempt to restore a user's data when their session has expired. Beyond the time specified in SESSION_EXPIRE_TIME_HOURS the data will be lost. If the value of this setting is less than one minute or 0, then the session data will be kept permanently until the server is restarted. This is not recommended.
SESSION_MONITOR_POLL_SECS	300	This is the number of seconds between periodic tests from the server, for session removal due to user expire time activity. The server will test to see if the user still has an open session at this interval. This is only used if USER_TIMEOUT_SESSION_REMOVAL is YES. and if the user's browser has not reported to the server that a session is still active with the KEEPALIVE_INTERVAL_SECS timer.
SESSION_NAME_EXPIRE_TIME_HOURS	24	This is the session expiry period measured in hours for a user who occupies a named user license and who remains idle. Note that this is different from the period for users who occupy a concurrent license. Their expiry period is defined in the behavior setting named SESSION_EXPIRE_TIME_HOURS. After the period set in USER_EXPIRE_TIME_HOURS and before the value of SESSION_NAME_EXPIRE_TIME_HOURS , ExtraView will attempt to restore a user's data when their session has expired. Beyond the time specified in SESSION_NAME_EXPIRE_TIME_HOURS the data will be lost. If the value of this setting is less than one minute or 0, then the session data will be kept permanently until the server is restarted. This is not recommended.
SESSION_WARNING_INTERVAL_SECS	0	This setting works in conjunction with SESSION_WARNING_TIME_SECS. When the idle time for a user reaches SESSION_WARNING_TIME_SECS before the expiry time, the user will see a message on their screen warning them of the impending session expiry. This warning will repeat every SESSION_WARNING_INTERVAL_SECS until the expiry time, when the session will finally expire. If the user takes some action to submit the form on the screen to the server during this process, the timers are reset and the user session remains active. A value of zero means that no repeat warnings will be given.

SESSION_WARNING_TIME_SECS	0	This setting works in conjunction with SESSION_WARNING_INTERVAL_SECS. When the idle time for a user reaches SESSION_WARNING_TIME_SECS before the expiry time, the user will see a message on their screen warning them of the impending session expiry. This warning will repeat every SESSION_WARNING_INTERVAL_SECS until the expiry time, when the session will finally expire. If the user takes some action to submit the form on the screen to the server during this process, the timers are reset and the user session remains active. A value of zero means that no warning of an impending session expiry will occur.
SIGNON_PERIOD_MINUTES	5	Once a user fails to sign on because of an invalid password, they will be allowed to make a total of MAX_SIGNON_ATTEMPTS in this period. Note that all users who have the administrator role defined by the setting ADMIN_OVERRIDE_ROLE are notified by email when a user's account is disabled
SPILL_SESSION_COUNT	1000	Please see the entry for NOSPILL_SESSION_COUNT
SYSTEM_LOG_EXPIRE_TIME_DAYS	30	This is the number of days of history in the system log and the user signon log that information will be kept before it is purged from the system. The default is 30 days but you may increase this if you want to retain the information for longer periods. Stale information is purged from the log each time you access the logs, but if this setting is changed before you access the logs, then the information is retained.
USER_EXPIRE_TIME_HOURS	8	Description - This is the time in hours for which a user will remain signed in to ExtraView, when his computer is idle. ExtraView recognizes a user as still active when an action causes the user to access the server. When a user's session expires, ExtraView will ask the user to sign in again, and will attempt to restore any data being inserted or updated, and take the user to the point where they were working. Note that the ability to restore data is only enabled for the period specified in the setting named SESSION_EXPIRE_TIME_HOURS. For this reason, SESSION_EXPIRE_TIME_HOURS should always be equal to or greater than USER_EXPIRE_TIME_HOURS. If the value of this setting is less than one minute or 0, then the session will not time out.
USER_TIMEOUT_SESSION_REMOVAL	NO	The value of YES implies that sessions are killed after USER_EXPIRE_TIME_HOURS of inactivity, unless the user has sessions that remain active in the add or edit screen mode. NO implies that these sessions are not killed after this period of inactivity. The user's browser is checked by the server every SESSION_MONITOR_POLL_SECS to see if the ExtraView session is still active.

User Settings

This section of behavior settings deals with the User settings. The available settings are:

Users menu - User Settings	Typical Value	Description

ALLOW_PASSWORD_CHG_AT_SIGNON	YES	This setting controls the visibility of the Change Password link on the user sign on screen. If the value is YES, the link is shown, if the value is NO, the link is not shown. Also note this link is inoperable if the setting named LDAP_USER_LOOKUP is equal to YES.
CHECK_SAME_FIRST_LAST_NAME	YES	This setting gives the administrator control over whether a check is instituted when adding a new user or updating an existing user, as to whether ExtraView will provide a warning if the combination of first and last name is already used
COMPANY_NAME_LIST_UDF		If this field is populated with the name of a User Defined Field with a display type of list, then the Company Name field on user account administration screens becomes a list, with the values being populated from the UDF. This accompanies the text field for entering company names. The UDF may also be used as a regular field on add and edit screens as well as being used on reports.
CONTACT_ADMINISTRATOR	NO	When this setting is YES, a prompt appears on the sign on page, with a default message of "Contact Site Administrator?". When this link is pressed, a mailto the user defined in the setting named EMAIL_ADMINISTRATOR_USER_ID is started, allowing the user to send a message. When the setting is NO, the prompt does not appear. This feature may not work correctly with Asian character sets or European accented characters with all email clients. The mailto protocol allows pass-through of non-ASCII characters in the subject and body using the RFC-2047 standard, which is not implemented correctly in many email clients. At this time, the only mail client that appears to be fully compliant with the standard is Mozilla Thunderbird. The RFC-2047 allows the specification of the character set for the subject and body (and other header fields) for the mailto protocol URI. The character set that ExtraView uses is the character set specified in the EMAIL_CHARACTER_SET behavior setting, or UTF-8 if this is not specified. RFC-2047 only comes into play for these fields when non-7-bit-printable ASCII appear in the subject and/or body fields. Thus, the fields will be encoded when double-byte or European accented characters are present in those fields. ExtraView uses the "B" (base64) encoding in all cases to transmit non-ASCII characters.
DEFAULT_START_PAGE	Home Page	This setting allows the Start Page to default to one of several screens when creating a new user. This sets the start page of the new user. The value of *Home Page* is the system default. The potential values are:

- Home Page
- Workspace
- Query Screen
- Report Screen
- Administration
- Add Issue Screen

It is possible for the administrator to add additional start pages through the administration utility which may also be selected as the default start page by using this setting

ENABLE_PRIVACY_GROUPS	YES	Turn the Privacy Group feature on and off. Valid values are YES and NO. If you set NO, then user administration screens will not offer the ability to set privacy groups for users, and you will not be able to create and maintain privacy groups
ENFORCE_DETAILED_USER_INFO	NO	This turns extra fields on for mandatory field checking on user account screen. Valid values are YES and NO. Note that although the user's primary email address may be mandatory when this

value is YES, it is not mandatory to enter the primary email address when the setting EMAIL_NOTIFICATION is set to NO

IGNORE_DEACTIVATED_USER_FIELDS	ORIGINATOR	This is a delimited list of fields with a display type of USER. Users will not be warned if they edit an issue which has a field in this list with a deactivated user. If a USER field does not appear in this list, the user will always be warned if the field has a deactivated user, when they edit the issue.
OMITTED_IMPORT_USER_COLUMNS		This is a comma-separated list of column names in the table named security_user. The list of columns named here will be ignored when an XML export file is imported into a target database. Thus fields that are frequently altered by a user, such as SECURITY_PASSWORD, will not be overwritten during the import of data. Note that this setting should be set in the target system that is importing the data. It is not referenced when exporting the data. The topic on importing user information has a complete list of the fields you can reference. You should not omit the SECURITY_USER_ID or the LOGIN_ID fields from the export, else you will not be able to successfully migrate users with the XML export / import
SELF_REG_USE_EMAIL_ADDR_AS_UID	NO	If this setring has a value of YES, then the email address is used for users who use the self registration process, as opposed to allowing the user to choose a User ID. The email address becomes the User ID when the database is updated
USER_DEFINED_START_PAGE	YES	If this is set to YES, the user is allowed to set their start page when entering ExtraView, to a commonly used page such as the Home Page, the Add Issue screen or the Search / Report screen. This is set on their personal preferences page
USERNAME_DISPLAY	FIRST	Display choice for user names (ID, LOGINID, FIRST, LAST). For example, if the users name is Mary Jones and their user ID is MSMITH, and their Alternative ID is MJONES then ID would display msmith, LOGINID would display mjones, FIRST would display Mary Jones and LAST would display Jones, Mary
USER_ADMIN_DISPLAY_COLUMNS		The default list of fields displayed on the user accounts maintenance screen is: User ID, User Name, Email Address, Company, Enabled, License type, Last access, Date Created, Date Last Updated. If you want to alter this list, you can place a list of the fields from the database table security_user in this setting. The fields you can place on the screen are chosen from: USER_ID, LOGIN_ID DISPLAY_NAME, PHONE, EMAIL, JOB_TITLE, TIMEZONE, ROLES, COMPANY_NAME, PAGER, and MOBILE. To include any of the user defined field names, use the convention USER_FIELD_1 through USER_FIELD_10. If you want to include a field that has been configured as a user attribute, then use the convention ATTR:field_name, where field_name is the name of the user defined field used for the attribute. This convention allows for the possibility that some of the attribute fields may have the same name as the database field name with the user information. Note that the fields ROLES and ATTR:field_name have the potential to contain multiple values in a single record.
USER_LIST_DISPLAY	LIST	POPUP or LIST. If POPUP, then users are selected for adding and updating issues via a popup window. If LIST, the user names appear in a select list. Typically, if you have a large number of users, you will use POPUP
USER_POPUP_COLUMNS	YES	The fields displayed in a user search popup window are configurable using this setting. This setting takes a list of fields (using the schema column names) from the security_user table and presents these to the user. If there is no value provided for this

setting, then the following fields are displayed: User ID, Name, Phone, Email address, Company name, Pager and Mobile number. You can select from any of the following fields: USER_ID, DISPLAY_NAME, PHONE, EMAIL, JOB_TITLE, COMPANY_NAME, PAGER MOBILE, USER_DEFINED_1 through USER_DEFINED_10

USER_SELF_REGISTRATION	YES	You can allow your users to self-register, or you can be the registrar. When this is set to YES, a prompt appears on the Sign On screen, allowing a user to navigate to a page where they can register themselves as a user in ExtraView. When a user registers in this way, they will only be given the privileges of the user role defined in the behavior setting named LIMITED_USER_ROLE, but no user role is set in their account. When a user self-registers, an email is sent to an administrator, as defined by the following settings with Email Settings on the Email Notification administration menu. The user created will be in the business area and project defined by the behavior settings named SELF_REG_DEFAULT_AREA and SELF_REG_DEFAULT_PROJECT. Note that for users to be able to self-register, that the GUEST user role must have read and write permission to user fields. The most important of these is USER.USER_LAST_NAME. The user's default AREA and PROJECT is assigned from the behaviour settings named SELF_REG_DEFAULT_AREA and SELF_REG_DEFAULT_PROJECT. You should ensure that these behavior settings have valid entries when you wish to allow users to self-register. The EMAIL_NOTIFICATION setting must be set to YES as well, in order for user self-registration to be active.

Workflow Settings

This section of behavior settings deals with the Workflow settings. The available settings are:

Workflow menu- Workflow Settings	Typical Value	Description
ADMIN_OVERRIDE_ROLE	ADMIN	Name of the user role that bypasses several security controls, and has special properties. Users who have this set as their current role have the following properties:

- Status change override privileges. The status change rules are not obeyed for this user role. This is a significant reason why the ADMIN_OVERRIDE_ROLE user role should not be used for normal operational work
- The security key CF_SECURITY_ROLES in combination with the admin-bypass-group, determines what user role you need to be in order to see and edit other users' privacy groups and user roles. For example, if the ADMIN role has read and write privileges, but SUPPORT is the ADMIN_OVERRIDE_ROLE user role, members of the ADMIN user role will only be able to edit privacy groups and roles for users who are also in the ADMIN role. If ADMIN is the ADMIN_OVERRIDE_ROLE user role, members of ADMIN can edit privacy groups and roles for all users
- If ExtraView access is disabled by an administrator, only members of the ADMIN_OVERRIDE_ROLE user role will be

<table>
<tr><td></td><td></td><td>able to access ExtraView, until ExtraView access is restored</td></tr>
<tr><td>ALLOW_BILEVEL_GROUPS</td><td>NO</td><td>This setting is provided for backwards compatibility. From version 5.0 onwards, bi-level relationship groups have been superseded by many-to-many and one-to-many relationship groups. Only set this to YES if your system was created prior to version 5.0 and you previously used bi-level relationship groups. With a value of NO, you cannot select bi-level as a value for a new relationship group.</td></tr>
<tr><td>AUTO_POPULATE_ALLOWED_VALUES</td><td>YES</td><td>When this setting is YES, then in a popup from an add or edit screen that adds a new value to a list, and that list is the child within an allowed value relationship, then the new value will be added to the list and will be automatically added to the currently selected parent(s). If the value is NO, then the user will be prompted to select the Business Area, Project and the specific parent values to which the newly created child value will be added within the allowed value relationship.</td></tr>
<tr><td>ATTACH_SELECT_CHECKBOX</td><td>UNCHECKED</td><td>This setting controls behavior of the Attachment Select box (ATTACH_SELECT) on the Add and Edit layouts. Valid values are UNCHECKED, CHECKED and CHECKED ON ADD. A value of CHECKED or UNCHECKED will set the checkbox for an attachment on entry to the screen or when a new attachment is added. A value of CHECKED ON ADD will cause the checkbox to be checked only when the attachment is first added, otherwise it will be unchecked</td></tr>
<tr><td>ATTACHMENT_CONTROL_FIELDS</td><td>NO</td><td>This setting is only used in conjunction with the Java user custom exit named ucAllowAttachmentOperation. This setting enables the functionality within this method. When the value is NO the user custom exit is not called. When the value is YES the code within the method is called and acted upon</td></tr>
<tr><td>AUTO_POPULATE_ALLOWED_VALUES</td><td>YES</td><td>When this setting is YES, then in a popup from an add or edit screen that adds a new value to a list, and that list is the child within an allowed value relationship, then the new value will be added to the list and will be automatically added to the currently selected parent(s). If the value is NO, then the user will be prompted to select the Business Area, Project and the specific parent values to which the newly created child value will be added within the allowed value relationship</td></tr>
<tr><td>AV_INTERSECT_WITH_NULL_PARENT</td><td>NO</td><td>When there are multiple allowed values defined with the same child field, and one of the parents does not have any children defined when the selected value is *None*, the default behavior is to show no values in the child field, as there are no valid values for *all* parent field selected values, and the default value for this setting is NO. However, older installations may have configurations where they expect that if one of the parent selected values is *None* that this parent field is not included in the determination of which child values to display, and the child field should display the intersection of the permitted child values for the other parent(s) field(s). In this case, set this behavior setting to YES.</td></tr>
<tr><td>AV_NULL_PARENT_IS_NONE</td><td>YES</td><td>If the value of this setting is NO and no parent value is selected in an allowed value relationship, then child</td></tr>
</table>

		fields will be populated with all the possible child values. If the value of this setting is YES, then the values displayed in the child field will be those defined for the parent value of * None *. If the allowed value relationship utilizes reverse lookup, this setting is ignored.
CLONE_RELATION_GROUP		Relationship group name for clone relationships
COMPANY_OVERRIDE_FIELDS		This setting contains a comma separated list of field names with a display type of User, that must exist and that contain the users that the company name security feature works with. By default company name security works with only the ORIGINATOR field. This setting extends the feature to these other fields
COPY_ATTACHMENT_ON_CLONE	YES	When this setting is YES, attachments will be copied to the new issue when an issue is cloned
DEFAULT_USER_ROLE	GUEST	Default role for users, if they have not been assigned a role
DISABLE_INACTIVE_UDF_WARNINGS	NO	When this value is set to YES the warning message caused by references to inactive UDF list values will be disabled. The default value is NO.
DISALLOW_AREA_0_DATA	NO	When this setting is NO, issue data may be entered into AREA 0. This is provided for backwards compatibility for installations created prior to version 4.2. Installations created with version 4.2 and greater should not allow issue data to be placed in AREA 0. With installations from 4.2 onwards, this should be set to YES
DISALLOW_PROJECT_0_DATA	NO	When this setting is NO, issue data may be entered into PROJECT 0. This is provided for backwards compatibility for installations created prior to version 4.2. Installations created with version 4.2 and greater should not allow issue data to be placed in PROJECT 0. With installations from 4.2 onwards, this should be set to YES
EMBEDDED_LAYOUT_REFRESHES	NO	This setting is provided for full backwards compatibility with all versions prior to 6.0. When the value is NO (the default), sublayouts such as those rendered with the LAYOUT SELECTED mechanism and with the FIELD/LAYOUT VISIBLE IF mechanism are refreshed using AJAX calls to the server. This is fast, but browsers do not have the capability to retain the initial presentation of the fields refreshed. If you set the value to YES, then the sublayouts are refreshed by a return to the server, and the alignment is retained. The tradeoff is between speed and the alignment of the fields on the screen. For many purposes the precise alignment of embedded layouts is not significant and not noticeable by users
ENABLE_COMPANY_NAME_ACCESS	YES	When this behavior setting is set to YES (the default value), a user's company name overrides the privacy setting for an issue and gives the user access to all records that were originated by other users with the same company name. Valid values are YES and NO
ENABLE_GOOGLE_LANGUAGE_API	NO	When this setting has a value of YES, then the Google Language API is enabled. This relies on the presence of a JavaScript function in UserJavaScript.js. This function may be customized by the administrator. See

the [Application Programming Interface guide](#) for more details on how to configure this feature.

ENABLE_PRIVACY_GRP_OVERRIDE	YES	**This setting may have the value of YES or NO. When set to YES, any one issue in a one-to-many relationship group may only have a single parent at one time. This implies that if you set a different parent ID for an issue, then ExtraView will remove the issue as being the child of the original issue, and set the new parent ID for the issue in its place. If you have NO for the setting, then an issue may have more than one parent at one time.**
ENFORCE_ONE_TO_MANY_RG	YES	This setting may have the value of YES or NO. When set to YES, any one issue in a one-to-many relationship group may only have a single parent at one time. This implies that if you set a different parent ID for an issue, then ExtraView will remove the issue as being the child of the original issue, and set the new parent ID for the issue in its place. If you have NO for the setting, then an issue may have more than one parent at one time
ENFORCE_STATE_CHANGE_RULES	NO	Switch to enforce state change rules. Valid values are YES and NO
ENFORCE_UNIQUE_RELEASES	YES	Enforces uniqueness of RELEASE_FOUND in a repeating record row. If the value of this default is YES, repeating rows cannot be added where the field named RELEASE_FOUND has a null value. In addition, you may not have duplicate RELEASE_FOUND values within a single record. If the value of this default is NO, a repeating row will be added if any of the writeable fields in the repeating record row on screen are not empty (including default values). Also note that the fields named RELEASE_FOUND and RELEASE_FIXED are the children of the field named PRODUCT_NAME. Therefore, if you use these fields on a repeating row record, then you must also have the PRODUCT_NAME field on the same layout.
FILTER_CHILD_VALUES	NO	This may have the values YES, NO or USER. The setting controls search behavior when the query filter contains repeating row values. If set to YES, only those rows that match the search criteria will be returned by the query. If set to NO, all rows are returned for any issue where at least one row matches the criteria. If set to USER, this behavior is controlled by a checkbox that appears on the search screen.
FILTER_MODULE_BY_CATEGORY	NO	Allow filtering of modules by category. Valid values are YES and NO. This allows selection of a subset of modules based on both the **module type** and a **category**. By selecting **Category**, only the corresponding module types are displayed as valid choices within the select list of modules on the *add* and *edit* screens. For example, products may have hardware, software and documentation modules. If a problem is related to software, only software the modules should be selectable. This feature relies on use of the **Category** field in the Item record to store a value for **Module Type**. When FILTER_MODULE_BY_CATEGORY is set to YES the value entered for CATEGORY is used as a second parent value for the module (MODULE_ID field) by providing a value for MODULE_TYPE in the same way that the product (PRODUCT_NAME field) is used as the first parent value. Note that this requires identical enumerated values for both CATEGORY and

		MODULE_TYPE. In the Data Dictionary setup for the MODULE_ID field, the following should be set: First Parent Field Name = PRODUCT_NAME First Parent SQL = where product_name Second Parent Field Name = CATEGORY Second Parent SQL = and module.category To make these changes in the data dictionary, you must sign on to ExtraView with the *admin* user account.
HIDE_ISSUE_ID	NO	Controls the title of the detailed report and the edit screens. When this is set to "YES" the standard title will be replaced with the text "ExtraView" and the current time
ISSUE_LINK	DETAILED_REPORT	This setting has two possible values, DETAILED_REPORT and EDIT_SCREEN. When ExtraView automatically places a link on an issue within a report, this setting defines the destination when the user clicks on the link.
ITEM_ID_DISPLAY	ID	When this setting has a value of ID, then the internal issue ID is used in all the main display screens as the issue number that the user is dealing with. However, if the administrator sets their own logic to process the unique reference number to each issue, then setting the ALT_ID as the value of this setting tells ExtraView to display the ALT_ID value in place of the ID value
LIMITED_USER_ROLE	GUEST	This role does not own and cannot be assigned problems. This is typically the guest or customer user role in your installation
LINK_MODULE_USER	ASSIGNED_TO	Links the module owner field to the specified user field. Can be ASSIGNED_TO, CONTACT or OWNER
MINI_HISTORY_FIELDS	STATUS, TIMESTAMP, LAST_CHANGE_USER	This setting defines the fields that will be displayed on the MINI_HISTORY field when it is used on a layout. The STATUS field should always be used, and up to four additional fields can be defined. The additional fields that are available are: SEVERITY_LEVEL, PRIORITY, STATUS, PRODUCT_NAME, OWNER, TIMESTAMP, ASSIGNED_TO, PRIVACY, LAST_CHANGE_USER, ALT_ID, AREA, PROJECT, CATEGORY, RESOLUTION, PRODUCT_LINE, DATE_LAST_STATUS_CHANGE, CONTACT, ORIGINATOR, and HIST_TIMESTAMP
MULTIPLE_FIELD_SEPARATOR	-/-	Used as a separator for child level multi- value UDF's
RELATION_GROUP_DEFAULT		The name of the relationship group that will be used for any issue being placed into a relationship group. This default can be overridden by using a layout cell attribute on the RELATED_ISSUE_DISPLAY, or by using the field RELATIONSHIP_GROUP on the add or edit layout to offer the user a choice of relationship groups
RELATIONSHIP_GROUP_EMAIL_LIMIT	30	When you update the status of an issue within a relationship group, each issue will be subject to the standard notification process. When there are more than RELATIONSHIP_GROUP_EMAIL_LIMIT entries in the group, notification is only made to the users in the parent issue, and a comment is inserted on the issue with this information
RELATIONSHIP_GROUP_LIST_UDF		This setting is the name of a user defined field with a display type of list. This UDF is used to maintain a list of the names of relationship group for related issue display. The UDF is maintained by ExtraView as relationship groups are created or deleted. The UDF can

also be aliased by other UDF's to provide an unlimited number of fields that can be used to identify the relationship group names to support multiple related issue displays on a single edit screen.

RELATIONSHIP_GROUP_MAX_DISPLAY	0	Maximum number of issues to display on the relationship group screen. A value of 0 will display all the records. Note that if your role is the one defined in ADMIN_OVERRIDE_ROLE then all the issues will always be displayed.
RELATIONSHIP_LINK_DISPLAY	SELECT	Display choice for relationship group indicator on edit screen (LINK or SELECT)
RELEASE_SORT_ORDER	DESCENDING	ASCENDING or DESCENDING
RG_UPDATE_BILEVEL_ONLY	NO	This behavior setting is for backwards compatibility with ExtraView version 4.x. If you set the value to YES, then related issue updates are only triggered when a change to the STATUS field takes place. If this is set to NO then updates to related issues may be triggered on different events and many updates to different fields are possible
RULES_DEFAULT_CALENDAR	24_BY_7	This setting selects the number of business days in a week. Valid values are 24_BY_7 or WEEKDAY. The WEEKDAY calendar runs Monday through Friday. The 24_BY_7 calendar uses all days
SAVE_AND_CONTINUE_ACTION	PAGE	This setting determines how the Update and Continue button on Edit screens behave. The valid entries for this setting are AJAX and PAGE. When the PAGE option is set, all LOAD directive business rules are triggered when the button is pressed by the user, after the issue is updated. When the AJAX option is set, the issue is saved, no LOAD directive rules are executed, but any values changed by the update process are displayed.
SAVE_AREA_PROJECT_CHANGES	NO	If the SAVE_AREA_PROJECT_CHANGES behavior setting is YES, then when a user changes their business area and/or projects while entering or updating an issue, the change will be reflected in both the current session and set as their personal default. If the behavior setting is set to NO, then any changes to the business area and/or project will not affect the user's personal setting or the value for the current session. This setting is not applied when the user is operating within a workspace
SAVE_CONTINUE_TIMEOUT_SECS	900	This is the amount of time in seconds between the execution of the SAVE: business rule and the time the issue is updated automatically within the database. If the user submits the issue within this time, no actions occur. This is used to perform a temporary save of an issue in case of a system or other failure.
SAVE_ROLE_CHANGES	YES	Unless you have custom code that dynamically switches roles back and forward for a specific requirement of your site, then this should always be set to YES, allowing any change of role for a user, by themselves or by an administrator to be saved into the ExtraView database. If there is custom code that switches a user's role and you do not want to make this role change stick in the database or be visible to the user, then the value should be set to NO.
SELF_REG_DEFAULT_AREA	0	This setting is used to identify the business area ID to associate with a new user, when the user is created thru self registration

SELF_REG_DEFAULT_PROJECT	0	This setting is used to identify the project ID to associate with a new user, when the user is created thru self registration
SELF_REG_USE_COMPANY_NAME	NO	When set to YES the value of COMPANY_NAME application default will be used for self-registered users. Otherwise, the user_id will be used for user's company name
SEPARATE_WORK_FLOW	PRODUCT	Allow separate work flows per USERGROUP (role) or PRODUCT or NONE
SHOW_CLOSED_REL_GROUPS_PERIOD	0	The list of items on a relationship group will be displayed until all issues have been closed for a minimum number of days specified by this value. Valid values are numbers equal to or greater than 0. 0 means that as soon as all items within a relationship group are of the status specified by STATUS_CLOSED_NAME they will not be displayed on the list of relationship groups.
SORT_SELECTED_VALUES	NO	When this is set to NO, all multi-valued lists will be displayed in the order set by the field's standard sort sequence. When this value is set to YES, all selected values will be displayed at the beginning of the list, using their sort order, then the non-selected values will be displayed using their sort order
STATUS_CLOSED_NAME	CLOSED	Data dictionary name of the closed status of a problem. This value is typically set when configuring a new system, and should not be changed after that time. The consequence of changing this value, and using a different value is that history is no maintained, and it will not be possible to either know when past issues were closed.

Business Rules

There are many occasions when you need to set the value of a field based upon the value of one or more fields. Contemporary products typically achieve this via modifying the programming language. ExtraView allows you to set field values using this administrative feature. Field values can be set at different points in the process.

Business rules are also used to link together values from different business areas, creating relationships between different Business Areas. For example, a relationship can be created to populate a set of fields on a field within a Customer Issue business area, based upon the value of a customer name created in business area that contains a list of customers.

Rules may be defined as either global in scope, or to only be executed within a specific business area within your installation. Global rules have precedence over rules defined in a specific business area. Therefore, rules in a specific business area are executed first, and then the global rules are executed. This ensures that rules defined with a global scope will be the last to be executed. There is no checking of whether rules in a specific business area may be in conflict with one or more rules in the global area. It is the administrator's responsibility to ensure that rules work together. To avoid the possibility of a conflict, you may set a policy that no rules will be created as global, and are therefore all local in their scope. When the administrator views the list of business areas, they will only see those to which they have permission, thus enforcing the privacy of rules set in a business area to which the current administrator does not have permission.

For rules created in the global area, you may specify the fields AREA and PROJECT as part of filters. Indeed you may need to specify these as filters for the desired execution. For rules specified in an individual area, there is no need to specify the AREA as a filter, and all rules imply that the AREA is selected as a filter. You must supply a PROJECT filter if you want to confine the scope of a rule to an individual PROJECT. Email rules can also be created, giving fine control over who will receive email upon the creation or updating an issue. The default notification scheme in ExtraView will send email to any user whose name appears on a form, and to members of any interest list. Email rules can complement or replace this arrangement, and give precise control over the notification process. For example, you might want to send notification to a user, only when the Status changes to a specific value, or if the Status

changes. Email rules are only applied after you submit an issue to be inserted or updated, and therefore the names that are added to the mailing list that is displayed on the *add* and *edit* screen will not display these names.

Note that ExtraView's inbuilt notification rules remain active, and users will be notified as normal, unless you set the behavior setting named SUPPRESS_STANDARD_EMAIL_LIST to a value of YES. This can be found in the Email Settings administrative function.

Editing the Business & Email Rules

Internet browsers have a limitation of approximately 1.5MB of content within the area into which you enter your rules. Under normal circumstances, this limit should not pose any problem, but if you cannot update your rules because of this limit, you should make sure that you write your rules as efficiently as possible. Other hints to help are to keep comments brief and to not have a significant amount of white space.

Automatic Backup of Rule Updates

As with all metadata changes, when you update the rules an entry is made in the System Log of the event. In addition to the usual information, a copy of the old rules text is also placed in the System Log. You may copy the text from the System Log and paste it back into the rules text area if you need to restore a prior set of rules.

If the rules text is greater than 3,800 characters in length, there is insufficient room to hold all the information in the System Log, so a partial entry is made there and the complete rule text is written to the application server log file.

Language Description

- [Add new comment](#)

The language that creates Business Rules is straightforward and composed of two primary elements.

Element	Purpose
Directives	There are a small number of directives that create holders for statements. These directives dictate when a statement is executed, for example before an *add* or *edit* screen is displayed, or before or after an update to the database is

performed. There are also a number of directives that help with other tasks such as mailing, linking items and debugging. All directives are contained within the characters <== and ==>. Each directive may be repeated any number of times within the rules, usually to maintain clarity and to group statements for a single business function together.

Statements Statements are logical expressions that perform specific actions within each directive. For example a statement may be assigned a discrete value or assigned the value of another field. You may use logical "if" statements. There are a number of special values and special actions that may be used within statements.

Business rules are organized by Business Area, and you may also define global rules.

Directives

Rule directives have one of the following forms:

```
<== directive ==>
```

```
<== directive ==> value(s)
```

```
<== directive name ==> value(s)
```

The same directive may be specified any number of times within the rules. This allows the grouping of rules for a specific function together, as opposed to grouping all the rules by directive type which may not be the logical way in which to peruse them. All rules after a type directive within the rules will be interpreted as belonging to that type until the next type directive is encountered.

Directives are listed in the following table.

calendar *calendar_name*	This directive is used to determine which business calendar is used in date computations. If no calendar directive is present, the calendar specified in the behavior setting named RULES_DEFAULT_CALENDAR is used. If this is not set to a valid calendar name, the 24_BY_7 calendar is used for date computations.
clone	This directive is used to execute rules and assignments during the clone process. For example, you can set the value, or reset the value of fields before the cloned record is saved to the database
debug	Logs rule activity and includes ExtraView internal values. Primarily for use by ExtraView support. All statements from the point where the directive is found, and forward from there, send deugging information to the application server log file
end	Stops parsing of the rule file. If this directive is encountered in the rules file, ExtraView stops further actions and ignores the remainder of the rules. Only the rules in the current business area stop executing. If you have also have rules in the global area then they will also execute, unless an `<== end ==>` directive also exists within there. This is useful in debugging your rules, as you can enter this directive at different points in your rules file as you test successive sections of rules.
info	Causes log entries to verify that rules are being executed. Minimal details are listed
link	**Linking Issues between Business Areas** This directive is used to specify how to connect linked issues from one business area to a different one, thereby creating a relationship between two business areas. For example, you can find one or more related issues in a different business area, from which a value is to be retrieved:

```
<== link customer ==> AREA='Customer', CUSTOMER=CUSTOMER
```

is interpreted that a rule action like:

```
ASSIGNED_TO=(customer).OWNER
```

which is used to set the Assigned To field in an Issue record to the OWNER field in a Customer record. If the link returns many records, and you only want to return a specific one, you can use a subscript as follows:

```
ASSIGNED_TO=(customer[nn]).OWNER
```

where nn is an integer which refers to the number of the record you want returned, starting with zero.

Linking Relationship Groups

The link directive can also be used to link related issue groups. For example:

```
<== link rglink ==> RG(RG_NAME = MY_RG, RG_TYPE = PARENTS)
```

where MY_RG is the name of an existing relationship group and RG_TYPE is one of PARENTS, CHILDREN, MEMBERS. If multiple records are referenced by the link, only the first one returned will be used for data assignments. For operations such as UPDATE all records will be processed. You may add an expression to the related issue link to form the link. For example:

```
<== link rglink ==> RG(RG_NAME = MY_RG, RG_TYPE = PARENTS), STATUS != Closed
```

will only form the relationship for issues that are not in the **Closed** status.

Relationship Group links are not order dependent; the type and name may appear in either order, and the RG_ may be omitted, so `RG(TYPE=CHILDREN, NAME=xxxx)` will work identically to `RG(RG_NAME=xxx, RG_TYPE=CHILDREN)`. Also, the keywords may be eliminated entirely, giving `RG(xxxx, CHILDREN)` as a valid link. Note that GREATGRANDCHILDREN and GREAT-GRANDCHILDREN are only supported in links, and are not supported at this time by RelationshipGroup code.

`{NULL}` and `{NOT NULL}` may also be used within link conditions. For example:

```
<== link AREA='Licenses', LICENSE_TYPE='License', COMPONENT={NULL} ==>
```

You may also add the `{EXISTS}` as a qualifier to a clause in a link directive. This will check the link, before execution, to find whether the value is present, or is a null value. For example, consider:

```
<== link cousinLink ==>AREA='Functionality', BROTHER_ID = PARENT_ID.{EXISTS}
```

If the `.{EXISTS}` was not present, all records with no value for the BROTHER_ID will be selected if the PARENT_ID is null.

You may also add a parameter of `DISABLE_LINK` to a LINK directive. This has the effect of suppressing the drilldown into the issue on the link. For example, the following is correct syntax.

```
<LINK AREA='Bugs' DISABLE_LINK>
```

If the value of the behavior setting named ITEM_ID_DISPLAY is ID, then the link uses the ID field to connect the linked issues. If the value is ALT_ID, then the ALT_ID field value is used in the links.

Linking with Wildcards

By default, reference to a link value returns the first value in the result list for an assignment statement. However, a subscripted value (such as `(linkname[n]).ID` may be specified, returning the nth value in the result list for an assignment statement. With this feature, all values in the list can be returned by specifying a wildcard subscript such as `(linkname[*]).ID`. This convention will return all values in the result list separated by semicolons.

Relationship Group Reference ID Feature

The RG link references currently permit the specification of RG_NAME and RG_TYPE. With this feature, you can specify RG_REF_ID, which specifies a field name containing the ID (or ALT_ID) of the relationship. For example, the following rules:

```
<== link rgParentIssueLink ==> RG(RG_NAME=MY_GROUP, RG_TYPE=PARENTS, RG_REF_ID=MY_TEXTFIELD)
<== load ==>
MY_TEXTFIELD = 12345;
MY_TEXTAREA = (rgParentIssueLink[*]).ID;
```

populate a semicolon-separated list of the parents of issue 12345 from `MY_GROUP` into the `MY_TEXTAREA` field.

load	Called before the *Add* or *Edit* screen is rendered. Rules following this directive are executed before the screen is displayed. Therefore you can set values of fields before the screen loads.
	Note that the load rules are only executed when the *Add* or *Edit* screen is first loaded. If the user changes the Business Area field or the Project field thereby generating a screen refresh, the load rules are not run again
log	From the point where the directive is encountered, information is inserted into the application server log, for administrators to verify rule execution. The log file shows conditional evaluations and assignments. The nolog directive ends the logging
mail	Mail directives are used to notify users of updates using the standard notification layouts. They can either complement the existing inbuilt notification rules or replace the inbuilt rules, depending on the value of the behavior setting named SUPPRESS_STANDARD_EMAIL_LIST. The mail directive will always contain rules that generate the notification based on the notification layout for the issue being inserted or updated.
	Note there is a key difference between the mail directive and the mail action rule which is typically used within the preupdate directive. The mail action is used to send an email based upon a template specified within the action.
nolog	This is the counterpart of the log directive and ends the logging set in progress with the log directive

object

This directive is used to define an object to be used as a link between two fields of different types. The most common purpose for this directive is to use it to enter a value into a text field, and have this value added to an existing list field.

Use the object directive to use a value entered in a text field to be inserted as a new value in a list field.

As an example, you may create directive rules similar to:

```
<== object cust ==> AREA='Customers', LINK=CUST_LIST, TITLE=SHORT_DESCR, PRIVACY=false
```

```
<== object cust.{guarantee} ==> AREA='Customers', LINK=CUST_LIST, TITLE=SHORT_DESCR, PRIVACY=false
```

```
<== object custpopup ==> AREA='Customers', PROJECT='Contacts', LINK=CUST_POPUP, TITLE=SHORT_DESCR,
PRIVACY=false, DO_NOT_MIGRATE=true
```

This link may be used in a rule similar to:

```
if (CUST_LIST.{changed}) {
  CUST_CONTACT = (cust).CUST_CONTACT;
  EMAIL_ADDRESS = (cust).EMAIL_ADDRESS;
}
```

This is a list of all the parameters that may be used with the object directive:

LINK=<LIST or POPUP field name>

Required - contains the name of the field with the list entry to be added, modified, or deleted

TITLE=<TEXTFIELD field name>

Required - contains the name of the field containing the title or text to be added to the list entry

PARENT=<field name>

Optional - contains the name of the parent field of the LINK field in an allowed value relationship

AREA=<area title>

Required - specifies the area of the issue where the object rule is executed

PROJECT=<project title>

Optional - specifies the project within the AREA where the object rule is executed. If this is not defined, then the object directive is executed for all PROJECTS within the defined AREA

RG_NAME=<relationship group name>

Unused at present

PRIVACY=TRUE | FALSE

Optional - the option to add `PRIVACY=true` will add the customer to a privacy group with the same title, and add the current user to that privacy group.

DO_NOT_MIGRATE=TRUE|FALSE

Optional - if this is provded and has a value of **TRUE**, then all User Defined Field values created with the OBJECT rule are not migrated from one ExtraView instance to another via the XML Export / XML Import utilities. This allows the addition of test data values within one instance using the OBJECT rule, but ensures the data is not migrated to the target instance. Note that the UDF list utility will show which values were added with this parameter.

Note: All object rules must reside in the global area, not in the rules sections within each business area

In summary, the link causes a value entered into the CUSTOMER field in the **Customers** business area to be automatically added to the list named CUST_LIST. This list is available in any other business area and is used in the rule to bring the values associated with the customer into the current screen, when a customer is chosen from its list.

The `{guarantee}` qualifier to the `object` rule allows an issue to be created when the rule is executed, but prevents the value to the title being added as a duplicate to the list. If the value does not exist, it will be created but if the value exists, that will be used.

onchange or refresh	When a field on your layout has a rule that references it, and that the user changes the value of that field, then an onchange rule is triggered. For example, you may have two fields on a layout, named **amount1** and **amount2**. You may have an onchange rule: `total = amount1 + amount2;` When the user changes either **amount1** or **amount2**, the **total** field on the layout will be updated via an Ajax call to the server In previous versions, this functionality was provided by the refresh directive, which also required an HTML modifier on the fields **amount1** and **amount2**. If you are upgrading from ExtraView version 6.0 or a prior release, you should remove the HTML modifier on these fields as they will cause an unnecessary screen refresh
postupdate	Postupdate rules are executed subsequent to a database update on the *add* or *edit* screen
preupdate or database	Preupdate rules are executed prior to a database update. After you press the button to submit an issue to the database on the *add* screen, or you press the Update button on the *edit* screen, these rules are executed. The directives preupdate and database are synonymous. In the future, the term database may be deprecated as it is not fully descriptive of the purpose of the directive

Rule Values

Value Identifiers are one of the following:

- A Data Dictionary Name, e.g. DESCRIPTION or DATE_CREATED
- A Data Dictionary Name, qualified with a Layout Name, e.g. ADD_PROBLEM.DESCRIPTION
- A User ID, e.g. BILL.SMITH
- A User Role, e.g. ADMIN or QA
- A Special Value as defined on this page.

These must be upper case, and must match the internal name of the respective value. Value Identifiers are interpreted first as a Data Dictionary Name, then User ID, and then User Role unless an IDENT: is specified. Rule values are expressions of the form:

- 'Value' – Values can be quoted or left unquoted if their interpretation is unambiguous – i.e. there are no embedded spaces, etc. If this is a lookup field (typically this is a field with a display type of LIST, TAB, or POPUP) this will be treated as the Display title and verified against the database
- Special values are what their name implies: a placeholder for a value the system will provide. These are values, such as USER or SYSDATE. See this page for a full list of special values
- DD_NAME – A Data Dictionary name may be used to reference the value associated with that data element, such as **ASSIGNED_TO**. Data Dictionary names may be used on either side of the operator. Data dictionary names should always be expressed in uppercase.
- (link name).DD_NAME – A link name may be specified before a DD_NAME to reference the value contained in a related issue. The link name must be specified in a Link Directive
- Qualifiers - Complex identifiers consist of an optional link expression, a DD_NAME followed by one or more qualifiers. Qualifiers are properties, such as {is external}, or conditions, such as {changed}. They may also be complex conditions, as {changed_from:'value'}, where 'value' is interpreted as a simple value. See this page for a list of qualifiers
- IDENT:'Value' – This is a special form, used to force evaluation of an identifier as a specific type. For example, this can be used to force evaluation of a value as a User ID if there happens to be a Data Dictionary entry with the same value. Use one of the following forms:
  - EMAIL to specify an email address
  - USER to specify a User ID
  - ROLE to specify a User Role

Comparison of Field Values

The internal and external representations of fields vary across their display type. When fields within rule expressions are compared (using relational operators such as "<", ">", "!=", or "="), the field values are coerced into objects that are appropriate or capable of comparison. The coercions used in rules are defined in the following:

- LIST – LIST : display titles in the default locale are compared as strings. Includes aliased lists
- LIST – TEXTFIELD: display title of list value in default locale is compared to the TEXTFIELD
- LIST – DATE: the display title of the list value is parsed as a date and compared to DATE field as Calendar object
- LIST – DAY: same as LIST – DATE
- LIST – NUMBER: the display title of the list value is parsed as a double (floating point) and compared to the NUMBER field as a Double object
- LIST – DECIMAL: the display title of the list value is parsed as a decimal and compared to DECIMAL field as a BigDecimal object
- LIST – CURRENCY: the display title of the list value is parsed as a currency and compared to CURRENCY field as a BigDecimal object
- LIST – USER: the display title of the list value is compared to the display name of the USER field as Strings
- TEXTFIELD – TEXTFIELD: the text field values are compared as Strings
- TEXTFIELD – DATE: the text field value is parsed as a date and compared to the DATE field as a Calendar object
- TEXTFIELD – DAY: same as TEXTFIELD – DATE
- TEXTFIELD – NUMBER: the text field value is parsed as a double and compared to the NUMBER field as a Double object
- TEXTFIELD – DECIMAL: the text field value is parsed as a decimal and compared to the DECIMAL field as a BigDecimal object
- TEXTFIELD – CURRENCY: the text field value is parsed as a currency and compared to the CURRENCY field as a BigDecimal object
- TEXTFIELD – USER: the text field value is compared to the display name of the USER field as Strings
- DATE – DATE: the DATE fields are compared as Calendar objects
- DATE – DAY: same as DATE – DATE
- DATE – NUMBER: unequal
- DATE – DECIMAL: unequal
- DATE – CURRENCY: unequal

- DATE – USER: unequal
- DAY – DAY: same as DATE – DATE
- DAY – NUMBER: unequal
- DAY – DECIMAL: unequal
- DAY – CURRENCY: unequal
- DAY – USER: unequal
- NUMBER – NUMBER: compared as double (floating point) numbers
- NUMBER – DECIMAL: the DECIMAL field value is converted to BigDecimal, and then Double, then the fields are compared as double (floating point) numbers
- NUMBER – CURRENCY: same as NUMBER-DECIMAL
- NUMBER – USER: unequal
- DECIMAL – DECIMAL: the DECIMAL field values are compared as BigDecimal objects
- DECIMAL – CURRENCY: same as DECIMAL-DECIMAL
- DECIMAL – USER: unequal
- CURRENCY – CURRENCY: same as DECIMAL-DECIMAL
- CURRENCY – USER: unequal
- USER – USER: the user display names are compared

Special Values

Special values are used within rules to reference values outside the issue, or to set specific values to aid in manipulating the issue.

EMAIL_IGNORE_GROUP	This allows you to check or uncheck the checkbox that sends notification with the EMAIL_CUSTOMER field on the *add* and the *edit* screens. Normally this is the **Guest** or **Customer** Role. Set to a value of 'Y' to set the value to checked, or to a value of 'N' to set a value of unchecked. Use a preupdate rule such as:

```
if (SCREEN_NAME='ADD' && STATUS='Fixed') {
  EMAIL_IGNORE_GROUP = 'Y';
}
```

Valid values are 'Y' and 'N'. Note that the value may be in upper or lower case.

GENERATE_EMAIL	This allows you to check or uncheck the checkbox that governs notification on the *add* and the *edit* screens. Use a preupdate rule such as:

```
if (SCREEN_NAME='ADD' && CATEGORY='Undetermined') {
  GENERATE_EMAIL = 'N';
}
```

Valid values are 'Y' and 'N'. Note that the value may be in upper or lower case.

SCREEN_LOAD	There are several occasions when `load` directive rules are executed:

- When an *add* screen loads
- When an *edit* screen loads
- When the user clicks the **Update & Continue** button on the menubar of an *edit* screen. This operation is fulfilled via an Ajax request to the server so no screen refresh occurs.

There are times when you want different rules processing to happen, especially if the **Update & Continue** button is clicked from an *edit* screen. The `SCREEN_NAME` special value allows the rules programmer to differentiate between the *add* and the *edit* screens. The `SCREEN_LOAD` special value allows the rules programmer to differentiate when the **Update & Continue** button is causing the issue to be loaded. The SCREEN_NAME special value may have the value `FULL` or `AJAX`, with usage such as:

```
if (SCREEN_NAME='EDIT' && SCREEN_LOAD='FULL') {
  ... do something ...
};
```

or

```
if (SCREEN_NAME='EDIT' && SCREEN_LOAD='AJAX') {
  ... do something else ...
};
```

SCREEN_NAME	This has four possible values, ADD, EDIT, ADD_CONFIRMATION or MASS_UPDATE. This allows you to define a rule that will only be executed on the Add, Edit, Add Confirmation or mass update screens. For example:
	
	if (SCREEN_NAME='ADD' && STATUS='OPEN') {
	  ....
	}
	
	will execute the rule, only if you are adding an issue and the status is OPEN.

SYSDATE	This provides a value of the current timestamp
SYSDAY	This provides the Date Value of the current time
USER	This references the current user

USER_ROLE	This references or sets the current user role. Setting the user role is only allowed under specific conditions:
	
	  a. This will only work on the *edit* screen, not the *add* screen
	  b. To set a new role to the current user, the current user must be able to adopt the new role. I.e., their user account must specify that they are allowed to adopt the new role as part of the user configuration.
	  c. If the rule attempts to set a role to a role to which the user does not have permission, the role will be changed to the role identified by the behavior setting named LIMITED_USER_ROLE. Thus, you may still affect a role change to a user who has a single role defined, but this will typically be the Guest or Customer role.

Value Qualifiers

Qualifiers are applied to fields stored within the Data Dictionary or special values to return a condition or special value. You apply the value qualifier to the dictionary name of the field. They may be entered as lower or upper case. Some directives may only be used within specific directives. These constraints are listed below in the las column of the table. Valid qualifiers are:

Value Qualifier	Purpose	Used with Directives

{changed}	This has two possible interpretations, both of which return true or false. First, this qualifies a change if a field value has been changed when the rule is triggered. For example, if you have a field named MY_FIELD and the user enters or selects a new value for the field, you can then trigger a rule such as:
	
	if (MY_FIELD.{changed}) {
	  ANOTHER_FIELD = 'value';
	}
	
	The second use is when the *Add another* button on a repeating row has been pressed, and you want to perform an action on the repeating row that is being added. This is a change action that allows
		onchange preupdate

you to manipulate fields, usually to set values in the repeating row being added. An example of this usage, where you have a repeating row named RR_LAYOUT and you want to alter the value of a repeating row field named FIELD_NAME is:

```
if (RR_LAYOUT.{changed} && RR_LAYOUT.{max_row_num} = 3) {
  FIELD_NAME[3] = 'value';
}
```

Note that the value of 3 in the above example means to change the values on the 4th row, as the first row is row zero, the second is row one, etc.

{changed from:*value*}	This returns true if the last value of the field was *'Value'*, else it returns false	onchange preupdate
{changed to:*value*}	This returns true if the field is changed to the new value *'Value'*, else it returns false	onchange preupdate
{clear}	This clears the value in the field	
{company name}	This may only be applied to a field with a display type of *user* and returns the company name of the user referred to	
{contains *value1*[, *value2*, *valuen*, ...]}	Used to determine if one or mores values are selected within a multi-valued list	
{days}	This value qualifier is applied date calculations to determine the number of days between two dates. For example: X_NUMBER = (X_DATE1 - X_DATE2).{days}; will return the number of days as an integer, between X_DATE1 and X_DATE2	
{excludes *value1*[, *value2*, *valuen*, ...]}	Used to determine if one or more values are not selected within a multi-valued list. This qualifier is only used with multi-valued fields	
{exists}	This qualifier is only used within a link directive to check whether a value is present or not present	
{is active}	This may only be applied to a field with a display type of *user* and returns true if the User field references an active user	
{is external}	This may only be applied to a field with a display type of *user* and returns true if the User's company name is different from the site company name, as defined by the behavior setting named COMPANY_NAME	
{is guest}	This may only be applied to a field with a display type of *user* and returns true if the user's only role is the role defined by the behavior setting named LIMITED_USER_ROLE	
{is inactive}	This may only be applied to a field with a display type of *user* and returns true if the User field references a deactivated user	
{is internal}	This may only be applied to a field with a display type of *user* and returns true if the company name of the User field specified matches the site company name defined by the behavior setting named COMPANY_NAME	
{hours}	This value qualifier is applied date calculations to determine the number of hours between two dates. For example: X_NUMBER = (X_DATE1 - X_DATE2).{hours}; will return the number of hours as an integer, between X_DATE1 and X_DATE2	
{like:*value*}	This performs a like comparison on the value of text fields, text area fields and log area fields, providing a return of true or false. Use must use either an asterisk (*) or a question mark (?) as a	

wild card character in the value that you are evaluating. You must use one or more wildcard characters in a single rule. If the string you are searching to match on may be in the middle of a value found, then you should use a wildcard at the beginning as well as at the end of the string.

For example {like:nation*} will find the word **national** but not the word **international**. However, {like:*nation*} will find both **national** and **international**. This value qualifier is case-sensitive.

If you are using the qualifier on a log area display type field, then only the most current entry is used in the comparison.

{match:*value*}	This performs an exact match using a regular expression on the value of text fields, text area fields and log area fields, returning true or false. True is returned only if the entire value of the text matches the value in the rule. This value qualifier is case-sensitive. If you are using the qualifier on a log area display type field, then only the most current entry is used in the comparison.

{max_row_num}   This qualifier is used only with repeating row layouts and is used to provide an action on a row. For example, given a repeating row layout named RR_LAYOUT, and wanting to set the value of a field named FIELD_NAME, you can use the construct:

```
if (RR_LAYOUT.{max_row_num} = 3 && RR_LAYOUT.{changed}) {
  FIELD_NAME[3] = 'value';
}
```

This changes the value of the FIELD_NAME on the fourth row of the layout, when that row is added to screen through the use of the *Add another* button on the repeating row layout. If you reference a row number that does not exist, an error is generated, of the form Repeating row index too large: FIELD_NAME.

onchange
preupdate

This indicates that your rule will not ever execute properly, and you must change the rule to avoid the error. If you set the value of a field with such a rule, and any of the preceding repeating rows do not have a value set, then a null value is set into the value of the field on the preceding row.

{minutes}   This value qualifier is applied date calculations to determine the number of minutes between two dates. For example:

X_NUMBER = (X_DATE1 - X_DATE2).{minutes};

will return the number of minutes as an integer, between X_DATE1 and X_DATE2

{not null}   This returns true if the field is not null

{null}   This is true if the field is null

{old value}   This returns the prior value of the field   onchange
preupdate

{owner}   This qualifier is used to determine whether the field referred to has the same value as the value of the built-in OWNER field

{seconds}   This value qualifier is applied date calculations to determine the number of seconds between two dates.

For example:

X_NUMBER = (X_DATE1 - X_DATE2).{seconds};

will return the number of seconds as an integer, between X_DATE1 and X_DATE2

User Field Qualifiers

Each field in a user's record has a value qualifier that can be accessed with the syntax USER.{*userfield*}. For example, if you want to assign the value of a user's company name to the field name COMP_NAME on screen, you would use this:

COMP_NAME = USER.{company name};

This is a complete list of the user field qualifiers that may be used.

Value Qualifier	Purpose
{area}	The user's default Business Area
{address} {address 1}	The first line of the user's address
{address 2}	The second line of the user's address
{city}	The city within the user's address
{cell phone}	The user's cell phone number
{company name}	The user's company name
{country}	The user's country within their address
{email}	The user's email address
{fax}	The user's fax number
{home phone}	The user's home phone number
{image}	The user's image or photograph. Note that this can be assigned to an image field within an issue
{job title}	The user's job title
{first name}	The user's first name
{language}	The user's language - this is expressed as the two character abbreviation of the country code
{last name}	The user's last name
{pager}	The user's pager number
{project}	The user's default Project within their Business Area
{state} {province}	The user's State or Province
{region}	The user's Region
{signature}	The user's signature. Note that this can be assigned to an image field within an issue
{time zone}	The user's time zone
{user defined nn}	where nn is a number from 1 to 10. These correspond to the user defined fields within the user's record
{work phone}	The user's work phone number
{zip} {postal code}	The user's Zip Code or Postal Code

Actions

Actions are rule statements that may be executed within directives. Not all actions may be included within all directives. For example, it is not valid to perform a REAUTHORIZATION rule within a `load` directive. This action is only valid within an `onchange` or `preupdate` directive. Note that the square brackets in the actions on this page refer to optional syntax.

ADD	{ADD [AS *role*] [NOWAIT]: AREA = *area_1*, PROJECT = *project_1*, *field_1* = *value_1* [, *field_2* = *value_2*] ... ... [, ATTACHMENT[.{selected}]]}
	This will add a new item to the ExtraView database, based upon a set of name value pairs, where the fields are the data dictionary names of the fields, and the values are valid for the display type of the field. This action only works within the `postupdate` directive, as it is essential to perform the underlying update

before you initiate actions that are dependent. As an example, the following will add a new record when a button named ADD_SERIAL_BUTTON is pressed:

```
if (ADD_SERIAL_BTN.{changed} && UNIT_SERIAL.{not null}) {
  { ADD: AREA='Calibration',
   PROJECT='Unit',
   UNIT_SERIAL,
   COMPONENT,
   SITE,
   FACILITY};
 } }
```

Note: You should always provide the parameters and values for the AREA and PROJECT fields with the ADD command. It is indeterminate as to which Business Area and Project into which the issue will be added without these.

The above example demonstrates the use of an implied assignment. When a field name is used in the rule like this:

```
    UNIT_SERIAL,
```

it implies that the rule is executed as:

```
    UNIT_SERIAL = UNIT_SERIAL,
```

If the field that provides the values for the ADD operation is of type Log Area (e.g. the COMMENTS field), then only the last entry from the source issue is added to the new issue.

The as role optional syntax allows the action to be implemented by altering the user's current role to the role, purely for the action. The role is the title of the role, not its name. This is useful when you need to add a record to a different Business Area or Project where the current user may not have sufficient permissions.

The optional NOWAIT keyword provides the facility to execute the ADD action as a background task, not waiting for the result of the action before returning control back to the user. This may be used if you have many ADD actions to perform, and you do not want the user to wait for these to complete before they may carry on working in the user interface. This keyword should be used with caution, as there are no indications to the user if there is a failure from the action. Any failures are reported in the application server log file. Also, note that if you place the action within an `onchange` directive, as opposed to within a `preupdate` or `postupdate` directive, then the values within the issue as they stand at that moment will be added to the new issue, as opposed to the final values when the user has submitted the issue for insertion or updating. The ADD_UPDATE task must be running for this option to work successfully.

The optional ATTACHMENT keyword allows all or selected attachments to be copied from the current issue to the new issue being created. If the ATTACHMENT keyword is further qualified with {selected}, then only attachments that the user has individually selected with the **Select?** checkbox will be copied. The **Select?** checkbox is enabled on the user interface with the security permission keys named PR_ADD_PROBLEM.ATTACH_SELECT and PR_RESOLUTION.ATTACH_SELECT.

The ADD action is not supported in these directives: load, clone, mail

ADD ROW

{ADD ROW: [TO *layout_type_name*] *field_1 = value_1*[, *field_2 = value_2*][, *field3 = sourceField*]] }

This will add a new row to a repeating record, where the square braces indicate optional elements and the parameter lists are the standard, as in the ADD or UPDATE actions. All the destination fields must be defined in the data dictionary as repeating row fields. The values may not be repeating row fields, as there is no certainty which of many repeating rows the value will refer to.

In the same way as with the ADD action, you may make implied assignments.

If the optional TO: layout_type_name is included, then the repeating row with that name is targeted for the addition of the new row. The fields being set must be in the layout that is loaded from the specified layout type, area, project and role of the user that is executing the rule operation.

Occasionally you might want to add a number of blank repeating rows to an issue. However, it is not valid to add completely blank rows to a repeating row. You can, however, use an unused field that is not even on

the layout in question. For example, to add three blank rows to a layout named MY_RR_LAYOUT, you could use this, utilizing the ALT_ID field which is not used on repeating rows and using a text field on the layout:

{ADD ROW: TO *MY_RR_LAYOUT MY_TXT_FIELD = ALT_ID*};
{ADD ROW: TO *MY_RR_LAYOUT MY_TXT_FIELD = ALT_ID*};
{ADD ROW: TO *MY_RR_LAYOUT MY_TXT_FIELD = ALT_ID*};

The ADD_ROW action is not supported in these directives: load, clone, mail

CLEAR

{CLEAR}

This causes the field to be cleared. As an example, to clear the contents of the field named PRIORITY:

PRIORITY.{clear}

COPY

{COPY [AS *role*]: *field_1 = value_1* [, *field_2 = value_2*] ... ... [, ATTACHMENT[.{selected}]]}

This action is similar to the ADD action, except it may be used entirely without the field/value list. Thus {COPY} by itself will create a new issue, with all the values from the original issue. Specifying a name and value allows that value to be used for that specific field name. This action only works within the `postupdate` directive, as it is essential to perform the underlying update before you initiate actions that are dependent.

If you do not specify a parameter and value for the AREA and PROJECT fields, then the issue is copied to the AREA and PROJECT of the current issue. You should not specify the PROJECT field without also specifying a value for the AREA field.

In the same way as with the ADD action, you may make implied assignments.

The optional NOWAIT keyword provides the facility to execute the COPY action as a background task, not waiting for the result of the action before returning control back to the user. This may be used if you have many COPY actions to perform, and you do not want the user to wait for these to complete before they may carry on working in the user interface. This keyword should be used with caution, as there are no indications to the user if there is a failure from the action. Any failures are reported in the application server log file. Also, note that if you place the action within an `onchange` directive, as opposed to within a `preupdate` or `postupdate` directive, then the values within the issue as they stand at that moment will be copied to the new issue, as opposed to the final values when the user has submitted the issue for insertion or updating. The ADD_UPDATE task must be running for this option to work successfully.

The optional ATTACHMENT keyword allows all or selected attachments to be copied from the current issue to the new issue being created. If the ATTACHMENT keyword is further qualified with {selected}, then only attachments that the user has individually selected with the **Select?** checkbox will be copied. The **Select?** checkbox is enabled on the user interface with the security permission key named PR_ADD_PROBLEM.ATTACH_SELECT and PR_RESOLUTION.ATTACH_SELECT.

The COPY action is not supported in these directives: load, clone, mail

DEFAULT

{DEFAULT}

This sets the field to its default value. As an example, to set the contents of the field named PRIORITY to its default value:

PRIORITY.{default}

ERROR

{ERROR: *Message*}

This is only valid in preupdate rules. It will prevent the update from going forward and return an alert to the user with the text Message. Note that several messages may be accumulated and returned in the alert to the

user as all the rules are executed without updating the database. Contrast this with the {STOP: Message} action where execution is stopped immediately.

You may use the standard escape characters in the error message. The most common requirement is to use **n** as a linefeed within your message.

LOG

{LOG: *Message*}

This writes the Message to the system log.

MAIL

{MAIL: '*template_name*', [INCLUDE:ATTACHMENTS] [*user* [, user2, user3, ...] | *field_name* [, field_name2, field_name3, ...]]}

The above syntax uses the predefined ad hoc email template named template_name. The email is sent to one or more users specified by user, or to the user whose USER ID or email address is contained within the fields specified within the field_name list. One or more users or fields may be specified, and you may either use the USER ID or an explicit email address.

The optional INCLUDE:ATTACHMENTS parameter is used to add any selected attachments (via the **Select?** check boxes by each attachment) into the outgoing email so that they will be delivered along with the email.

It is not valid to use this action within the mail directive. It is most typically used within the preupdate or postupdate directives.

Both HTML and text-based email templates may be sent with this action. This is controlled by the format of the template itself, and a simple naming convention for the template. The processing for this action requires either the template_name or the template_title to follow the MAIL: action. If this name or title has a suffix of .txt, then any email generated to a user whose personal email format is full, brief or very brief text will be generated in text. All other mails will be generated as HTML. This implies that if you want email templates where some users will receive HTML mail and others will receive text, you should create two templates. For example, you might have one template named **Customer_Response_Template** with the HTML and another named **Customer_Response_Template.txt** with the text.

When creating text-based email templates, make sure that you use the WYSIWYG (what-you-see-is-what-you-get) mode of the email template utility and be sure that the template does not contain any HTML tags.

In the email rule action, use the HTML template name or title, but without the .txt suffix.

If a text template does not exist, all users will receive the HTML template as a result of the action. If a text template exists, users with text email format preferences will receive the text email template. If an HTML template does not exist and a text template does exist, then all users will receive the text template.

Examples

```
{ MAIL: 'Sales Response Template', BSMITH, GLONG }
```

```
{ MAIL: 'Contact Form', CONTACT_FIELD }
```

The first example emails the Sales Response Template to the users BSMITH and GLONG. The second example sends the template named Contact Form to the user whose email address appears within the CONTACT_FIELD on the form.

REAUTHORIZE

{REAUTHORIZE: *user_field_name* [NODISPLAY] '*message to display*'}

- This special action implements electronic signatures and is only available with the ExtraView *Enterprise* version
- This action is only valid within the onchange and preupdate directive sections of the business rules. However, if it is used within an onchange directive, the issue will be updated immediately after the successful reauthorization. This is to avoid confusion with the user, who might process the electronic signature and believe they have completed their update

- The administrator may set up the reauthentication of the current user at the time the issue is updated. If the re-authentication is successful, the issue is updated and the name of the current user is placed in the field user_field_name. The reauthentication or electronic signature mechanism demands that the user must enter their password before the transaction is allowed to continue. The application will continue to request the user's password until it is entered correctly. The user may cancel out of the signature request, but will not be able to update the record
- The field user_field_name must be placed on the *add* or *edit* screen. The field also requires read and write access for the user roles that will perform the re-authentication. If you want to hide the field from the user's view until the REAUTHORIZE action and associated rules provide values, set a layout cell attribute of the type VISIBLE IF and a value of NOT NULL
- Do not set the value of `user_field_name` user_field_name within the rules that are part of the reauthorization code. The value is set automatically by the action and unpredictable behavior will result if you assign it any value. You can use the rules to set any other field values, such as the timestamp of the event. Of course, you can assign a value `user_field_name` by rules that are not triggered with the `REAUTHORIZE` REAUTHORIZE command
- The `NODISPLAY` option alters the display of the popup. This is an optional within the `REAUTHORIZE` action. When this is not present, the current user's name appears within the popup where the reauthentication is carried out, in read-only format. The user need only enter their password to complete the action. When this is present, the user must enter both their User ID and their password. It is not valid for any other user than the one who is currently signed onto the ExtraView session to complete the reauthentication.
- With most electronic signatures, you will want to record a timestamp of the event. You can set up a field with a display type of **day** or **date** and place it on the form in the same way as the user_field_name. It should also be given a similar layout cell attribute to the user_field_name, of VISIBLE IF NOT NULL
- As an example, create the following business rule in the `preupdate` directive section of the rules:

if (STATUS.{changed to: 'Approved'} && APPROVED_BY.{is null}) {
  {REAUTHORIZE: APPROVED_BY 'By clicking the Submit button you are approving the update of this issue. This is an electronic signature which is recorded in the database. Any appropriate text may be placed here as a warning.'};
}
# if we get here, we have assented to the reauthorization, so set the trigger fields:
if (STATUS.{changed to: 'Approved'} && APPROVED_BY.{is not null}) {
  APPROVAL_DATE = SYSDAY;
}
# clear approval if reopened
if (STATUS.{changed from: 'Approved'} && STATUS != 'Closed') {
  APPROVAL_DATE = "";
  APPROVED_BY = {null};
}
if (STATUS.{changed from: 'Closed'} && STATUS != 'Approved') {
  APPROVAL_DATE = "";
  APPROVED_BY = {null};
}

Assign the APPROVAL_DATE to SYSDAY if the field is of type **day** as opposed to **date**. This will set the **APPROVAL_DATE** to the current date and time when the **APPROVED_BY** field is set, and set it to a null value if the **APPROVED_BY** field is set to null.

The key condition that triggers the reauthorization requires a little explanation. In the above example, the condition is:

if (STATUS.{changed to: 'Approved'} && APPROVED_BY.{is null})

The reauthorization is special, in that the user is not able to exit from the action, until they either successfully enter their password, or cancel the complete action. The first part of the trigger expression (STATUS.{changed to: 'Approved'}) is the change upon which you want to trigger the reuthorization. Once the user enters their password, ExtraView validates this, and then executes the expression again. This is where the APPROVED_BY.{is null} part of the expression comes into play. A successful reauthorization will have set the APPROVED_BY field to a value, usually the value of the current user. Therefore, the reauthorization is not reentered, as the evaluation of APPROVED_BY.{is null} now returns a *false* value

- The message to display in the REAUTHORIZE syntax allows the display of any text on the popup window where the electronic signature is provided. Many organizations have a message approved by their legal departments that is used here. Notice that the message to display may contain HTML, allowing you to format the message. A typical electronic signature popup window will look like the following:

Electronic signature display

- The administrator may set up any number of electronic signature requests, each with a different logical expression that triggers the requirement for the user to enter their signature
- If you implement the REAUTHORIZE rule within a repeating row, then the user field that contains the electronic signature must be a popup user field and the user must complete the electronic signatures in order, starting with the first repeating row
- As explained in the section on Creating & Updating Layouts, you should be careful not to place the user_field_name and the **day** or **date** field on a row within the layout by themselves. With the VISIBLE IF NOT NULL layout cell attributes, the entire row would not be rendered within the browser's DOM, and therefore ExtraView could not alter their values with the REAUTHORIZE rule.

The REAUTHORIZE action is only supported in the `preupdate` directive. As stated above, users have the ability to alter field values within their issues after an `onchange` directive is processed, and the electronic signature would not reflect the values within the issue at the point in time the user provided their signature.

SAVE

{SAVE: }

This rule is valid within `onchange` directives only. It provides functionality equivalent to use of the **Save and Continue** button. When the action is triggered, the issue is saved with its current values, and the user may continue working within the issue to make further changes.

SAVE POINT

{SAVE POINT: }

This rule is valid within `onchange` directives only. It provides a temporary "checkpoint" save of an issue for the current user. Issues are only submitted for permanent saving to the database when the user subsequently clicks on the **Submit** button, or they are idle for a period that amounts to the value stored within the behavior setting named SAVE_CONTINUE_TIMEOUT_SECS. This command is typically used within sites with complex workflow, where users spend a large amount of time processing an issue, and where there is a concern that they may be interrupted by a system failure or that they may leave their computer screen open with unsaved work in progress. Once the SAVE POINT action has been triggered and before the **Submit** button has been clicked, other users may not update the same issue. If they attempt to do so, they will receive an error message.

The command is typically executed within the rules when the user completes a part of the screen and is about to move to a new section. For example:

```
<== onchange ==>
if (FOLD_PRE_DETAILS.{changed}) {
  {SAVE POINT: }
}
```

When the user clicks on the fold field named `FOLD_PRE_DETAILS`, the issue will be saved temporarily. Processing continues normally for the user, and they will need to click the **Submit** button in the normal way to permanently update the issue, or they leave their screen open for the amount of time specified in the bahavior setting named SAVE_CONTINUE_TIMEOUT_SECS. However, if the user closes the screen or there is some failure, and they edit the issue again within the timeframe specified within SAVE_CONTINUE_TIMEOUT_SECS, they regain their place and all field values at the checkpoint where the issue was last saved.

During the time that the issue has been saved at a checkpoint, until the issue is **Submitted** or automatically saved, no other user can edit the issue.

STOP

{STOP: *Message*}

This will interrupt the processing of preupdate rules and returns to the user without executing further rules. This action works differently from the ERROR action in that the ERROR action continues execution and may accumulate more than one message.

You may use the standard escape characters in the error message. The most common requirement is to use **n** as a linefeed within your message.

UPDATE

{UPDATE [AS *role*]: *linkName field_1 = value_1* [, *field_2 = value_2*] ... ... [, ATTACHMENT[. {selected}]]}

This will update one or more records, based upon the definition of the link specification. If the link points to more than one record, all affected records will be updated. This action only works within the `postupdate` directive, as it is essential to perform the underlying update before you initiate actions that are dependent. As an example, the following will update all customer contact records, if the contact is changed:

<== link custCSRInfo ==> AREA='CSR Calls', CUST_LIST
if (AREA = 'Customers' && CUST_CONTACT_PHONE.{changed}) {
  {UPDATE: custCSRInfo CUST_CONTACT, CUST_CONTACT_PHONE}
}

If the field that provides the values for the UPDATE operation is of type Log Area (e.g. the COMMENTS field), then only the last entry from the source issue is added to the new issue.

In the same way as with the `ADD` action, you may make implied assignments.

The as role optional syntax allows the action to be implemented by altering the user's current role to the role, purely for the action. The role is the title of the role, not its name. This is useful when you need to update a record in a different Business Area or Project where the current user may not have sufficient permissions. The optional ATTACHMENT keyword allows all or selected attachments to be copied from the current issue to the new issue being created. If the ATTACHMENT keyword is further qualified with {selected}, then only attachments that the user has individually selected with the **Select?** checkbox will be copied. The **Select?** checkbox is enabled on the user interface with the security permission key named PR_ADD_PROBLEM.ATTACH_SELECT and PR_RESOLUTION.ATTACH_SELECT.

The optional NOWAIT keyword provides the facility to execute the UPDATE action as a background task, not waiting for the result of the action before returning control back to the user. This may be used if you have many UPDATE actions to perform, and you do not want the user to wait for these to complete before they may carry on working in the user interface. This keyword should be used with caution, as there are no indications to the user if there is a failure from the action. Any failures are reported in the application server log file. Also, note that if you place the action within an `onchange` directive, as opposed to within a `preupdate` or `postupdate` directive, then the values within the issue as they stand at that moment will be update to the issue, as opposed to the final values when the user has submitted the issue for insertion or updating. The [ADD_UPDATE task](#) must be running for this option to work successfully.

There is a 5,000 issue limit to the number of linked issues which will be updated with this action.

The UPDATE action is not supported in these directives: load, clone, mail

Permissions

Business Rules ignore security permissions for fields, for all roles. This is similar to developing user custom extensions to ExtraView using Java.

It is assumed that the developer of business rules understands that they have the ability to update the value of any field, without restriction, by using a business rule. Thus, a business rule may make an assignment to a field that is read-only to the user interface, or is not even placed on an *add* or *edit* screen layout.

Writing Rules

* [Add new comment](#)

General Syntax

Rules consist of an action and an optional condition. If a Condition is omitted, the action will always occur; otherwise the action will be taken if all the conditions are true.

Rules will have one of the following syntaxes:

* <Rule Action>
* if (<Rule Condition >) <Rule Action> ; <Rule Action>
* if (<Rule Condition > && <Rule Condition> ...) {
     <Rule Action> ; <Rule Action>;
     <Rule Action> ; <Rule Action>;
     <Rule Action> ; <Rule Action>;
     <Rule Action> ; <Rule Action>;
  }

There is also a special syntax for negative conditions:

if !(<Rule Condition >)

Rules also support the following conditional operators. These conditional operators will work with all numeric and date display field types. They also work as operators using a string compare operator when used with text and list display types.

Operator	Meaning
>	Greater than
<	Less than
<=	Less than or equal to
>=	Greater than or equal to

Note: The condition(s) may extend over multiple lines, and are controlled by the parentheses.

Note: You may not set the value of the fields AREA and PROJECT with a load, refresh or clone directive. Within the rules, all referenced fields must exist in the data dictionary, and must appear on the layout to which the rule applies. You cannot define variables or new fields "on-the-fly".

Assignment

By definition, all fields have a type, specified by its display type in the data dictionary. As with all other computer languages, assignments must be made between fields with the result being of the correct type. For example, you can assign a text field to another text field, but cannot assign a text field to a number field. However, you might assign the a value in a list field to a text field, as the individual values within the list field are of type text.

For the most part, automatic conversions are made between different data types. For example, if you assign a list type field to a text field, the display title of the list value is automatically provided and used. Similarly, if you assign a text field to a list value, then a

reverse lookup is performed on the text value to populate the list field. Of course, if you make an assignment of this last type, then the value in the text field must correspond to an existing value of a list title else an error will be generated.

An important feature of rules within ExtraView is that you can only perform a single operation with a single rule. For example, to add two fields together (FLD_1 and FLD_2) and place the result in a third field (FLD_TOTAL) you must take two steps:

FLD_TOTAL = FLD_1;
FLD_TOTAL += FLD_2;

You cannot and must not use the form:

FLD_TOTAL = FLD_1 + FLD_2;

When applied to a repeating row field, the above syntax updates the field value on all repeating row fields. The following syntax demonstrates how you make an assignment to a repeating row field on a single row. Note that the repeating rows are numbered from zero, not from one.

RR_FIELD[0] = 'value1';
RR_FIELD[1] = 'value2';

Assignments within Search Layouts

There is a special form for assignment that is used within layouts of type SEARCH. Consider a SEARCH layout that is embedded within an ADD_PROBLEM or EDIT_PROBLEM layout, or a Screen usage layout embedded within one of these. This is the only occasion on which fields of the same name may reside within the same overall layout structure.

It can be useful to populate a field or fields within the SEARCH layout with the value of a field in one of the Screen usage layouts. To differentiate the field within the SEARCH layout, from a field in the other layouts, where the same field name may be in use, within an ONCHANGE directive use the form:

SEARCH_LAYOUT_NAME.FIELD_NAME = FIELD_NAME2;

When the FIELD_NAME2 embedded within one of the Screen layouts is set to a value, or its value is changed, the value of the field FIELD_NAME within the SEARCH_LAYOUT_NAME is set to the value of FIELD_NAME2.

"IF" statements

IF statements allow the conditional processing of statements. For example, consider the following rule:

if (PROJECT = 'Customer Tickets') {
  ASSIGNED_TO = 'BSMITH';
}

This will work equally well as a load rule, an onchange rule or a preupdate rule. It will not work as a postupdate rule, as no update will be made once the rule has been read and executed. Next, consider this rule:

if (PROJECT.{changed to:'Customer Tickets'}) {
  ASSIGNED_TO = 'BSMITH';
}

This will only work as an onchange or a preupdate rule, as it is during these times that changes can be detected.

Nested *if* statements, *else / else if*, (), and *or* clauses are not supported at this time. However, you may write statements in a way that achieves your purpose. For example, a statement such as:

```
if (AREA = 'Customer Support') {
  if (STATUS='Submitted') {
    SUB_STATUS = 'Needs Review';
    OWNER = ASSIGNED_TO;
    DUE_BY_DATE = SYSDAY;
  }
} else if (AREA = 'Bugs') {
  STATUS = 'New';
} else {
  STATUS = 'Unassigned';
}
```

IS NOT SUPPORTED It should, and must be, written as:

```
if (AREA='Customer Support' && STATUS='Submitted') {
  SUB_STATUS='Needs Review';
  OWNER=ASSIGNED_TO;
  DUE_BY_DATE = SYSDAY;
}
if (AREA = 'Bugs') {
  STATUS = 'New';
}
if (AREA != 'Customer Support' && STATUS != 'Submitted' && AREA !='Bugs') {
  STATUS = 'Unassigned';
}
```

Recognizing a User's Changes within the ONCHANGE Directive

You must always write a rule that recognizes the changes a user makes when they alter a field value. This is achieved within the onchange directive, and you must consider and use one of three change events.

- changed
- changed from
- changed to

You must have one, and only one, of these qualifiers in a statement. For example, the following rule will work:

if (CATEGORY.{changed to:"Software"} && PRIORITY = "P 1")   { IMPORTANCE = "High" }

whereas the following will not work:

if (CATEGORY.{changed to:"Software"} && PRIORITY.{changed})   { IMPORTANCE = "High" }

The reason is that a user implicitly changesonly one item on the user interface at one moment. The rule will never trigger the change if it looks for two or more changes at the same moment.

Quoting & Comments

Quoting

Literal values may be quoted with either single or double quotes. No escape mechanism is currently provided.

Comments

Comments are lines that begin with a # character. All characters following the # on the line are ignored.

Limitations

- Rules may only use fields that physically store their value in the database or use the special values as specified above. They will not work with computed fields from the data dictionary. For example, you cannot create a rule that uses fields such as **DATE_CLOSED_SINCE, DATE_CREATED_DAY, DATE_CREATED_MONTH, DATE_CREATED_WEEK, DATE_CREATED_TRUNC, DATE_CREATED_YEAR, DATE_LAST_STATUS_CHANGED_SINCE, DAYS_IN_STATUS, DAYS_OPEN, MONTHS_OPEN, WEEKS_OPEN, TIMESTAMP_MONTH, TIMESTAMP_SINCE, TIMESTAMP_TRUNC, TIMESTAMP_WEEK, TIMESTAMP_YEAR, WEEKS_IN_STATUS, WEEKS_OPEN**, or any field with a display type of *Label* or *Custom*.

  However, you can often provide your own expression in a rule to derive the same information. For example, to calculate the number of days since the creation of an issue you may create a field named **MY_DAYS_OPEN** with a display type of **Number** and then construct a rule such as:

  MY_DAYS_OPEN = (SYSDATE - DATE_CREATED).{days};

- This version of the Rules Engine does not support multi-value UDF's on repeating rows.
- Rules cannot be localized with this version. All values should use the title or value of the field in the base locale, as opposed to a localized value.

Rule Parsing & Recursion

Rule Parsing

Directives are case insensitive.

Conditional statements are case insensitive.

VALUE_IDENTIFIERS are case sensitive, and generally expected to be uppercase.

Values are generally display titles, and are case sensitive. A reverse lookup is done on the value to determine it's internal value if it a lookup type (list, popup, etc.). An exception is for user fields, which are assumed to be ExtraView User Ids, and are case insensitive.

Implied Assignments

When a field name is used in an ADD, ADD_ROW, COPY or UPDATE action with an equal sign and a field name on the right-hand side of the statement, it implies that you want the value of the field from the current issue to be used to assign to the add, copy or update value. This is simply a shorthand convention to save time when developing your rules:

```
SERIAL_NUMBER
```

is equivalent to using:

```
SERIAL_NUMER = SERIAL_NUMBER
```

Recursion

Programmers will be familiar with the topic of recursion. With rules, it is possible to write a rule that causes an update to happen, and when an associated rule fires, it causes the same or a different rule to fire that then causes another update to occur. This can happen in a repeatable cycle with no way for ExtraView to recognize that this is happening. The rules may continue firing on the server with no indication to the user that an endless cycle has started. For this reason, your rules should be examined carefully, especially those which add or update issues with the postupdate directive. If you accidentally set an infinite loop running, you should immediately restart the application server and then change your rule to prevent this happening again.

Debugging Rules

- [Add new comment](#)

There are a number of directives and techniques that help with the debugging of rules. These all rely on the placement of entries into the application server log. These entries may be viewed with the **Admin** --> **System Controls** --> **System Log** --> **View Application Server Log** utility. Note the ability of this screen to be able to refresh the most recent entries. You can leave this window open as a message area while you alter rules and test their results in a different window.

After debugging, it is strongly suggested that you remove, or comment, the debugging directives, as they have a small impact on performance.

Directive	Use
debug	Turns on the debugging. All messages from rules execution are written to the log, until the end of rules execution, or until an <== end ==> directive is encountered
end	Ends all processing of rules and stops debugging messages being sent to the log
info	Causes log entries to verify that rules are being executed. Minimal details are listed
log	Logs rule activity in enough detail for end users to verify rule execution. Shows condition evaluation and assignments until the end of the rules, or until a <== nolog ==> directive is encountered
nolog	Turns off the <== log ==> directive

For example, create these rules:

```
<== load ==>
<== nolog ==>
```

```
if (AREA.{is not null}) SHORT_DESCR = 'Set at NO logging';
<== info ==>
if (AREA.{is not null}) SHORT_DESCR = 'Set at INFO level of logging';
<== log ==>
if (AREA.{is not null}) SHORT_DESCR = 'Set at LOG level of logging';
<== debug ==>
if (AREA.{is not null}) SHORT_DESCR = 'Set at DEBUG level of logging';
<== info ==>
```

The results from this will look something like this:

```
2010-07-20  16:10:34  [  info   ] ExtraView. TP-Processor7 ,122118,TP-Processor7,
>>>Entering service,Display.doAddDisplay,,,sc,0,,,uid,BSMITH,tmem,41,fmem,10,nid
,http://nerdvana.extraview.net/evj ON WS_A:
2010-07-20  16:10:35  [  info   ]  TP-Processor7 ** Rules60: * Executing:
# L0000
if (AREA.{is not null}{not null} ()) {
    SHORT_DESCR = Set at NO logging (Set at NO logging);
}
2010-07-20  16:10:35  [  info   ]  TP-Processor7 ** Rules60: * Executing:
# L0001
if (AREA.{is not null}{not null} ()) {
    SHORT_DESCR = Set at INFO level of logging (Set at INFO level of logging);
}
2010-07-20  16:10:35  [  info   ]  TP-Processor7 ** Rules60: * Executing:
# L0002
if (AREA.{is not null}{not null} ()) {
    SHORT_DESCR = Set at LOG level of logging (Set at LOG level of logging);
}
2010-07-20  16:10:35  [  info   ]  TP-Processor7 ** Rules60: * Executing:
# L0003
if (AREA.{is not null}{not null} ()) {
    SHORT_DESCR = Set at DEBUG level of logging (Set at DEBUG level of logging);
}
2010-07-20  16:10:35  [  info   ] ExtraView. TP-Processor7 ,122118,TP-Processor7,
>>>Leaving  service,Display.doAddDisplay,time,1357,sc,0,cc,46,mc,0,uid,BSMITH,tmem,41,
fmem,12,nid,,http://nerdvana.extraview.net/evj ON WS_A:
```

Note the effects of the different directives.

Order of Operations

There are occasions when you need to be aware of the order of operations of the execution of rules.

The most sensitive of these operations is the use of the mail action to send an email template within the preupdate directive. If your preupdate directive also contains rules that modifies some of the values that are used within the template, then the modified values are used, irrespective of the order of the rules. This is because the email notification is handled as a background process, and all updates to the record are completed before the notification task sends the mail.

Rules & Accordion Folds

Accordion folds are described on the page Adding Accordion Folds to Layouts. You may use rules to set the initial state of an accordion fold to be open or closed, and you may open or close folds using rules to determine the state of the fold..

Examples

- You have created two folds within the data dictionary, named Fold A and Fold B. This implies the fields named FOLD_PRE_A, FOLD_POST_A, FOLD_PRE_B and FOLD_POST_B all exist
- You have placed these fields onto an *add*, *edit* or embedded layout
- The default behavior is that both folds are initialized in a closed state
- The first example shows how you might use a load directive to open Fold A when the screen is first presented to the user

  ```
  <== load ==>
  FOLD_PRE_A = {open};
  ```

- At the point where the user has opened Fold A and has performed data entry within there, we want Fold B to open when they click on the fold, and we want Fold A to close

```
<== onchange ==>
if (FOLD_PRE_B.{changed to:{open}}) {
    FOLD_PRE_A = {closed};
}
```

- Whenever we click on Fold B, and no matter its state, we open another fold

```
<== onchange ==>
if (FOLD_PRE_B.{changed}) {
    FOLD_PRE_A = {open};
}
```

- In this case, we have a more complex `if` statement, and we both open and close other folds when we open Fold B. This example also presumes the existence of `FOLD_PRE_C` and `FOLD_POST_C`

```
<== onchange ==>
if (FOLD_PRE_B.{changed to:{open}} && STATUS = 'Fixed') {
    FOLD_PRE_A = {closed};
    FOLD_PRE_C = {open};
}
```

- This last example shows how you might preset a number of folds, based upon the value of another field being changed by the user

```
<== onchange ==>
if (STATUS.{changed}) {
    FOLD_PRE_A = {open};
    FOLD_PRE_B = {open};
    FOLD_PRE_C = {closed};
}
```

Rules & Checkbox Fields

By definition, checkbox fields have two values, *checked* and *unchecked*. However in reality database technology really has three values under the covers:

- *Not Set*
- *Checked*
- *Unchecked*

There are differences in the way that *Not Set* and *Unchecked* are handled internally, but if you adhere to the following way to set and to unset the values of a checkbox, ExtraView will handle the interal differences.

To set a value:

`MY_CHECKBOX = 'Y';`

To uncheck a value:

`MY_CHECKBOX = 'N';`

To check whether a checkbox is set:

`if (MY_CHECKBOX = 'Y') { ..... };`

To check whether a checkbox is not checked:

`if (MY_CHECKBOX = 'Y') { ..... };`

Rules & Date Calculations

Date calculations where you want to compute the difference between two dates, giving a numeric result are an exception to the above syntax. This is because you desire a numeric result and you cannot assign a date to a number without causing an error. The correct syntax to calculate the difference between two dates (DAT_1 and DAT_2) and place the result in the field DIFF is:

DIFF = (DAT1 - DAT2);

Date differences in rules may specify qualifiers to get results computed in units other than days. The qualifiers are {seconds}, {minutes}, {hours}, and {days}. Each of these results in an integral value after the date difference is computed. The fractional part of the computation is truncated, not rounded. This implies that the meaning is "the integral number of between the two dates".

Example:

```
## compute number of seconds between two dates
X_NUMBER = (X_DATE1 - X_DATE2).{seconds};
## compute integral number of days between two dates
X_DECIMAL = (X_DATE1 - X_DATE2).{days};
## compute total (including fractional) days between two dates
X_DECIMAL_DAYS = (X_DATE1 - X_DATE2);
```

You can specify a specific business calendar to use with date calculations. This is done with the calendar directive. For example, if you use the WEEKDAY calendar provided with ExtraView, and you have a calculation such as:

```
<== calendar WEEKDAY ==>
DATE_RESULT  = SYSDATE;
DATE_RESULT += 3;
```

where SYSDATE is a Thursday, the result will be Tuesday, i.e. it is 3 workdays, plus the 2 weekend days.

If you add 0 to a date using a business calendar, the result will skip to the first work day beyond that date if that date is not a work day.

Null date and day field values have special treatment. When a date or day field has a null value, and is being used in a calculation, it is assumed the null date is *January 1st, 1900*. This avoids errors and null results occuring. However, tests on date and day type fields with value qualifiers such as date_field.{is null} and date_field.{not null} return true or false depending on whether there is a null value.

Date fields and Day Fields

Using a Day type field within a Business Rule is consistent when it is compared or combined in expressions with other Day type fields. When compared with Date fields, the Date is converted to a Day value using the user's time zone (to which the Date value is, by definition, relative) and the comparison is done with a Day field to Day field comparison or calculation.

Rules & List Fields

This page refers to all enumerated lists, with the exception of multi-valued lists.

Enumerated lists are fields with display types of *List*, *Popup*, *Radio Button*, *Checkbox, User* and *Tab*.

The three principal abilities of rules with enumerated lists are to set a value in the list, to change a value in the list and to remove the value from the list. Adding a value and changing the value in the list are accomplished with the same assignment:

```
MY_LIST = 'Valid value';
```

The value you assign within the rule must exist, else the rules will not compile.

If you want to remove the value in the list assign it a null value as follows:

```
MY_LIST = '{null}';
```

Rules & Multi-Valued List Fields

Rule that access multi-valued list fields have a number of specific points that you should understand. The examples in this section all assume you have a multi-valued list field with a name of FRUITS, and that valid values within the list are *apple, pear, orange, banana*

and *raspberry*.

Setting Values in a Multi-Valued List Field

For each assignment you make of a value within a multi-valued field, you are selecting that value. For example, if you want to set both *apple* and *orange* you would use:

```
FRUITS = ['apple','orange'];
```

The quotes around the values are optional, unless you have a space character within the value. For example, this is a valid statement:

```
FRUITS = [apple, 'orange', 'honeydew melon'];
```

If you want to set a single value of *pear* within the list you use a simple assignment statement.

```
FRUITS = 'pear';
```

Note that when you assign one or more values to a multi-valued field with an assignment, you are **adding** to the existing list values. For example, if the list contains both *apple* and *orange* and you execute this:

```
FRUITS = ['orange', 'raspberry'];
```

the list will then contain *apple, orange* and *raspberry.*

Adding a Value to an Existing Multi-Valued List Field

```
FRUITS += 'strawberry';
```

Clearing All Values from a Multi-Valued List Field

Simply set the the list as follows:

```
FRUITS = '';
```

or

```
FRUITS.{clear};
```

Checking Values Exist within a List

This syntax will find whether the single value of *orange* is currently set within the multi-valued list:

```
if (FRUITS.{contains 'orange'}) { ... };
```

This syntax will find whether the values of *pear*, *orange* and *banana* are currently set within the multi-valued list:

```
if (FRUITS.{contains 'pear', 'orange', 'banana'}) { ... };
```

This syntax will find whether the values of *pear*, and *orange* are currently set within the multi-valued list, but the value of *banana* is not within the multi-valued list:

```
if (FRUITS.{contains 'pear', 'orange'} && FRUITS.{excludes 'banana'}) { ... };
```

Assignments with Multi-Valued Fields

You can assign the selected values in a multi-valued list field to a text, text area, print text, or an HTML Area field. When you make the assignment, the text, text area, print text, or HTML Area field will contain the entire list of selected values, delimited with semi-colons.

Rules & Numeric Fields

Numeric fields are fields with display types of *Number*, *Currency* and *Decimal*.

The following assignment operators are used with numeric fields.

Operator	Meaning
+=	Add
-=	Subtract
*=	Multiplication
/=	Division

Null numeric field values have special treatment. When a numeric field has a null value, and is being used in a calculation, it is assumed the value is zero. This avoids errors and null results occuring. However, tests on numeric type fields with value qualifiers such as numeric_field.{is null} and numeric_field.{not null} return true or false depending on whether there is a null value.

Rules & Repeating Rows

In general, rules are executed on repeating rows, just as they are executed on the main *Add* and *Edit* screens. However there are some important differences.

- The first significant difference is that a rule such as:

  ```
  <== onchange ==>
  RR_FIELD = 'value';
  ```

  is executed for <u>all</u> repeating rows in existence at the time of the onchange event, in the order of the repeating rows. You can take advantage of this to perform an evaluation of all repeating rows in existence, and then set a value of a field in the main *Add* and *Edit* screen based upon what you find. An example of this is that you might have the repeating row field named RELEASE_STATUS with values of *New*, *Open* and *Closed*. There may be any number of repeating rows in existence with any one of these values in each repeating row. The task might be that you want to set the STATUS field on the main *Add* or *Edit* screen, to *New* if any repeating row has the value of *New*, then secondarily set the STATUS field value to *Open* if there is no *New* value on a repeating row, but at least one repeating row has the value of *Open*. Finally, you might then want to set the STATUS value to *Closed* when all the values of the repeating rows are *Closed*. The following rules will achieve this:

  ```
  <== onchange ==>
  # First, push our changed status to the STATUS field, so we can then trigger a reset
  # based on all repeating row fields, not simply the one that changed
  if (RELEASE_STATUS.{changed}) {
    STATUS = RELEASE_STATUS; }
  # Now do the actual roll up by resetting the main status based on whether specific
  # RELEASE_STATUS values exist. Note that the order of these rules is critical
  if (RELEASE_STATUS = 'Closed') {
    STATUS = 'Closed'; }
  if (RELEASE_STATUS = 'Open') {
    STATUS = 'Open'; }
  if (RELEASE_STATUS = 'New') {
    STATUS = 'New'; }
  ```

  The rules are executed in order for all rows in the repeating row set. Therefore the last rule will have the last effect, setting *New* for the STATUS field if any repeating row has the value of *New* in the RELEASE_STATUS field. This progresses back up the list, so that if all repeating rows have the value of *Closed*, then the STATUS value is set to *Closed*

- You can execute a rule when a new row is added via the *Add another* button. For example, the rule:

  ```
  <== onchange ==>
  if (RR_LAYOUT.{changed}) {
    RR_FIELD = 'value';
  }
  ```

  This has the effect of setting the value of the repeating row field named RR_FIELD on all new rows that are added

- You can have more finite control, and only execute rules on specific repeating rows through the use of a rule such as:

  ```
  <== onchange ==>
  ```

```
if (RR_LAYOUT.{changed} && RR_LAYOUT.{max_row_num} = 2) {
  RR_FIELD[2] = 'value';
}
```

Note the use of max_row_num to determine that the rule is executed only for the row being added. In this example, the rule is only triggered when the third repeating row is added, and the value of RR_FIELD is only set on the third row. Remember, the rows are numbered starting at zero, therefore the number 2 represents the 3rd row

Rules & Text Type Fields

Text fields are fields with display types of Text Field, Text Area, HTML Area, and Print Text. The following table shows the operators that can be used when making assignments with text fields.

Operator	Meaning
+=	Prepends text. For "area" display type text fields, the text to be added includes a line feed before and after the text
&=	Postpends text. For "area" display type text fields, the text to be added includes a line feed before and after the text
*=	Prepends text. For "area" type text fields, the text to be added does not include a line feed before or after the text
/=	Postpends text. For "area" type text fields, the text to be added does not include a line feed before or after the text

Business Rule How To Examples

- How do I set the **Assigned To** of the issue to the person updating the issue, if the current value is null?

  if (ASSIGNED_TO={null} && USER.{is internal}) { ASSIGNED_TO = USER; }

- How do I set the value of a field on the *add* screen?

  if (AREA=Issues && PROJECT=Issues && SCREEN_NAME='ADD')
    {FIELD_NAME=VALUE;}

- How do I set the value of a field on the *edit* screen?

  if (AREA=Issues && PROJECT=Issues && SCREEN_NAME=EDIT)
    {FIELD_NAME=VALUE;}

- How do I set the value of a field without any conditions?

  FIELD_NAME=VALUE;

  Note that this construct does not work within an onchange rule

- How do I set the value of a field within an onchange rule?

  if (FIELD_NAME.{changed}) {
    OTHER_FIELD = FIELD_NAME;
  }

- How do I set the value of a user field such as ASSIGNED_TO to the current user's name?

  if (AREA="Contracts" && SCREEN_NAME='ADD') {
    ASSIGNED_TO=USER;
  }

- How do I check to see if the current user is the user assigned to the current issue?

```
if (AREA = 'Issues' && USER = ASSIGNED_TO) {
  IS_MANAGER='Y';
}
```

- How do I add multiple values with one rule to a multi-valued list field?

```
if (AREA = 'Requests' && BUS_UNIT.{CHANGED TO:'ABC'}){
  ASSIGNED_USRS =[PCOLLINS,MGREEN];
}
```

- How do I check to see if a value is selected in a multi-valued list?

```
if (MY_LIST.{contains 'Yellow'}) { COLOR = 'Yellow';}
```

- How do I check to see if several values are selected in a multi-valued list?

```
if (MY_LIST.{contains 'Yellow', 'Blue', 'Orange'}) {
  COLOR = 'Yellow';
}
```

- How do I check to see if a multi-valued list contains some values, but does not contain some other values?

```
if (MY_LIST.{contains 'Yellow', 'Green'} && MY_LIST.{excludes 'Blue', 'Red', 'Purple'}) {
  COLOR = 'Yellow';
}
```

- How do I set a field with a display type of checkbox to be checked?

```
checkbox_fld = 'Y';
```

- How do I set a field with a display type of checkbox to unchecked?

```
checkbox_fld = 'N';
```

- How do I check whether the current user has the *Administrator* role to allow an operation?

```
if (AREA='Bugs' && STATUS.{changed_to:'Closed'} && USER_ROLE='Administrator')
  {COMMENTS='OK!'};
```

- How do I alter a value, if another value has changed?

```
if (STATUS.{changed}) {FIELD_X='Changed'};
```

- How do I set the values of fields to values from the user's personal options?

```
if (AREA='Issues' && SCREEN_NAME='ADD'){
  PHS_COMPANY_NAME = USER.{company name};
  PHS_FIRST_NAME = USER.{first name};
  PHS_LAST_NAME = USER.{last name};
  PHS_JOB_TITLE = USER.{job title};
  PHS_ADDRESS = USER.{address};
  PHS_WORK_PHONE = USER.{work phone};
  PHS_EMAIL = USER.{email};
  PHS_USER_DEFINED_1 = USER.{user defined 1};
}
```

- When the **Status** field first changes its value from *Open* to *Fixed*, how do I set the UDF field titled **Date First Fixed** to the current date:

```
if (STATUS.{changed_to:Fixed} && STATUS.{changed_from:Open} && DATE_FIRST_FIXED.{is null} ) {
  DATE_FIRST_FIXED = SYSDATE;
}
```

- How do I add two values named **UNIT_PRICE** and **TAX** together and put the results in a field named **TOTAL_PRICE**?

```
if (AREA='Bug' && UNIT_PRICE.{not null} && TAX.{not null}){
  TOTAL_PRICE = UNIT_PRICE;   TOTAL_PRICE += TAX;
}
```

- How do I add subtract a value named **DISCOUNT** from a field called PRICE and put the results in a field named **DISC_PRICE**?

```
if (AREA='Issues' && DISCOUNT.{not null} ){
  DISC_PRICE = PRICE;   DISC_PRICE -= DISCOUNT;
}
```

- How do I multiply two values named **HOURS** and **AMOUNT** together and put the results in a field named **PRICE**?

```
if (AREA='Customer Issues' && HOURS.{not null} ){
  PRICE = AMOUNT;   PRICE *= HOURS;
}
```

- How do I divide a value named **TOTAL_COST** by a value of **HOURS_WORKED** and put the results in a field named **PRICE_PER_HOUR**?

```
if (AREA='Issues' && HOURS_WORKED.{not null} ){
  PRICE_PER_HOUR = TOTAL_COST;
  PRICE_PER_HOUR /= HOURS_WORKED;
}
```

- How do I make the **Comments** field required, when the **Status** changes to *Fixed*? At the same time, inform the user of their error if they do not provide a Comment (database only rule):

```
if (STATUS.{changed_to:Fixed} && STATUS.{changed_from:Open} && COMMENTS.{is null} )
  {Error: Comments are required when an issue is marked fixed.};
```

- How do I build a text field from other text fields?

```
if (AREA="Customers" && SCREEN_NAME='ADD') {
  SHORT_DESCR="New Customer - ";
  SHORT_DESCR&=CUST_NAME;
  SHORT_DESCR&=" - ";
  SHORT_DESCR&=CUST_LEGAL_INDENTITY;
}
```

- How do I set the value of a date field to the current date?

```
if (AREA='Issues' && ACT_PERCENT_COMPLETE.{changed}) {
  COMPLETE_AS_OF_DATE = {SYSDATE};
}
```

- How do I increment a date field?

```
NEXT_MAINTENANCE_DATE = MAINTENANCE_COMPLETION_DATE;
NEXT_MAINTENANCE_DATE += MAINTENANCE_INTERVAL;
```

- How do I set the value of a field based upon a button being pressed?

  if (AREA=Reviews && ADD_EXPENSE_BTN.{changed})
   {EXPENSE_ADD='Y';} }

- How do I pop up an error message?

  if (AREA=Issues && SEVERITY_LEVEL=1 && STATUS=Pending)
   {
    ERROR: Invalid status with this severity level
   }

- How do I pop up error messages in different languages?

  if (AREA=Issues && SEVERITY_LEVEL=1 && STATUS=Pending && USER.{language}='fr')
   {
    ERROR: Statut invalide avec ce niveau de gravité
   }
  if (AREA=Issues && SEVERITY_LEVEL=1 && STATUS=Pending && USER.{language}='en')
   {
    ERROR: Invalid status with this severity level
   }

- How do I set the value of a field to the value of a linked issue field?

  if (AREA='Contracts' && CUST_LIST.{not null}) {
   RELATIONSHIP_GROUP_PARENT = (customerLink).ID;
  }

- How do I set the value of the built-in field RELEASE_FOUND within a rule?

  The important factor here is that the RELEASE_FOUND field is dependent upon its parent fied, PRODUCT_NAME. Therefore you must set both values at one time. The RELEASE_FIXED field has the same dependency.

  ```
  PRODUCT_NAME = 'My Product';
  RELEASE_FOUND = 2.3.4;
  ```


- How do I "roll-up" the values in repeating rows, so that the main issue represents a concensus of all the repeating row values? A commonly-defined use case for this rule is a scenario where you have a STATUS field on the main issue, which represents the RELEASE_STATUS values held in mulitple repeating rows where each row may have the same or different values.

  For example, let's say that your STATUS and RELEASE_STATUS fields have four values, **New**, **Open**, **Fixed** and **Closed**. You want the STATUS field to show the value that corresponds to the earliest one in the list that is encountered across al the RELEASE_STATUS values. Thus, if there are three repeating rows, with values of **Open**, **Fixed** and **Fixed**, you want the STATUS field to indicate a "roll-up" value of **Open**. Likewise, if all three values of RELEASE_STATUS are **Open**, then you want the STATUS to also show **Open**. This is achieved with the following onchange rules:

  # First, push our changed status to the STATUS field, so we can then trigger a reset
  # based on all repeating row fields, not simply the one that changed
  if (RELEASE_STATUS.{changed}) {
   STATUS = RELEASE_STATUS;
  }
  # Now do the actual roll up by resetting the main status based on whether specific
  # RELEASE_STATUS values exist. Note that the order of these rules is critical
  if (RELEASE_STATUS.{changed} && RELEASE_STATUS = 'Closed') {
   STATUS = 'Closed';
  }
  if (RELEASE_STATUS.{changed} && RELEASE_STATUS = 'Fixed') {
   STATUS = 'Fixed';
  }
  if (RELEASE_STATUS.{changed} && RELEASE_STATUS = 'Open') {
   STATUS = 'Open';

```
}
if (RELEASE_STATUS.{changed} && RELEASE_STATUS = 'New') {
  STATUS = 'New';
}
```

- How can I add a number of blank repeating rows when I first display an *add* screen? There is no rule action that allows the addition of a blank repeating row. However, you can use the following technique to simulate this, in the following way:
  - Say you want to add 3 blank rows to the repeating row named MY_REPEATING_ROW
  - Create a new text field that is not going to be used on the layout, say MY_HIDDEN_FIELD. This will not have any value associated with it
  - You will use a field on the repeating row layout to store the MY_HIDDEN_FIELD value - remember this is null. Let's say the field on the layout is a text field named MY_RR_FIELD
  - Now use the ADD ROW action to add 3 new rows:

    ```
    if (SCREEN_NAME = 'ADD') {
      {ADD ROW TO MY_REPEATING_ROW MY_RR_FIELD = MY_HIDDEN_FIELD}
      {ADD ROW TO MY_REPEATING_ROW MY_RR_FIELD = MY_HIDDEN_FIELD}
      {ADD ROW TO MY_REPEATING_ROW MY_RR_FIELD = MY_HIDDEN_FIELD}
    }
    ```

- How do I "roll-up" the values in sibling related issues, so that they update the status of their parent issue to **Closed**, when all the statuses of the child issues change to **Closed**. This is a common requirement, where a parent issue reflects the status of several or many child issues. The child issues are in a sibling relationship to the parent. The following example can be expanded to provide a roll-up over many different status values, and the same technique can be used to update any other field on the parent issue, using a common value over many child issues. To achieve this structure, consider the following postupdate directive rules:

  ```
  # Define a link to the parent issue
  <== link myParent ==>
  RG(RG_NAME = relatedChildren, RG_TYPE = PARENTS)
  # Now define a link for the child issues of the same parent - i.e. the SIBLINGS
  <== link openChildren ==>
  RG(RG_NAME = relatedChildren, RG_TYPE = SIBLINGS), STATUS != 'Closed'
  <== postupdate ==>
  # If we have a parent and all its children are Closed - i.e. there are no children that are not closed
  if (STATUS.{changed to: 'Closed'} && (myParent).ID.{is not null} && (openChildren).ID.{is null}) {
    {UPDATE: myParent STATUS = 'Closed'};
  }
  ```

- How do I add a value entered into one field to become a member of a list in another field?

  First define the object to be used to move the value from a text field to a list field:

  ```
  <== object cust ==> AREA='Customers', LINK=CUST_LIST, TITLE=CUSTOMER, PRIVACY=false
  ```

  Now, use the rule in a way that is similar to:

  ```
  if (CUST_LIST.{changed} && AREA='Issues') {
    CUST_CONTACT = (cust).CUST_CONTACT;
    EMAIL_ADDRESS = (cust).EMAIL_ADDRESS;
  }
  ```

- How do I handle multi-valued fields?

  ```
  if (AGENCY.{contains 'Federal'})
    {GROUP = 'Fed';}
  ```

- How do I check if a field has a value of null?

  ```
  if (AREA=Issues && THEME={null})
    { DRAFT=Y }
  ```

- How do I check if a field does not have a value of null?

  ```
  if (AREA=Issues && THEME={not null})
    { DRAFT=N }
  ```

- How do I create an email action that sends a specific email template I have generated, to a user depending upon a condition when I update an issue?

  ```
  <== preupdate ==>
  if (AREA=Issues && STATUS.{changed_to:Pending Approval}) {
    {MAIL:'Issue Approval Request', APPROVER} };
  ```

- How do I create an email directive to send the standard notification email to all the users in a role?

  ```
  <== mail ==>
  if (AREA="Contracts" && STATUS.{changed to:Finished}) {
    ROLE:MANAGER;
  }
  ```

- How do I send standard notification email to the **ORIGINATOR**, all users in a role and to specific users?

  ```
  <== mail ==>
  if (AREA='Trouble Report' && SCREEN_NAME='ADD') {
    ROLE:CE;
    ORIGINATOR;
    JLACEY;
    KMORRIS;
    SCHRISTIANI;
    PREEVES;
    BKISER;
    LMJAMES;
  }
  ```

- How do I add a new record based upon a user pressing a button on a screen?

  ```
  if (ADD_SERIAL_BTN.{changed} && UNIT_SERIAL.{not null}) {
    { ADD: AREA='Calibration',
        PROJECT='Unit',
        UNIT_SERIAL,
        COMPONENT,
        SITE,
        FACILITY
    };
  }
  ```

- How do I update existing records based upon a field on the current screen changing?

  ```
  if (CALIB_DATE.{changed} && CALIB_DATE > (unitLink).CALIB_DATE && AREA='Calibration') {
    {UPDATE: unitLink, CALIB_DATE, UNIT_CAL_EXPIRES };
  }
  ```

- How do I add a related issue record to the database, following the creation of an issue? This example also shows some other useful techniques. This rule would be executed in the <== postupdate ==> section of the rules. Note the use of the "if" condition to prevent recursion after the new record is added and when the rule fires again for the record just added. Next note the technique where the **SHORT_DESCR** of the issue is modified before it is added to the new record. The relationship between the records is created by assigning the value of the ID of the current issue to the field named MY_PARENT_ID, which will exist within the *add* and *edit* screens of the destination Business Area / Project.

  ```
  if (AREA='Helpdesk' && EMPLOYEE_NAME.{not null} && SCREEN_NAME='ADD' && STATUS='New') {
    SHORT_DESCR = 'Install new computer for - ';
  ```

```
   SHORT_DESCR &= EMPLOYEE_NAME;
   {ADD: AREA,
       PROJECT,
       MY_PARENT_ID = ID,
       ASSIGNED_TO = 'BSMITH',
       STATUS = 'Open',
       PRIORITY = 'P 3',
       IT_SELECT_TABS = 'New Employee',
       SHORT_DESCR,
       IT_DATE_REQUESTED,
       IT_APPROVED_BY,
       ORIGINATOR,
       EMPLOYEE_NAME,
       EMPLOYEE_DEPT,
       EMPLOYEE_START_DATE
   };
}
```

- How do I only allow the person in the **ASSIGNED_TO** field have update permission to an issue?

  This is achieved by altering the current role of the user, if that user is not the **ASSIGNED_TO**. Given that the Guest user is typically not allowed to update issues, then changing the role to that of the Guest or the role identified by the behavior setting LIMITED_USER_ROLE, will cause the record to appear in read-only mode if the user is not the **ASSIGNED_TO**.

  if (USER != ASSIGNED_TO) USER_ROLE = Guest;

- How can I use a link in a relationship group to detect no Open sub-issues?

  <= link rgOpen ==> RG(RG_NAME=MY_GROUP, RG_TYPE=CHILDREN), STATUS != Closed
  if (STATUS.{changed to: Closed} && (rgOpen).ID.{is not null})
    {ERROR: Cannot set the status to Closed when there are open child issues.};

- How can I calculate one number as a percentage of another, and display the result with two decimal points?

  If you simply have two rules that state the following:

  TOT_PERCENT_DEFECT = DEFECTS;
  TOT_PERCENT_DEFECT /= SAMPLE_SIZE;

  where the value for DEFECTS is 25, and the value for SAMPLE_SIZE is 100, the result returned is .2, not .25. The reason for this result is that rules try to make sense of operations, especially division, and therefore avoids results like 1.333333333336, and returns 1.3 instead. The rules engine looks at the number of decimal places (the precision) of each of the values, and then limits the number of decimal places returned to no more than the number of decimal places used to express the value, plus the minimum additional needed to get a non-zero value. Generally this works well, but as seen in this case there can be display problems, not the least of which is that there is no way to guarantee that either of the numbers has a certain number of decimal places.

  To avoid this, write the rules as follows:

  TOT_PERCENT_DEFECT = DEFECTS;
  # multiply by 100 first to preserve accuracy
  TOT_PERCENT_DEFECT *=100;
  TOT_PERCENT_DEFECT /= SAMPLE_SIZE;
  # explicitly add '.00' to preserve 2 digits of precision in the result
  TOT_PERCENT_DEFECT /= 100.00;

  By first multiplying by 100, you guarantee that there will be an integer value, and then by specifying the .00 when scaling it back causes the rules to keep at least two digits of precision

- How can I test an incoming email, uploaded with the EVMail utility to see if it is a new or an existing issue?

```
if (SHORT_DESCR.{like : 'RE: Extraview [*]*'} )
  { STATUS = "Response received";}
```

This rule processes the issue and looks to see if the incoming issue matches the string 'RE: Extraview [*]*' (excluding the outermost quotations. The * is a wildcard. If the SHORT_DESC field in the issue contains a value similar to 'RE: Extraview [#223456] This is a report of a problem.' then the STATUS of the issue is set to the value of 'Response received'

- How can I set the Assigned To field of an issue to the value of the owner of a specific user defined field?

```
if (CUST_LIST.{changed}) ASSIGNED_TO = CUST_LIST.{owner};
```

This onchange rule will look for the owner of the user defined field named CUST_LIST and set this user's ID into the issue as the Assigned To

- How can I ensure that a user enters a date that is in the future, not the past?

First, create a user defined field in the data dictionary. This is used to hold the number of days different between a date (or day) field, and the current date. The field does not need to be on any layout, but it should have read and write permission. For this example, the field is DIFF_DAYS. The following preupdate rule will check that the date is not in the past.

```
if (PROJECT = 'Action' && DATE_FOLLOW_UP.{not null})   {
  DIFF_DAYS = (DATE_FOLLOW_UP - SYSDAY);
}
if (PROJECT = 'Action' && DATE_FOLLOW_UP.{not null} && DIFF_DAYS < 0)
  { STOP: The Follow Up Date must be greater than todays date };
```

- How can I add new repeating rows and populate them with values?

The following onchange rule will populate the first blank repeating row, and add two new repeating rows, when the user clicks the button named GENERATE_PARTS_LIST_BTN and the value of the field named ASSEMBLY has a value of *Primary*.

```
if (GENERATE_PARTS_LIST_BTN.{changed} && ASSEMBLY='Primary' ) {
  PARTS='Part 1';
  { ADD ROW: PARTS='Part 2'};
  { ADD ROW: PARTS='Part 3'}; }
```

If you want to set additional values on the repeating rows, you can modify the above along these lines:

```
if (GENERATE_PARTS_LIST_BTN.{changed} && ASSEMBLY='Primary' ) {
  PARTS='Part 1', RELEASE_STATUS='New';
  { ADD ROW: PARTS='Part 2', RELEASE_STATUS='New'};
  { ADD ROW: PARTS='Part 3', RELEASE_STATUS='New'};
}
```

- How can I set field values, based on the user's choice, when clicking on a navigation bar item?

The following load rule will work on an **Add** screen only, and will set values on the screen as it is loaded.

```
if (SCREEN_NAME = 'ADD' && MY_ADD_NAV_BUTTON= 'Add Task' ) {
  PRIORITY = 'P 2';
  CATEGORY = 'Task';
}
```

If the user chooses the Add Task navigation bar item, then the PRIORITY and CATEGORY fields are set as shown above.

- How can I add a number of blank repeating rows to an issue?

It is not valid to add completely blank rows to a repeating row. You can, however, use an unused field that is not even on the layout in question. For example, to add three blank rows to a layout named MY_RR_LAYOUT, you could use this, utilizing the ALT_ID field which is not used on repeating rows and using a text field on the layout:

{ADD ROW: TO *MY_RR_LAYOUT MY_TXT_FIELD = ALT_ID*};
{ADD ROW: TO *MY_RR_LAYOUT MY_TXT_FIELD = ALT_ID*};
{ADD ROW: TO *MY_RR_LAYOUT MY_TXT_FIELD = ALT_ID*};

- I want to add a user's personal signature to a letter or email I am going to send. How do I accomplish this?

  Users may store their photograph and their signature within their account settings. A common requirement is to place the current user's signature within an HTML Area field, so it may be sent by email, or sent to a PDF to be used as an outgoing letter.This requires the following configuration:

  - Have an HTML Area field within an *add* or *edit* screen

  - Create an image field and place this on the same layout. This is used to hold the user's signature, so it remains as part of the record and part of the audit trail

  - Create a rule similar to:

    ```
    <== load ==>
    MY_IMAGE_FIELD = USER.{signature};
    ```

  - Within the HTML Area field have text such as `$$MY_IMAGE_FIELD$$`. This provides the link from the field containing the signature into the body of the email or letter that you are composing.

Security Permissions

Granting Security Permissions, controls each user role's access to all fields, buttons and features in ExtraView. In setting permissions, the System Administrator has the ability to make these kinds of items read only, write only, readable and writeable, or invisible.

The security system uses the concept of inheritance. Each Area and each Project can have different security permissions for each key. However, if no setting is made at the Area level, then the value of the key at the global level is used. In the same fashion, if no value is given at the Project level, then the key is set to the value at the Area level. This provides an economic means of administering ExtraView where you need only provide settings for security keys that differ from the global level.

When you update a security permission key, or a number of keys, the change in permissions is immediate for the administrative user who made the change. All other users will see the change when the time period defined by the behavior setting named SECURITY_CACHE_MINUTES, causes the security cache for each user to be refreshed. The default time period for this is 30 minutes, so any user signed on will see the change within this time period. Any user who signs on will see immediately, the newly updated permissions.

The security permissions for individual fields may also be set within the data dictionary and within the layout editor. This provides a convenient means of setting the permissions while working on a field.

1. Click on **Grant Security Privileges** within administration.
2. The following screen appears:

Grant Security Privileges screen

3. This screen gives you access to alter the user role permissions for all of the System Security Keys. The options are as follows:
    - Business Area access by user role
    - Projects within Business Area for each user role
    - Navigation bar buttons and other screen options
    - The individual administration menus
    - Fields and other permissions used when adding issues
    - Fields and other permissions used when editing issues and querying/reporting
    - Access to personal and public report types
    - Security features, roles, password and privacy settings
    - Permissions to the values of the STATUS field
    - Fields on users' account records
    - Individual security keys, or a selection of individual security keys
4. After selecting a user role (or all user roles) and a security key category or individual security key, click the **Set Permissions for Selected Keys** button. You can select multiple keys by using the standard combination of CTRL and SHIFT keys on your keyboard, with the mouse button.
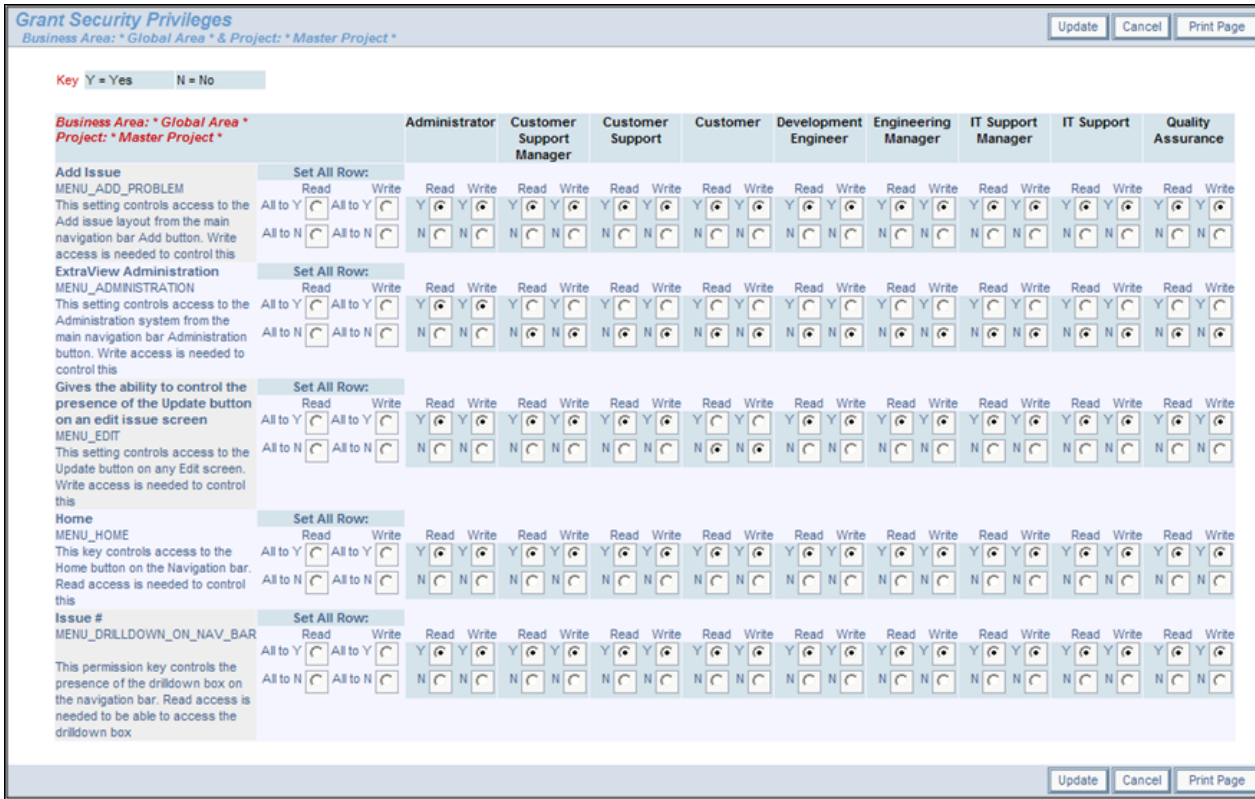
Grant Security Privileges screen

If youwant to modify permission keys at an inherited level, i.e. you have not selected the Global Area and Master Project, the screen is modified , allowing you to see where security permissions for an individual key are inherited, and where they are overwritten. The following screenshot has an example.
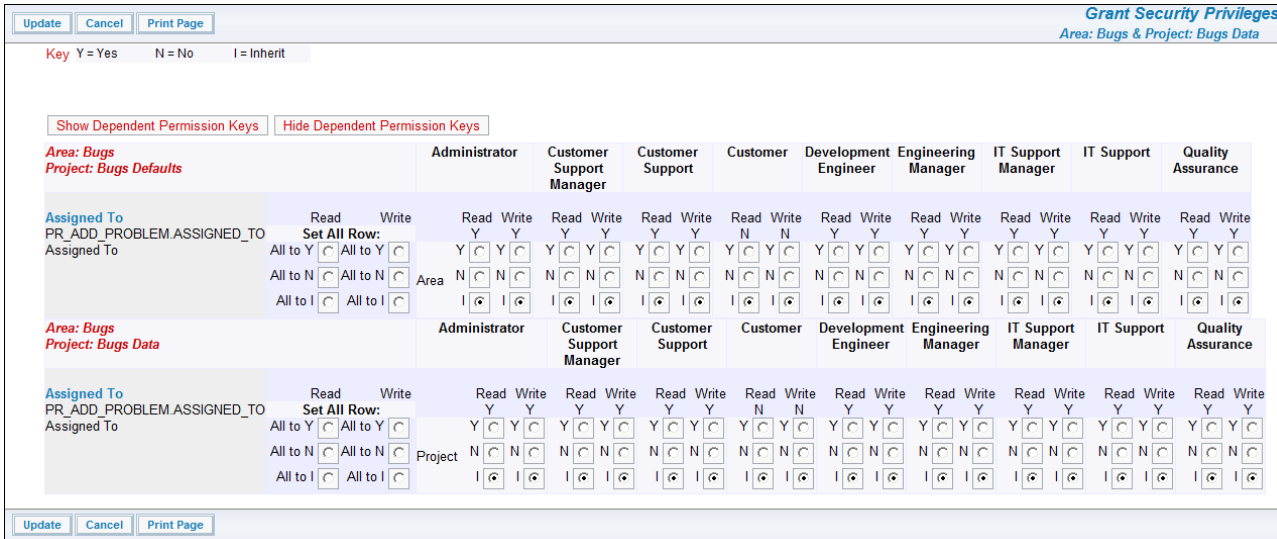
Setting permissions for an inherited field

Make a Field or Option Read Only

1. Identify the column head for the User Role whose privileges you would like to change.
2. Click the radio button in the **Read** box for that User Role only
3. Click the **Update** button.

Make a Field or Option Write Only

1. Identify the column head for the User Role whose privileges you would like to change
2. Put a check mark in the **Write** box, for that User Role only
3. Click the **Update** button.

Make a Field or Option Read and Write for a Particular User Role

1. Identify the column head for the User Role whose privileges you would like to change
2. Put check marks in both of the boxes, for that User Role only (Read and Write)
3. Click the **Update** button.

Make a Field Invisible to a Particular User Role

1. Identify the column head for the User Role whose privileges you would like to change
2. Remove the check marks from both of the boxes for that User Role (Read and Write)
3. Click the **Update** button.

Note that you may alter all the permissions for a single field or object, across all user roles, by using the radio buttons beneath **Set All Row:**. Use the buttons to set all read and/or write permissions to **Yes** or **No** for any row of settings.

Inheritance

Permissions set in the global area and master project are inherited and used by all other business areas and projects unless the inheritance is broken at an individual business area or project. By doing this, you can establish a different set of permissions for any field or object at any level within your installation. Setting a different permission for a field within a business area, causes that permission to be used for all the projects within that business area. Setting a different permission within a master project causes all the other projects within that business area to use these permissions. Setting different permissions within a project other than the master project causes these permissions only to be used within that single project.

This provides an economic means of setting permission through your system. You set them at the root level, i.e. the global business area and master project. These permissions are then used with all junior objects, unless you choose to override them.

You provide the override by editing the field or object within the business area and project where you want to provide the override. When you first look at the permissions within a specific business area and project, you will see that the permission is set to "I", for inherit. You simply set the read and write privileges to override the inheritance. You can always set the permission bak to the "I" value to restore the inheritance.

Allowed Values

Allowed Value Lists give you the opportunity to have certain field lists and their values be dependent on other values in other list fields; for example, a list of specific *Platforms* may only be displayed if the connected parent Product is first selected. The following Parent ◆ Child relationships would be set up so that OS 9 or 10 platforms would only appear for Mac products, and Red Hat only appear for Linux.

Product	Platform
Macintosh Client	OS 9.x
Macintosh Client	OS 10.x
Linux Client	Red Hat
Windows Client	Windows 98
Windows Client	Windows XP

In the above example, when you select the product named Macintosh Client, the ExtraView Add Issue or Edit Issue screen will refresh, and the field titled *Platform* will have the two values OS 9.x and OS 10.x. This feature can be used both to ensure data entered is valid, and that data entry can be accomplished with a minimum of searching. Allowed Values can be "chained" together, providing a cascading set of values. Common examples of allowed values are:

1. *Product name ◆ Module*
2. *Module ◆ Owner*
3. *Category ◆ Module ◆ Release (example of a cascading allowed value)*
4. *Category ◆ Product name*

You may create chains of any length of allowed values; in concept, think of a grandfather ◆ father ◆ child ◆ grandchild type relationship. With some limitations it is also possible to build multiple allowed value relationships where the same child field has up

to fiveparent fields in the allowed value relationship you are building. In this case, the child field is rendered using the intersection of the allowed values from each of the parents.

Not every combination of fields is allowable to be created as allowed values. For example, it is not possible to create an allowed value with a combination of MODULE_ID or MODULE_NAME and PRODUCT_NAME, as another inbuilt mechanism exists to handle this requirement.

Allowed Value Usages

These are typical usages for different configurations of allowed values, within different places in the ExtraView environment.

Both parent and child fields exist within an *Add* or *Edit* screen	This is the most common usage of allowed values. The child fields are simply filtered according to the parent value chosen. Before a parent field is chosen, the child field has no values
A parent and multiple child fields exist within an *Add* or *Edit* screen	In this instance, each child field is filtered according to the parent field that is chosen. Before a parent field is chosen, the child fields have no values
Both parent and child fields are configured to use the **Reverse Allowed Value** option	It is recommended that a parent only is configured to have unique children, as opposed to a child field being the child of more than one parent. Both parent and child fields are populated initially with all values. If the user selects a child value before selecting a parent, the parent list is filtered to only show the valid parent. If the user selects a parent before a child, the child list is filtered to only show valid children of the chosen parent
The parent field resides on an *Add* or *Edit* screen and the child field exists within a Search layout embedded within the *Add* or *Edit* screen	This allows a search field to be populated with a value dependent upon the selected parent field. The user may still choose a different value from the child list
The parent and child fields both reside within a Search layout embedded within the *Add* or *Edit* screen	Allowed values work in the expected way, where the child fields are simply filtered according to the parent value chosen.

Creating Allowed Values

The administrator creates allowed value definition (the allowed value type) from the **Fields & Layouts** tab on the administration menu, click **Allowed Value Types and Combinations**. The administrator also creates the allowed value combinations from within the same utility. The administrator can configure list type fields such that end users may add new values to a list (the **ADD_NEW** layout cell attribute). See the section at the end of this page regarding the configuration of allowed values when an end user adds a value to a list field. The following screen appears when the administrator uses the **Allowed Value Types and Combinations** utility:

Allowed Value Types screen

To create a parent-child dependency:

1. The Title can be any text that describes your Allowed Value Type. The title is not used in any processing and is purely a label
2. For the Parent field, select the field that is the parent in the relationship. This fields is required
3. Select the Child field that is dependent on the parent value you selected. This fields is required
4. You may enable and disable the allowed value using the checkbox with the title of **Enabled**
5. If you check the box against **Warn end users about stale values** then end users are warned when they edit a saved issue and a parent or child value in an allowed value relationship has been deactivated since the record was saved. The end user is informed that they can continue, but if they alter the saved value, they must choose a current valid value or leave the stale value. If this box is not checked, then the user is not issued the warning but the same rule applies. If the end user chooses a new value, it must be valid.
6. Reverse allowed values are a special case for allowed values. These allow the end user to not only choose a valid child after selecting a parent value but also allow the end user to choose a child value and to then see the valid parents. You will typically only use this in a scenario where each parent has unique children and the children are not shared across more than one parent. You must set the behavior setting named AV_NULL_PARENT_IS_NULL to a value of NO when using reverse allowed values
7. If you are creating an allowed value where the Parent field is either AREA or PROJECT, you should select **Page Refresh** as the refresh type. This is because a page refresh always occurs when a user changes the AREA or PROJECT field on an *Add* or *Edit* screen. If you select a different type of refresh when the parent is AREA or PROJECT, you are adding unnecessary overhead to the processing, which might degrade the performance of your system. Most allowed values should use the **Ajax Refresh**. When you choose a new parent value, the child list is refreshed via a call to the server from the user's browser. The Ajax refresh type also ensures that any business rule or user custom programming call is made when the parent value changes. This is fast and efficient
8. Click the **Add** button.

Note: On occasions, you might want to create an allowed value list where only a subset of the values in the list will appear. For example, you might have retired some of the values in the child list and only want current values to appear. You can achieve this by using the AREA field as the parent, and the list field as the child.

New Allowed Value Type

More about Refresh Methods for Allowed Values

There are three methods that can be used:

1. **Ajax Refresh**. When the parent value is selected, an Ajax call is made to the server to retrieve only the child values. The child list is refreshed, but the remainder of the screen is untouched. This provides a fast means of dealing with large lists, without the need to repaint the entire screen. For most purposes, this method should be used for maximum efficiency.
2. **Page Refresh**. On all changes of the parent value in the allowed value combination, the program returns to the server for the new data in the child list. This is primarily used when the parent field is either AREA or PROJECT. These fields, by definition, require a page refresh, therefore setting this refresh method minimizes the amount of work needed on the server, and delivers the smallest amount of code to the browser.
3. **JavaScript Refresh**. This is deprecated with ExtraView version 9.1, and new allowed value types can no longer be created that use this method, but its use is retained for existing allowed value relationships that had this set in previous versions. You should update the allowed value type to use **Ajax Refresh**. The JavaScript refresh method pre-calculates all the possible combinations of parent and child values and delivers these to the browser. For relatively small lists, this is very efficient. However, if either the parent or child value has a business rule that needs to be triggered when its value changes, then this type of refresh does not have the built-in intelligence to call back to the server to execute any business rule. Ajax Refresh allowed values have proved very efficient and have none of the drawbacks of JavaScript refresh, hence the reason for their deprecation.

The same field may be used as a parent to more than one child field by creating more than one allowed value type. Each of these allowed value types may have any one of three refresh types. When a parent field is changed by a user on an *add* or *edit* screen, only one of the above three methods can be used. Which method chosen is decided as follows:

1. If there are any JavaScript refresh types for the parent, then a JavaScript refresh is done for all child values
2. If there are any Page refresh types for the parent, then a Page refresh is done for all child values
3. If neither of the above was true, an Ajax refresh is performed.

Cascading Allowed Values

It is possible to construct allowed value relationships that "cascade", i.e. you define a parent – child relationship for one pair of fields, and then define a second parent – child relationship where the parent is the child of the first relationship. This forms a grandparent – parent – child relationship between three fields. The data dictionary display type for all the fields must be **List, Popup** or **Tab**. Other display types will not work.

The child field select list will always show only *** None *** as a possible value, until both the grandparent value and the parent values have been selected. If you alter the metadata associating the allowed values, in such a way that editing a record causes a child record

to be invalid, a warning will be displayed by ExtraView, informing the user that the child value is no longer valid. The invalid value is still displayed at this time though. If the user alters the values of the grandparent or parent, the invalid value in the child field will no longer be displayed.

Reverse Allowed Values

It is strongly recommended that you only use this feature when there is a single parent to all the children within a list and the lists are not excessively large. You may use the feature with cascading allowed values, but all the individual allowed value relationships must set the **Enable reverse allowed values** checkbox. When reverse allowed values are enabled, ExtraView initially fills all the fields in the allowed value relationship with all the parent values and all the child values. When a user selects any value in any of the fields in the allowed value relationship, the remaining fields are all filtered, both parents of the selected value and children of the selected value.

For example, if you have an allowed value relationship with a parent and child field, the user may then select a child value and the appropriate parent is then selected. If there is a cascade of three fields, and the user chooses a value in the middle field list, then the value in the parent field is selected. The child values in the third field list are then filtered to only display the valid children for the second field.

When a user selects a parent, then all child fields are filtered as with regular allowed value lists.

If the user selects *** None *** in any list within the allowed value relationship and reverse allowed values are enabled, then all fields in the allowed value list are reset to their initial values.

For reverse allowed values to work, all the fields in the entire chain of allowed values must be on the layout, and must all have read and write permission. This includes using reverse allowed values on a report or Quicklist layout.

Allowed Value Parent Fields with Multiple Child Fields

One parent field may be the parent to multiple child fields. When you set up layouts with this scenario, you may also use JavaScript refresh to maintain the lists for users, provided that there are no more than 10 child fields for the parent. In the very rare and exceptional circumstance that you need more than 10 child fields to a single parent, then server-side refreshes will still work, and with no restriction on the number of child fields to a single parent.

Allowed Value Child Fields with Two or Three Parent Fields

You may create two or three allowed value relationships for the one child field, with different fields as the parent field. When you create this scenario you individually populate the allowed children for each of the parents in their respective allowed value lists. When you include the two or three parent fields and the one child field on the same *add* or *edit* screen, then the children that are visible will be those that have common parents. If a child value only has one of the parent fields as an allowed value, the child will not be displayed and it is not a valid selection. If only one parent and the child exists on the *add* or *edit* screen, then this combination will work correctly as the simpler single-parent allowed value combination. The parent and child fields used for this combination must all be enumerated lists with a display type of List, Popup, Tab, Checkbox, or Radio button. User display types are not supported at this time. Note that you should not use JavaScript refresh for an allowed value with multiple parents.

Allowed Value Additions by End Users

The **ADD_NEW** layout cell attribute controls the ability of an end user being able to add a new value to a list on the *add* or *edit* screens. If a field with this layout cell attribute is the child in an allowed value relationship, there is control over the behavior of adding the new field value as a child of the allowed value relationship. The control is achieved through a behavior setting named AUTO_POPULATE_ALLOWED_VALUES. When this setting is YES, then in a popup from an add or edit screen that adds a new value to a list, and that list is the child within an allowed value relationship, then the new value will be added to the list and will be automatically added to the currently selected parent(s). If the value is NO, then the user will be prompted to select the Business Area, Project and the specific parent values to which the newly created child value will be added within the allowed value relationship.

- If you are adding a child list value from the *add* or *edit* screens, you must have the parent field value accessible on the layout *add* or *edit* layout
- If the child field belongs to more than one parent, the value of AUTO_POPULATE_ALLOWED_VALUES is overriden and always behaves as if it is set to NO
- If the parent field is multi-valued, and multiple values are selected, ExtraView starts with all the selected parent values checked and the user can de-select or select additional values as they wish
- When selecting the business area and project field values, the process starts with the currently selected business area and project. There is a check to see if there are allowed values defined at that level. If not, the process moves up the layout inheritance tree until an allowed value is found. If none are defined, but the allowed value type exists, the allowed value is created in the global area and master project.

Entering Allowed Value Combinations

Once the allowed value type has been created, navigate to **Allowed Value Types** administration menu. The following screen with your metadata appears:

Allowed Value Types

Once you have created the dependency relation above, you can view or specify the details of the relationship. For example you can view or modify which values of *Owner* will appear in a list box based on a selected *Category*. To do this, click the **List** icon next to the applicable Allowed Value Title. Note that the prompts for the Business Area and Project are only displayed if these are enabled in your system.

Allowed Value List

If Business Areas and / or Projects are enabled, you can select any combination from the select lists to view the entire list of values defined for the allowed value relationship. To modify an allowed value list, click on the **Edit** button. If Business Areas and / or Projects are enabled, you can define the values for the global area and global project, or you can define the values such that they only exist within a selected Business Area and Project.
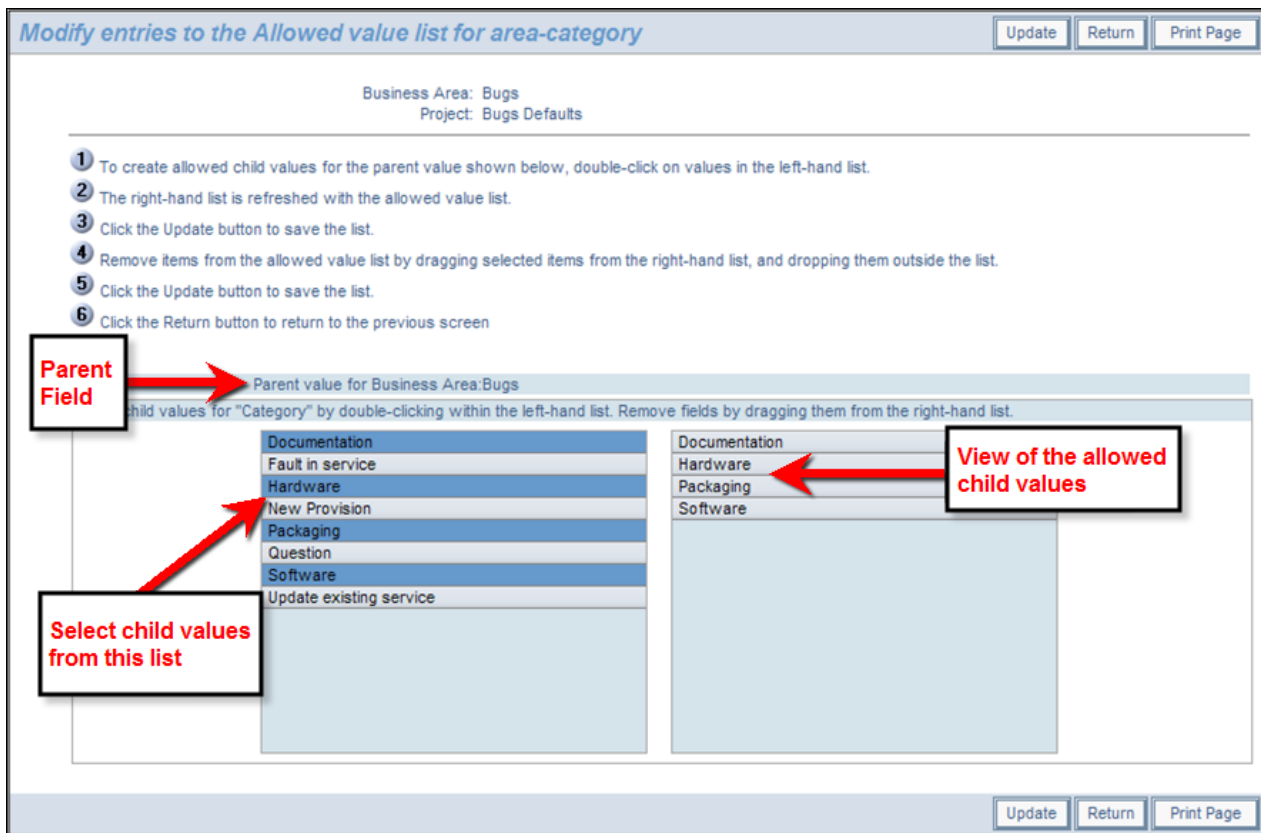
Managing the Allowed Value List

When you modify an allowed value list, the screen will look similar to the example above. You select the allowed child values from the left-hand list, by double-clicking with your mouse on each value you want to allow as a child. The value will then appear in the right-hand list. You remove allowed child values by holding your mouse button down on the list item in the right-hand list, and dragging the value out of the selection box. Note that if the parent field is AREA, then you cannot modify the parent list value from this screen, but you need to return to the previous screen to make a new selection. When you have selected the values as the valid list of children in the left-hand list, values are updated when you press the **Update** button. Note that the selected list displays the child values in the order they will appear in lists on other screens. This order is set in the list management screen for the value, unless you want to use the default alphanumeric order. When you have modified the list, you must press the **Update** button to make your changes in the database. You can select other parents and modify their children before pressing **Return** to return to the previous screen.

Note: If the parent value is either the AREA or PROJECT field, you can only select its value from the previous list of Allowed Values.

Note: You can also maintain the allowed values from the List maintenance screens. When a list field value is a child in an allowed value relationship, you can select the parents to which it belongs from the List maintenance screen. The two methods of maintaining allowed values are complementary. The method described here is more convenient to use if you want to make the entries from the parent field, while the method available from the List maintenance screen is more convenient if you want to make your modifications from the child field. You will typically place both the parent and child values in the allowed value relationship onto an *add, edit*, or *embedded* layout. If you place just one of them on a layout then the allowed values are not rendered; the field on the layout behaves as a normal list. You should always place the parent field on the layout before the child field, as there are occasions when the fields are processed in order, from left to right, top to bottom.

Allowed Value Considerations

Not all fields are valid combinations as parents and children in allowed value relationships. The following is a list of valid combinations:

Parent Display Type	Valid child display types
List	List Pop-up Tab Radio button User
Pop-up	List Pop-up Tab Radio button User

Tab	List Pop-up Tab Radio button User
Radio button	List Pop-up Tab Radio button User
User	List Pop-up Tab Radio button User

Note: Other display types (Date, Log Area, Print Text, Text Area, and Text Field) are not supported in Allowed Value relationships.

Allowed Values with Repeating Rows

There are some restrictions on the use of allowed value relationships with repeating row records. These are as follows –

- JavaScript refresh for the allowed value relationship must be used for the child values to be populated correctly in the child list
- Both the parent and child fields must be UDF's
- The mass update feature within repeating row fields with allowed values right is not supported at this time.

Allowed Values and the Radio Button Display Field Type

Due to limitations within browsers with HTML and JavaScript, it is only permitted to use server refresh on field combinations where the child field is a radio button.

Allowed Values and Different Refresh Types

Allowed values are applied upon each refresh, with the following priorities:

- **Page** refreshes have the highest priority
- **Ajax** refreshes have next highest priority
- **JavaScript** refreshes have lowest priority (these are deprecated with version 9.1)

These priorities are applied, irrespective of what has caused the refresh. The refresh cause can be triggered by the allowed value relationship, by a business rule being triggered, by a layout cell attribute, custom code or some other trigger. If any relationship causes a **Page** refresh, then everythng is refreshed by the **Page** refresh. If you have a chain of allowed values, say **Field_A** ==> **Field_B** ==> **Field_C** and the refresh between the first and second pair of fields is of a different type, then only one refresh type is used during processing, according to which allowed value definition has the highest of the above priorities.

It is strongly recommended that you define and use **Ajax** refreshes as much as posssible and that you do not mix the refresh types within a chain of allowed values. The exception to this is when the parent field in a relationship is either AREA or PROJECT. These fields always imply the use of a **Page** refresh. This will offer the highest performance, except in the case of simple allowed value relationships with a single parent and child where there are a relatively small number of combinations of allowed values. In this case, **JavaScript** refreshes are more efficient.

Allowed Values Types with the Same Parent and Child Fields

You may create multiple allowed value types with the same parent field and child field. However, if you do, only one of the allowed value types may be active at one time. All inactive allowed value types are ignored.

* Default * Parent Allowed Value

Each parent allowed value has an entry named *** Default ***. This allows you to set child allowed values to be used as a default, when the parent has no other allowed values set. This provides a means of economically setting up allowed values where you have a large number of parent values, with many requiring the same set of child values, and only a few parents needing their own specific set of child values.

Set up the allowed values for the parents where you need a specific set of values. Then set up the child values for the *** Default *** parent. All other parent values will now adopt the *** Default *** value children. Also, note that the *** None *** value on the *add* and *edit* screens for a child field will adopt the default children.

Uploading from a File

Note: This option also allows you to create both the parent and the child list values at the same time that you create the allowed value relationships. Both the parent and child values should be user defined fields with a display type of a list form - i.e. List, Popup, Tab or

Radio button. It is sometimes convenient to upload a set of allowed value relationships from an external file. The file to be uploaded is of the form

```
parent_value_1, child_value_1
parent_value_1, child_value_2
parent_value_2, child_value_3
parent_value_2, child_value_4
parent_value_2, child_value_5
parent_value_3, child_value_6
… …
```

The file may be comma-delimited or tab-delimited. Each set of allowed values to be uploaded may only be uploaded into the Business Area and Project that is selected on the screen, shown in this screenshot:

Selecting the Business Area & Project for the Upload

When you press the **Import Allowed Values** button, you see the following dialog:

Uploading the Allowed Value File

Note that you must select the delimiter you have used within the file you are importing. When the file has been successfully uploaded, you will see the following dialog:
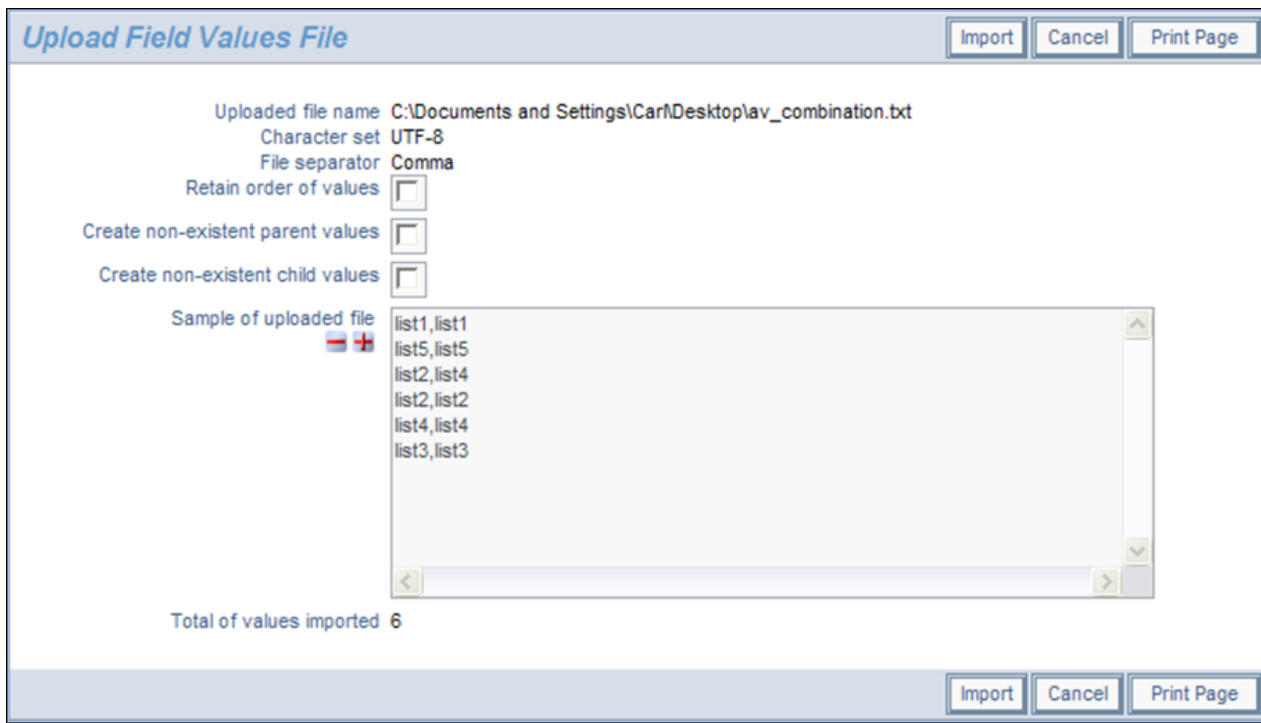
Setting Options for the Upload of Allowed Values

As you can see from this screenshot, you can set the allowable parent and child allowed value relationships within the values in the file. At the same time, you may create parent or child values that do not exist. You may also use the sort order of the values as they exist in the file you are uploading, or you may let ExtraView sort the values alphanumerically.

Workflow Setup

In addition to user and security management, ExtraView allows the System Administrator to manage workflow and processes with many behavior settings, creating rules with a scripting capability and by creating and enforcing status change rules.

Business rules allow the administrator to configure ExtraView with field value assignments based upon the values of any other fields. In a similar way, email rules allow you to control the notification process with a set of rules, as opposed to using the defaults for notification.

Status change rules allow you to determine the valid transitions from status to status. This is completely under the control of the administrator, and can be set individually for each business area, project and user role.

You can also maintain relationship groups as part of your workflow. Relationship groups allow you to connect issues together. When linked, and you apply an update to one of the issues, you can automatically provide a status update and comments to the other issues in the relationship group.

Service Level Agreement management allows the administrator to set up the conditions to measure whether the handling of individual issues fall within predetermined conditions that you have set up to manage your customers. For example, you may have an obligation to respond to all customer issues within two hours, and an obligation to provide closure on an issue within two days. This utility allows you to set up these criteria. Under reporting, your users may use these criteria to prepare and run reports that show your organization's performance against these SLA's.

Status Change Rules

Status change rules allow the Administrator to control the process by which the status of issues can be altered. To conform to your company's workflow, ExtraView can create status change rules based on three different workflow formats:

1. **Default Format**: All users must follow the same rules for all different product, projects, categories, modules etc.
2. **User Role Format**: Different state change rules apply to different user roles within your company.
3. **Product Format**: Certain products can be changed to statuses that are different from those that apply to other products.

Status change rules can be switched off altogether, with the behavior setting named ENFORCE_STATE_CHANGE_RULES.

In addition, if you are using business areas and / or projects, the status change rules can be set independently for each combination. This gives the flexibility of setting status change rules for any combination of Area, Project and User Role, or any combination of Area, Project and Product. Thus the status change rules allow diverse tracking systems to be set up within a single database, each with their own process and workflow.

An important concept with status change rules is inheritance. When you have defined a combination of business areas and projects together with user roles or products, inheritance allows you to define a workflow at any point in the hierarchy, and have all the subordinate points inherit the same workflow. This provides an economic way of defining workflow, without having to provide a workflow for each and every combination of business area, project, user role and / or project that you define.

Note:If you provide even a single entry in the workflow matrix at any level in the hierarchy, then this is used as the workflow for the entire level.

Enabling Status Change Rules

1. From the Administration menu, under the **Workflow** tab, click on **Workflow Settings**.
2. Click the **Edit** button next to the default value entitled ENFORCE_STATE_CHANGE_RULES.
3. Change the value from NO to YES, and click the **Update** button (if the value is already YES then leave it so).

Interaction with the ADMIN_OVERRIDE_ROLE Behavior Setting

Status change rules are not obeyed by a user whose current role is that specified by the behavior setting named ADMIN_OVERRIDE_ROLE. Thus users whose current role is that of an administrator, typically have freedom to alter the status of an issue from any value, to any other value. The audit trail for the issue will still show the transition.

Choosing the Workflow Process

Note: For this next step, you have to decide which workflow is best for your company. You are able to choose between DEFAULT, PRODUCT and USERGROUP (as discussed above).

1. From the **Workflow** tab of Administration, click on **Workflow Settings**.
2. Click the **Edit** button next to the SEPARATE_WORK_FLOW default value.
3. Change the value to DEFAULT, USERGROUP or PRODUCT and press the **Update** button.

Building Status Change Rules

From the Administration menu, under the Site Configuration tab, click on **Status Change Rules**. Based on your choice of workflow format, one of the following three screens will appear (the screens may look slightly different than what is represented here, but the functionality is identical).

Note the use of *** None *** as a value in the To Status and From Status column and row of the table. As well as control the movement of issues between different values, you can also control whether *** None *** is an allowable value. *** None *** may be set as a default value for the status field in the data dictionary.

When accessing the matrix showing the valid status transitions, all checkboxes may appear "grayed-out". This implies that the values shown for the combination of the user role or product, the business area and the project are being inherited and cannot be set explicitly for this combination. To break the inheritance and set explicit values for the combination, either press the Update button once, or use the Copy From option to select the values from another combination of user role or product, business area and project.

Status Change Rules

Update   Return   Print Page

Status change rules can be defined for one Business Area by filling in the matrix.To fill out the matrix, check the boxes of the status changes you want to allow users to make.
Remember that the rules can be switched off or on altogether using the ENFORCE_STATE_CHANGE_RULES setting in the "Workflow Settings" administration screen. The current value of ENFORCE_STATE_CHANGE_RULES is YES.
Remember that the role set by ADMIN_OVERRIDE_ROLE bypasses the status change rules, and workflow is not enforced for this role.

View all ☐ ?   View disallowed ☐ ?   View unpermitted ☐ ?

From Status		To Status						
		New	Open	Fixed/Pending	Closed	Duplicate	Not Found	* None *
	New		☐	☐	☐	☐	☐	☐
	Open	☐		☐	☐	☐	☐	☐
	Fixed/Pending	☐	☐		☐	☐	☐	☐
	Closed	☐	☐	☐		☐	☐	☐
	Duplicate	☐	☐	☐	☐		☐	☐
	Not Found	☐	☐	☐	☐	☐		☐
	* None *	☐	☐	☐	☐	☐	☐	

Update   Return   Print Page

Status change rules applied universally to all issues

Status Change Rules

Update   Return   Print Page

1. Set the values for the status rules to be modified in the list following the 'Define Settings' radio button.
2. Status change rules can be defined for one user role, by selecting the 'Define Settings' radio button and filling in the matrix.To fill out the matrix, check the boxes of the status changes you want to allow users to make.
3. You may press the 'Copy From' radio button and then the Update button to copy from the user role to the current selection in the 'Define Settings' list.
4. Remember that the rules can be switched off or on altogether using the ENFORCE_STATE_CHANGE_RULES setting in the "Workflow Settings" administration screen. The current value of ENFORCE_STATE_CHANGE_RULES is YES.
5. Remember that the role set by ADMIN_OVERRIDE_ROLE bypasses the status change rules, and workflow is not enforced for this role.

◉ **Define Settings**   ○ **Copy From**

User role [Administrator ▾]   User role [Administrator ▾]

View all ☐ ?   View disallowed ☐ ?   View unpermitted ☐ ?

From Status		To Status						
		New	Open	Fixed/Pending	Closed	Duplicate	Not Found	* None *
	New		☐	☐	☐	☐	☐	☐
	Open	☐		☐	☐	☐	☐	☐
	Fixed/Pending	☐	☐		☐	☐	☐	☐
	Closed	☐	☐	☐		☐	☐	☐
	Duplicate	☐	☐	☐	☐		☐	☐
	Not Found	☐	☐	☐	☐	☐		☐
	* None *	☐	☐	☐	☐	☐	☐	

Update   Return   Print Page

Status change rules applied to different user roles

Status Change Rules

Update   Return   Print Page

1. Set the values for the status rules to be modified in the list following the 'Define Settings' radio button.
2. Status change rules can be defined for one Product, by selecting the 'Define Settings' radio button and filling in the matrix. To fill out the matrix, check the boxes of the status changes you want to allow users to make.
3. You may press the 'Copy From' radio button and then the Update button to copy from the Product selected, to the current selection in the 'Define Settings' list.
4. Remember that the rules can be switched off or on altogether using the ENFORCE_STATE_CHANGE_RULES setting in the "Workflow Settings" administration screen. The current value of ENFORCE_STATE_CHANGE_RULES is YES.
5. Remember that the role set by ADMIN_OVERRIDE_ROLE bypasses the status change rules, and workflow is not enforced for this role.

● **Define Settings**   ○ **Copy From**

Product   Tracker   Product   Tracker

View all ☐ ?   View disallowed ☐ ?   View unpermitted ☐ ?

		To Status						
		New	Open	Fixed/Pending	Closed	Duplicate	Not Found	* None *
From Status	New		☐	☐	☐	☐	☐	☐
	Open	☐		☐	☐	☐	☐	☐
	Fixed/Pending	☐	☐		☐	☐	☐	☐
	Closed	☐	☐	☐		☐	☐	☐
	Duplicate	☐	☐	☐	☐		☐	☐
	Not Found	☐	☐	☐	☐	☐		☐
	* None *	☐	☐	☐	☐	☐	☐	

Update   Return   Print Page

Status change rules applied to different products

Open-Ended Rules

Note: The fields **Status** and **Business Area** are subject to the allowed value relationship with a title of **area-status**.

The following table shows which statuses in this workflow do not have any predecessors and do not have any successors. This is displayed for information purposes only.

Values without predecessors	Values without successors
Not Yet Tested	New
Tested - Passed	Not Yet Tested
Tested - Failed	Tested - Passed
Fixed	Tested - Failed
Duplicate	Open
Not Found	
* None *	

Viewing allowed values and other information

Towards the bottom of the screen that manages the status change rules there are two optional sections. If the status change rules are subject to the STATUS field being the child of either AREA or the PROJECT field as a parent, this is displayed. The second piece of information is a table that shows which settings in the status change rules has no predecessors and which rules have no successors. The first group represents the statuses that will be start points in the workflow; the second group represents those that are end-points in the workflow.

Enforcing Status Change Rules

1. For Product or User Role workflows, select a Product or User Role from the dropdown list in the Define Settings list.
2. Start by clicking the check boxes for each *From Status* (left-hand side) to the *To Status* (top). Click on the box at the junction of the *From* and *To* statuses where you want to allow a status transition.
3. After you are satisfied with the *From* and *To* values, click the **Update** button.
4. Select another Product or User Role from the list, and follow steps 1 through 3. Do this for each Product or User Role.
5. Note that you may copy the settings for any Product or User Role from a different Product or User Role. To do this –
   1. Select the Product or User Role you want to define in the Define Settings List
   2. Choose the Product or User Role you want to copy the settings from in the Copy From List
   3. Click on the Copy From radio button

The screen will refresh with the copied values in place.

After your final update, your Status Change Rules are fully implemented. You can now test out the functionality by editing an issue and making status changes. Note that at any moment, you will only see the valid statuses in the list that you can move the status to, according to the present status of the issue.

Enforcing Status Change Rules with Areas and Projects

As discussed above, you are able to use the principle of inheritance to control status change rules over multiple user roles, multiple areas and multiple projects.

Inheriting Status Rules from a different area or project

At any level in the hierarchy of user roles, areas and projects, the default is that the status change rules are inherited from the level above in the hierarchy. There is a fuller explanation of this in the section on inheritance in the Layout Editor section of this guide. The screenshot above shows the view at the global level. As you can see, you cannot inherit from any higher level. When you choose to Define Settings for a level beneath the global area and global project, the section of the form entitled Inherit From will show the available levels from which you can inherit the values and then alter them to suit your needs at the level in Define Settings. Note that you can also copy the settings from a different combination of user role or product, and area and project, to the location set in Define Settings.

Status Change Rule Views

It is possible to create an infinite number of statuses within an ExtraView installation. This can make it difficult to view and set the status change rules on a single screen, as the matrix displayed can grow very large, both horizontally and vertically. Commonly, business areas will use the AREA field as a parent in an allowed value relationship, and the STATUS field as the child in the same relationship.

The Status Change Rules screen will restrict the statuses shown to the statuses that are the children of the current business area. This makes the screen much more manageable.

At the same time, it is possible to use security permission keys in any valid combination of user roles and business areas to turn off the status.

A problem may arise, if you restrict a valid status change transition, then turn off the security permissions, or alter the allowed values for the allowed transition, in that you will no longer see the checkbox for the transition on the screen. To facilitate the management of your metadata under these conditions, the status change rule screen has three checkboxes, which will turn on hidden transitions, allowing you to check for incorrect settings.

View All – This will turn on all the checkboxes, irrespective of permissions, and allowed values

View Disallowed – This will turn on the checkboxes that were omitted due to the allowed value relationship

View Unpermitted – This will turn on the checkboxes that were omitted due to security permission settings.

Status Change Rules Example

This example shows the key elements of setting up the status change rules to form a workflow. This example shows the set up for two of the user roles, the *Development Engineers* and the *Quality Assurance* engineers. In most companies, there will be additional roles that will be configured, and the steps in the workflow (i.e. the statuses) can be composed of many additional steps.

The basic workflow tasks are centered on a defined set of statuses, as set up within the list manager **STATUS** field.

Task	Development Engineers	Quality Assurance Engineers
Create **NEW** issues	✘	✘
OPEN issues to work on them from the NEW status	✘	
Re-**OPEN** issues that were marked as Fixed	✘	✘
CLOSE issues that are marked as Fixed		
OPEN issues that were previously marked Duplicate	✘	✘
Re-**OPEN** an issue that was previously Closed		
Re-**OPEN** an issue marked as Duplicate	✘	✘

No user will be allowed to set the status of *** None *** within the **Status** list, therefore a layout cell attribute of FIELD REMOVE NONE will be set within the Status field on the *Add* and *Edit* Layouts.

Status Rules Example for Development Engineering role

Status Rules Example for Quality Assurance role

With these two diagrams, the Status field will look as follows within the Edit Screen, dependent upon the value of Status as follows. Note that these examples display the status values using the STATUS_TRANSITION field. If you use the STATUS field on your *edit* screen layout, you will see the same values for each role in each status, but in a drop-down select list.

Status	Development Engineers see	Quality Assurance see
New		
Open		
Fixed/ Pending		
Closed		
Duplicate		
Not Found		

Closing Issues

ExtraView maintains an accurate record of the date that each issue is closed. This is achieved by inserting the current system timestamp into the issue, when the issue is moved to the closed state. The closed state is defined by the behavior setting named STATUS_CLOSED_NAME. Its default value is *CLOSED*, but the administrator may change this. The value should only be changed during the implementation of a new system, as once data accumulates in a system, you will lose the historic references to when issues were closed, if you change this setting.

However, your workflow may (or may not) allow closed issues to be re-opened. The rules for setting the DATE_CLOSED date in the database are as follows:

1. The initial value for DATE_CLOSED is *NULL*. As long as the status of the issue is not set to the value of the STATUS_CLOSED_NAME behavior setting, DATE_CLOSED remains equal to *NULL*
2. If the Status is changed to STATUS_CLOSED_NAME, then DATE_CLOSED is assigned the value of the current date
3. If a user has write permissions the security permission key named PR_RESOLUTION.ALLOW_EDIT_CLOSED, and they change the status to a value other than STATUS_CLOSED_NAME, then DATE_CLOSED is reset back to NULL, until the issue is again changed to the status of STATUS_CLOSED_NAME.

Note: The field named DATE_LAST_STATUS_CHANGE always is initially set to the date that the issue is first created, and subsequently is only modified if the status changes value.

Reporting & Querying

This section provides information to the Administrator on reporting options that may, or may not, be granted to users of the system.

Counting Issue Records versus Counting Repeating Row Records

It can sometimes be important to understand the distinction that ExtraView makes between rows on a report and records on a report. This difference in semantics is used to distinguish between the times when ExtraView returns a precise number of *records* on a report and when it returns a set of *rows* that may or may not correspond exactly to the number of records.

The difference comes when a query may return a single record multiple times on a report, or count the same record multiple times on the same report. This happens when there are one-to-many relationships within your data. The two most common times this happens are:

- You prepare a report that uses repeating row records, then use a repeating row field to sort the report. When you use a field on a repeating row record to sort, it will generate a row on the report for each occurrence of the repeating row. Therefore, if you have 3 repeating rows within a single issue, it will generate 3 *rows* on the report for each *record*.
- You use reporting hierarchies. In a similar manner to the above bullet, one record at a level in the hierarchy may have multiple child records, and when you sort by a field at the child level, you will generate one *row* on the report for each *record*.

Report & Query Permissions

There are a variety of security permission keys that give a large degree of control over access to reports. There are two principal groups of permission keys, providing overall control, and then control over individual report types.

Permission Keys - Overall Control

Permission Key	Purpose
SR_ALLOW_ADVANCED_QUERIES	When this setting is YES, the user role can create, edit, save, delete and execute reports composed with the Advanced Query option. When this is set to NO, reports composed with Advanced Query may only be run, and only if the user role has permission to the appropriate SR_PUBLIC_REPORT, SR_PERSONAL_REPORT and SR_USERROLE permission keys
SR_ALLOW_EXPANDED_QUERIES	When the user role has read access to this key, the users can switch between condensed and expanded queries. When this key is set to no access, the users will only be able to create queries with the condensed mode and will not be able to switch into the expanded mode

SR_ALLOW_REPORTS_ACCESS	This is a master switch for access to the ability for a user role to either create new reports, or to run existing reports. Unless a user role has Read Access to this key, they are not permitted to undertake any reporting activity from the Reports screen
SR_ALLOW_HIERARCHY_REPORTS	Write permission to this key allows the user the ability to create reports that use hierarchical filters
SR_DASHBOARD_ON_HOME_PAGE	This permission key is controlled by user custom code. When this is set for Read permission for a role, then a dashboard report inserted by user custom code will appear on the Home page of the user
SR_FILTER_BUTTON	Grant access to the filter button used to change report filters.
SR_KB_ON_HOME_PAGE	Read permission to this key will allow the user role to view and use the knowledgebase search box on the Home Page
SR_MENUBAR_REPORTS	If the user has permission to this security key, then they will see and be able to run any report from the menubar at the top of the screen
SR_RELATIONSHIP_GROUP	This key controls access to allow the user to group issues on column and Quicklist reports with the Group Issues button. Write access is needed to control this.
SR_REPORT_GROUP	Write permission to this key allows the user role to share reports
SR_REPORT_REPOSITORY_ACCESS	The user role requires write access to write to the repository from the report scheduler. Read access is required to be able to view reports stored within the repository. Write access also provides the ability to view and use the **Manage Repository Folders** feature.
SR_REPORT_SCHEDULE	Write permission to this key allows the user role to use the report scheduler
SR_SET_HOME_PAGE_REPORTS	Read access to this key allows the user to select and set their Home Page reports

Permission Keys - Individual Report Control

Reports fall into two major categories, **Public** reports and **Personal** reports.

Public reports appear on the Query screen underneath the section entitled **Public** reports while **Personal** reports appear in the **My Reports** section. There is a sub-division of **Public** reports whereby reports can be made public across one or more user roles.

For each type of report, there are three separate security permission keys. In total, this allows each report type to have its security permissions set for each role. The keys are:

Personal Key	Public Key	User Role Key
SR_PERSONAL_AGING	SR_PUBLIC_AGING	SR_USERROLE_AGING
SR_PERSONAL_CALENDAR	SR_PUBLIC_CALENDAR	SR_USERROLE_CALENDAR
SR_PERSONAL_CHART	SR_PUBLIC_CHART	SR_USERROLE_CHART
SR_PERSONAL_COLUMN_REPORT	SR_PUBLIC_COLUMN_REPORT	SR_USERROLE_COLUMN_REPORT
SR_PERSONAL_CONTAINER	SR_PUBLIC_CONTAINER	SR_USERROLE_CONTAINER
SR_PERSONAL_DASHBOARD_REPORT	SR_PUBLIC_DASHBOARD_REPORT	SR_USERROLE_DASHBOARD_REPORT
SR_PERSONAL_LINKED_REPORT	SR_PUBLIC_LINKED_REPORT	SR_USERROLE_LINKED_REPORT
SR_PERSONAL_MATRIX	SR_PUBLIC_MATRIX	SR_USERROLE_MATRIX
SR_PERSONAL_PAGE_LAYOUT	SR_PUBLIC_PAGE_LAYOUT	SR_USERROLE_PAGE_LAYOUT
SR_PERSONAL_PLANNING	SR_PUBLIC_PLANNING	SR_USERROLE_PLANNING
SR_PERSONAL_SUMMARY_REPORT	SR_PUBLIC_SUMMARY_REPORT	SR_USERROLE_SUMMARY_REPORT
SR_PERSONAL_TASKBOARD	SR_PUBLIC_TASKBOARD	SR_USERROLE_TASKBOARD

If a user role has read permission, then the reports are visible on the query list, and the user role may run the report. If a user role has write permission, then the user role may save or overwrite reports of the type.

Saving Reports

The section of the window **Visibility of Report** is present if you can save the report for different roles, i.e. you have permission to the SR_USERROLE_xxx permission key. You can save the report for all user roles or you can select multiple roles from the list, and only these roles will have the abilirt to access the report.

Within the **Existing Folders** section, you can select the **My Reports** section to save the report for your personal use. This requires permission to the SR_PERSONAL_xxx permission key. If you want to save the report within an existing personal folder, select that folder.

Within the **Existing Folders** section, you can select the **Public Reports** section to save the report for public use or for use across one or more user roles. This requires permission to the SR_PUBLIC_xxx or SR_USERROLE_xxx permission key. If you want to save the report within an existing public folder, select that folder.

To create a new subfolder within the **My Reports** section, select **My Reports** and then enter the name of the new folder in the bottom field. To create a new subfolder within the **Public Reports** section, select **Public Reports** and then enter the name of the new folder in the bottom field.

Quicklists & Detailed Reports

Quicklist layouts and only Detailed Report layouts can be defined for each Business Area in your system. These reports are defined within the layout editor, in exactly the same way that screen layouts are created and maintained. The Quicklist and Detailed Report that are created at run time are dependent upon the current Business Area and Project of the user. The following security permission key is useful to understand when creating and maintaining the Quicklist and Detailed Report.

Security Key	Purpose
PR_RESOLUTION.QUICKLIST	Controls access by user role to the Quicklist report. Most user roles should be given this privilege.

There are several points that you should be aware of when creating report layouts:

- There is a field in the layout list named VIEW_BUTTON. This allows you to place the drill down button to the detailed report on the screen. Most often used on the Quicklist, and never used on the detailed report itself.
- There is a field in the layout list named HISTORY_BUTTON. This allows you to place the drill down button to the history report on the screen. The user must have read permission for the PR_RESOLUTION.MENU_HISTORY security key in order to see the button on the screen.
- There is a field in the layout list named EDIT_BUTTON. This allows you to place the drill down button to the *edit* issue screen.
- There is a field in the layout list named DELETE_BUTTON. This allows you to place the delete issue function on a report. The user must also have privileges to the PR_RESOLUTION.DELETE_BUTTON security key in order to be able to delete issues.
- You may want to use the Highlight layout element attribute to alter the display color of some fields. For example, you may make critical issues appear in red. If you do this, then the color used is defined in the behavior setting named HIGHLIGHT_COLOR.
- If you have a dependency within one field of a layout with another field on the layout, such as Highlight, then both fields must be on the layout.

Quickedit Mode

The Quickedit mode allows the inline editing of fields within a row or issue of a report. You may place the QUICKEDIT button onto any Quicklist layout, any Detailed Report layout, onto column reports and onto taskboard reports. Quickedit provides a fast and efficient means of updating many records in succession. For example, if you have a weekly status meeting where you review many issues and make changes to details such as priorities and who an issue is assigned to, then the Quickedit mode is very useful.

Column Report Quickedit mode

When a **Quickedit** button is configured on a Column report, a button with the title **Gridedit** is placed on the menubar of the window or panel. This allows the user to place all the issues on the report into Quickedit mode with a single click. The user can update any of the issues that are displayed, but they must update these one at a time when the edits to the issue are complete. It is not practical to allow users to update all the records within a single button, as each update may have dependencies which result in a popup - e.g. to fill in required fields. This has the potential to generate a huge number of popups which users could not manage easily.

Taskboard Quickedit mode

When you press the Quickedit button, you are able to edit the fields that appear on that row of the report. This is often much speedier than going into the edit mode of the issue. When you are in the Quickedit mode, you must exit by either updating the issue, or canceling the update. There are some caveats using the Quickedit function, as follows:

- The user must have write access to a field in order to update its value
- As the administrator, you must have placed the field on the edit layout for the appropriate business area and project in order to update its value. If the field is part of the issue and not on the layout, you may still see it in a read-only mode
- The – character will be displayed in a field, if the field is not an ExtraView built-in field and it has not been placed on the *edit* screen layout. This may happen if the user is sitting in one Business Area, but the record they are trying to edit resides in a different Business Area or Project, and does not have that field
- Calculated fields such as **Days Open** and **Days in Status** will be displayed as read-only
- If a field has a link using the "Display as URL" function, this link is not active when in Quickedit mode
- The user cannot ever edit the Business Area or Project fields in Quickedit mode
- If repeating rows are on the report or layout, the user may edit the existing values on a row, but cannot add a new row or delete an existing row
- All allowed value relationships are maintained. However, if the parent value is not on the report, the user may only select a new child value within the current parent value
- Quickedit mode does not operate within a report displaying the results of a hierarchical report. This is because these reports display multiple issues on a single row
- All business rules will be executed in the same way as if the user were using the full edit mode on the issue
- All issue notifications will be sent out as normal
- If an Alternate Title layout cell attribute is defined for a field on the *edit* layout used when performing a Quickedit, then this overrides the Alternate Title defined (if any) defined on the report

Any HTML modifier placed on a field that appears on a report in Quickedit mode will be ignored. Instead, there is a user custom JavaScript exit that will allow you to define any code required for functionality. This is named ucHtmlModifierQuickEdit() and if it

exists in the UserJavaScript.js file you may place code that emulates the HTML modifiers.

Quickedit on Detailed Report Layouts

There are the following restrictions in using Quickedit on Detailed Report layouts:

- Quickedit is not supported if the Detailed Report contains embedded layouts
- Quickedit is not supported on Detailed Reports if the behavior setting named REPORT_LABELS_POSITION has a value of LEFT. Only when the labels are rendered with REPORT_LABELS_POSITION equal to TOP is Quickedit supported

Keyword Queries & Quickfind

Keyword queries are enabled with the following actions:

- Place the inbuilt field named KEYWORD on a query filter layout
- This is subject to the normal PR_RESOLUTION.KEYWORD security permission key, allowing you to control which roles have access to the facility
- There is a further security permission key named PR_RESOLUTION.SEARCH_ATTACHMENTS. When this is enabled for a user role, they will see a checkbox appear beneath the keyword search. When the user checks this box, attachments to issues will also be searched for the keywords entered.
- ALLOWED_ATTACH_SEARCH_FILE_EXT – this behavior setting controls which file attachment types are searched. For example, you may have large image files within your database, through which there is no point searching for keywords
- ALLOW_SEARCH_TEXT_UDFS – When this is set to YES, all User Defined Fields with a display type of TEXT are included in the keyword query search
- SEARCH_ATTACH_THRESHOLD – This is another control to stop huge searches when using keyword queries on attachments. ExtraView will first calculate the total size of the attachments that are to be scanned for keywords. If the size (in bytes) is greater than the number in this setting, then ExtraView will ask the user to confirm that he wants to execute the search.

ExtraView composes SQL queries to search for keywords in its underlying database by default. However, the Administrator may turn an extremely fast, indexed search mechanism named Quickfind. Quickfind requires additional resources and storage to provide this performance boost, but for large sites with a significant amount of text and attachment files, it significantly improves performance.

Searching Microsoft Documents for Keywords

Microsoft documents, such as Word and Excel, are stored using a character set known as **UTF-16LE Unicode 16-bit LittleEndian**. If you want to search these documents for keywords, you should store the documents, when you upload them, using this as the character encoding. This is especially important for Asian languages. Many searches with Roman alphabets will work fine without correctly identifying the character set within the Microsoft document.

Quickfind

If ExtraView Corporation is hosting your installation, you do not have direct access to the file system of the server to configure, alter or use this feature without contacting ExtraView support.

Quickfind uses indexes built and maintained by ExtraView to speed up the search process for keywords. Understanding these indexes and constructing your search terms properly is critical in generating the expected results from your query. The feature is built upon the Apache Lucene technology.

To enable this feature, see the section on Managing Quickfind and the section on setting up the task that performs the indexing. This is managed in the Manage Tasks and Threads administration utility.

The indexes built are based on words extracted from the data being searched. The database automatically extracts words from your documents, ignoring most special characters.

Since only words are indexed by the database ExtraView will use a more complete, but slower method to search if the keywords that the user is searching for contain special characters. This search can be much slower, especially if the user is searching for attachments. Special characters are any characters other than alphanumeric characters, a-z, A-Z, and 0-9. For example, if the keyword search is for the character string **FIND_ME**, a slower search will be used to ensure that words containing the underscore are found. This is a limitation of the database operation underneath ExtraView, not a limitation of ExtraView.

One of the biggest implications of the word index is that searches for fragments of a word that do not start at the beginning of the word will not return that word. For example, a search for **FORM** can return results for **FORM**, but not **inFORMal**.

ExtraView extends the keyword search term entered to search for words that start with the same characters. For example, a search for **APPLE** can return results for **APPLE** as well as **APPLEs** and **APPLEsauce**.

The default installation of Quickfind into your database environment performs case insensitive matching. Any combination of upper and lower case characters will match the same list of characters with any other combination of capitalization. For example, the following list of words would all be found when searching for **apple: applE, ApPlE**.

You may configure which text fields and file attachments are indexed. The indexing happens as a background task run on a timer within ExtraView. This timer is controlled by a behavior setting on the management screen for Quickfind, with the default being every five minutes. This means that text entered or attachments uploaded are not immediately available for finding with the keyword search capability, but will be included in the results following the next completion of the Quickfind background task. To enter the settings, click on the **Manage Quickfind Settings** of the **Display & Reports** administration section.

Managing Quickfind settings

Note that in addition to the **Estimate Storage Requirements** button, the two other buttons at the bottom of the screen:

Manage Content Types – This allows you to manage the different content types that are indexed by Quickfind. It is not very likely that you will need to modify these settings as the default values provided are fairly extensive.

Manage Character Set Mapping – This utility is only used for Oracle databases. Again it is not likely that you will need to alter the settings provided.

Quickfind versus Standard Search

- Quickfind is faster searching for keywords, compared to the standard search mechanism.
- Quickfind requires significantly more database disk storage than the standard search mechanism
- Text indexed with Quickfind will not be found immediately it is inserted. There will be a delay, up to the time specified in the poll interval of the task, before the text may be found. In most circumstances this is not a significant issue
- The standard search mechanism will not find mixed case words

- All queries using Quickfind are case-insensitive and support the wildcard character (*)
- Quickfind ignores most non-alphabetic characters when indexing the content and your users can search for and retrieve these. The standard search mechanism does not attempt to retrieve words when performing a query, if they contain non-alphabetic characters
- Quickfind indexes numbers as well as alphabetic characters, therefore text fields that contain numbers are indexed. This is not true for standard search.

Reporting Hierarchies

Reporting hierarchies allow the administrator to set up relationships for reporting purposes. These relationships may be more than just a single level with a single parent and single child. You can set up reporting hierarchies of up to ten levels although it is unlikely that use cases go beyond three or four levels.

Some examples of reporting hierarchies may be:

Customers ◆
Customer Issues

A single customer may report many issues, and you may want to display customer information such as their name, contacts, and email addresses on a report along with the details of all the individual issues that were reported by the customer, such as the issue title, status and resolution.

Requirements ◆
Engineering Orders

You may want to set up a business area within ExtraView that has the high-level requirements to build new products. Each of these high-level requirements may be the parent of many individual engineering orders that build the component parts of the new product.

Line of Business
◆ *Action Plans* ◆
Action Items

A company may be split into many lines of business, and each of these lines of business may have a number of high-level action plans, which themselves are broken down into action items. This can be represented by a reporting hierarchy.

The relationships between the parent and child levels of the reporting hierarchies are represented by relationship groups within ExtraView. Please see the section of this guide on how to set up and maintain Relationship Groups, and to see how issues are placed within these relationship groups.

A reporting hierarchy is simply the definition of the structure that makes it possible to create and view reports based upon the relationship groups. For instructions on how to create a report that uses a reporting hierarchy, see the Column Report section in the ExtraView User Guide.

To create a report hierarchy, select the **Reporting Hierarchies** entry on the **Site Configuration** tab of the **Administration** section of ExtraView. This presence of this menu entry is controlled by the security permission keys named CF_HIERARCHY and CF_HIERARCHY_LEVEL. If you grant access to one of these permission keys, you should grant access to both in order to be able to set up the reporting hierarchies. The following screen is displayed when you select the **Reporting Hierarchies** option:
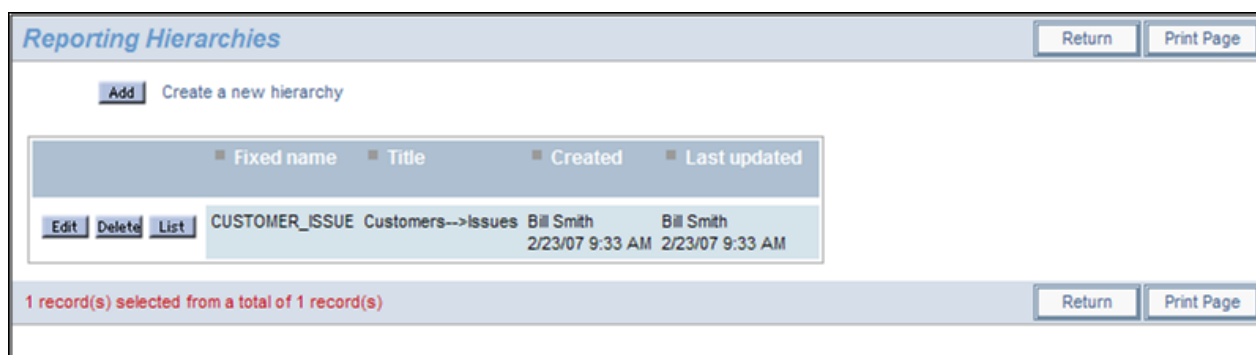
Reporting hierarchies management

From this screen you may use the **Add** button to create new hierarchies, use the **Edit** button to alter the title of an existing reporting hierarchy, use the **Delete** button to eliminate an existing reporting hierarchy, or use the **List** button to manage the hierarchy.

Adding a new reporting hierarchy offers you a simple screen where you enter the fixed name and the title of the hierarchy as seen below:

Adding a new reporting hierarchy

Once you have entered a fixed name and the title, click the **Add new hierarchy button**. To add a new hierarchy level to an existing hierarchy, click on the **List** button from the Reporting Hierarchies Management screen. You will see the existing levels of the hierarchy, and you can add additional levels to this. This screenshot shows an existing reporting hierarchy level of one level:

Single level reporting hierarchy

If you use the **Edit** button to alter the metadata at this level, it will look similar to this:

Editing a hierarchy level

Note from the screenshot that the level is **1** and this cannot be changed. As you add new levels to the hierarchy they will automatically be set at an incrementally higher level. So, the next level you add will be **2**, etc. When adding a new hierarchy level you must choose an existing relationship group to represent the relationship, and you will then choose the relationship type. This must be one of *Children, Parents, Related, Members, Linked* or *Siblings*. These are defined in the section on relationship groups.

The above series of screenshots represent a reporting hierarchy of customer issues, where customers are the parents of issues.

The Business Area and Project in the last screen have special significance. If the behavior setting named DISPLAY_ALL_FIELDS is set to a value of NO, then the following the following behavior restricts the fields visible as filters:

- Set the behavior setting to a specific business area and project that represents an *edit* layout
- The normal rules of inheritance are obeyed if there is no *edit* layout at that business area and project level
- When you create or edit a report using the hierarchy, then only the fields on the *edit* layout and its sublayouts will be used to populate the field list for this level in the hierarchy. The normal security permission rules also apply.

This behavior helps provide clarity on the report editor screen. It is highly unlikely that a user will ever want to select a field at a hierarchy level that is not present on the *edit* screen and this behavior will most probably reduce the number of selectable fields at a

level in the hierarchy from hundreds to tens.

Report Expressions

Report expressions are used to calculate or derive new field values which are only placed on report output. For example, you may want to multiply the values held in two columns of a report to derive a new calculated value. Another example may be to subtract one date from another to calculate the number of days between two events. Report expressions are first defined by creating a field in the data dictionary that is placed on the report. The actual calculation or expression is defined in the default value of the field, or in an attribute that is stored with the field within a report. Using the default value allows you to define a calculated field that may be shared across many reports without further configuration. However, you can use a single Expression type field defined in the data dictionary and reuse this on many reports for many purposes. You achieve this by selecting the field for use on any report, then using its **Alternative Title** and **Expression** attributes to modify its purpose for that report.

Syntax for Report Expressions

The general syntax of an expression matches a subset of that in ANSI-99 SQL expressions, with arithmetic, string, and date data types. This includes infix and prefix operators, with standard precedence rules. Also, parentheses may be used to group sub-expressions.

Note that although the expressions utilize ANSI-99, the different databases upon which ExtraView may be installed do use different syntax for many operations, and the syntax you use within your expressions must be correct for the underlying database. To make matters even more complex, the syntax of some expressions varies across different versions of the same database.

Literals of each type are supported and although full ANSI syntax is not promised, it should be sufficient to allow specification of a specific value for each data type. Variable names may refer to fields in the database. Each variable is in the form **$$DD_NAME$$** and may refer to a standard ExtraView field name or special variable. Usually, the specification of a variable name refers to its report value (viz., its title or rendered form). However, inside an expression, the **value** of the variable is used, not its rendered form. Also, see the note on enumerated types below.

Limitations

1. Expressions are not supported for use on repeating row fields within your data dictionary
2. Text Area, HTML Area, Print Text and Log Area fields are limited in that they can only display (approximately) the first 4,000 characters of any entry. In many circumstances this is sufficient, but if you have really long entries, they will be truncated with report expressions. This is because these field types are actually concatenated in 4,000 chunks in the database and special logic is used to handle word boundaries so that words are always searchable as keywords in the database.

Variable References to Enumerated Types

Enumerated types (fields with a display type of List, Popup, Checkbox, and Tab) may be referenced only as string variables, in which case their title is used in the expression. Only single-valued list type fields may be referenced.

Evaluation of Expressions

As much as possible, report expressions are evaluated by ExtraView by processing at two levels:

- Select list construction within the detail report query; and
- Retrieval of the detailed expression values for rendering in the report row

Syntax checking includes tests for single-quoted strings, double-quoted strings, parentheses and the "cast" function. Syntax checking is done before the attribute is entered and stored as part of the report.

Null Values

The existence of a null value in an expression renders the result of the entire expression value to be null.

Expression Error Handling

Each report expression is syntax-checked and variable references are validated before allowing the containing report to be updated or run. There may be opportunity for the user to create erroneous expressions, as not all expressions can be determined without reference to the values in the database.

Report Field Titles

To support report-specific titles on the expression fields, the **ALTERNATE FIELD TITLE** attribute is supported. This compensates for the difficulty for users to create column headings when they are sharing a data dictionary entry, such as the EXPRESSION type entries. The **ALTERNATE FIELD TITLE** attribute appears within the GUI of the report builder to allow the user to define a title for a report through a simple selection mechanism.

Use of Session Variables

Session variables defined in the data dictionary may be used as substitutable variables in a report expression. Each referenced session variable value must be set using a runtime filter for the report. Session variables may also be used within runtime filters to provide a mechanism to provide the same value for the occasions when the same field is used multiple times within filter queries. This can happen with advanced queries that use multiple **and** and **or** conjunctions. Date fields within the ExtraView database are stored as *timestamp* type fields, which is done for internal efficiency and to facilitate a wide number of internal computations. This does not make them very simple to work with in expressions, but the following examples should help significantly. The session variable must have been defined with a compatible display type, such as DATE, DAY, or TEXTFIELD for use within the expression.

Date Calculations

A frequently use of expressions is to calculate the difference between two date fields. The internal database representation of date fields use the TIMESTAMP data type. This data type does not lend itself to simple addition and subtraction to calculate differences between dated. The date_diff function is an ExtraView function that simplifies date calculations. It is used in this way:

NUM_DAYS = date_diff($$DATE_FIELD_1$$, $$DATE_FIELD_2$$)

This returns the difference as a whole number, between the two date fields.

Examples

1. Compute the elapsed time, in number of days, from when an issue was created, to when it was last updated, and place the result in a field named DAYS_ELAPSED:

   DAYS_ELAPSED = date_diff($$TIMESTAMP$$, $$DATE_CREATED$$)

2. Compute a total sale value with sales tax based on a rate of 8%:

   $$currency_field$$ * 1.08

3. Count the number of selections of a value in a field named **my_field** with a value of **Yes**, when another field named **other_field** has a specific value of **val**:

   case when ($$my_field$$ = 'Yes' and $$other_field$$ = 'val') then 1 else null end

4. Add seven days to the current date:

   Oracle and SQL Server - $$SYSDATE$$ + 7

   MySQL - DATE_ADD($$SYSDATE$$, INTERVAL 7 DAY)

   Derby - {fn TIMESTAMPADD(SQL_TSI_DAY,7,$$SYSDATE$$)}

5. This is a more complex example, which concatenates several fields together, and places the results in a single output column on a report. The example takes the fields SALUTATION_TITLE, FIRST_NAME, LAST_NAME as the first line of the output. The chr(10) provides a linefeed so that the field ADDRESS appears on the second line. The next line contains ADDRESS_2, then followed in the next line by ADDRESS_3. The next line contains a concatenation of the CITY, STATE and POSTAL_CODE. The final line contains the PHONE field. The decode statements are standard SQL that allow the suppression of the field, if there is no value returned from the query.

```
decode($$SALUTATION.TITLE$$,'','',$$SALUTATION.TITLE$$ || ' ') ||
decode($$FIRST_NAME$$,'','',$$FIRST_NAME$$) || chr(10) ||
decode($$LAST_NAME$$,'','',$$LAST_NAME$$) || chr(10) ||
decode($$ADDRESS$$,'','',$$ADDRESS$$ || chr(10)) ||
decode($$ADDRESS_2$$,'','',$$ADDRESS_2$$ || chr(10)) ||
```

```
decode($$ADDRESS_3$$,'','',$$ADDRESS_3$$ || chr(10)) ||
decode($$CITY$$,'','',$$CITY$$ || ', ') ||
decode($$STATE$$,'','',$$STATE$$ ||' ') ||
decode($$POSTAL_CODE$$,'','',$$POSTAL_CODE$$) || chr(10) ||
decode($$PHONE$$,'','',$$PHONE$$)
```

The end result is that you have composed a single field that contains the address in the expected format, although the components were each stored as individual fields.

Remember Filter Values

In between sessions, ExtraView automatically remembers the filter values on advanced search screens. This is a time-saving device, as a user often wants to return to where they left off when returning to ExtraView. Further, it is only a single-click operation to clear the filters. However, there is one item to recognize with this, in that the administrator may alter the metadata supporting the filters, or may even remove access to a field for a user. To support this, all remembered filter values will be removed if the administrator alters metadata such as a search layout.

Complex Runtime Filters

There are occasions when you might want a user to run a report that has a complex set of advanced filters, including the use of several **and** and **or** conjunctions. For example, you might want to pose the following query filters (this is not a real query, but it is useful as an example):

  (Customer = * **Ask at runtime** * AND Status = **Open**)
OR (Customer = * **Ask at runtime** * AND Status = **Fixed**)
OR (Customer = * **Ask at runtime** * AND Status = **Duplicate**)

This would be represented in the report editor like this:
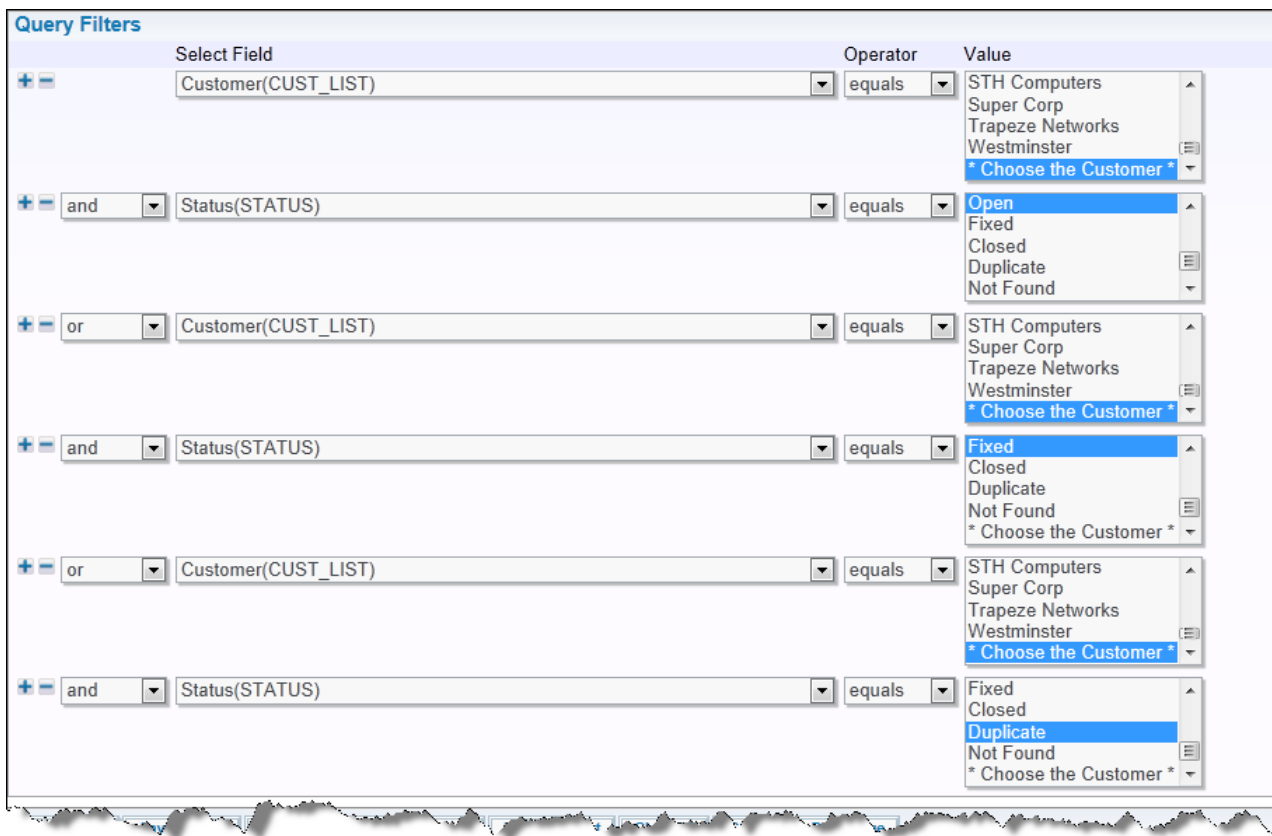
Setting up a complex query with runtime filters

When the report is run, the user will see the following screen:

| Run Report | Cancel | Print Page | | **Report that uses session variables** |

This report uses the value(s) you select for the following field(s), and will substitute the value(s) you select into all the runtime filter values below that use the filter field. This saves entering the same filter value several times.

Filter Field	Value to be used in substitution
Choose the Customer	* Any * * None * Bank of America Bear Inc. Chrysler

Provide values for the following filter(s) for the report

	Field	Operator	Value
	Customer	=	* Any * * None * Bank of America Bear Inc. Chrysler
AND	Status	=	Open
OR	Customer	=	* Any * * None * Bank of America Bear Inc. Chrysler
AND	Status	=	Fixed
OR	Customer	=	* Any * * None * Bank of America Bear Inc. Chrysler
AND	Status	=	Duplicate

| Run Report | Cancel | Print Page |

Running a complex query with runtime filters

When the user selects a customer from the top list, all the bottom filters that require a Customer will be set to use the value set from the top list. To accomplish this type of query with complex runtime filters, you set up Session Variable type fields in the Data Dictionary. Session Variable fields are created as follows:

- **Administration** menu
- **Site Configuration** tab
- **Data Dictionary** menu
- **Session Variables** tab
- **Add** button
- Provide a **Fixed Name** for the field
- Provide a **Title** for the field. It is helpful to use a title like *Choose the XXXXX* so that the runtime filter will show a useful title. Obviously, *XXXXX* represents the title of the filter field
- Choose the **Display Type** for the field. It is important to understand that you need a session variable field for every filter that you will use in your complex queries, and that the display type must match the display type of the filter field it is being used with
- **Add** the field to the data dictionary.

Once this has been done, you will see that the advanced filter lists look something like this to the end users:
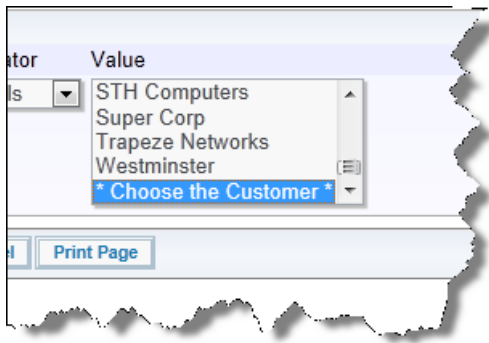
Session Variable field to be used as a runtime filter

Note that the Session Variables appear at the bottom of list fields. If you are using the Session Variable as a filter within a text, date or number field, you should enter it as $$FIELD_NAME$$, substituting FIELD_NAME with the name of your Session Variable field.

Charts

There is overall control to access charting from within the Search / Report screens, as well as control over access to creating individual chart types. These are all controlled by security permission keys, as explained in the following table.

Security Key	Purpose
SR_PERSONAL_CHART	This security permission key allows access to produce charts and to save them for personal use. Read permission is needed.
SR_PUBLIC_CHART	This security permission key allows access to produce charts and to save them for public use or to save them for use by an individual user role. Read permission is needed.

Aging Reports

When looking at the trends and movement of issues through a process as part of aging reports, it would be misleading to see only a partial view of an issue's movement through the workflow.

Therefore aging reports do not use the STATUS.xxxx permission keys and the user will see all the potential status values on the report. If this behavior is not desirable you should remove the user's access to all aging reports, using the SR_PUBLIC_AGING, SR_PERSONAL_AGING and SR_USERGROUP_AGING permission keys.

Custom URL Linked Reports

ExtraView reports has a feature that makes it possible to gain access to reports or pages generated outside the system, or reports generated internally (via the API) that involve custom code. The administrator can create these links, and then make the reports available to a general user community by saving the report as a Public Report.

To use this feature, or to turn this off for your users, use the security permission keys named SR_PERSONAL_LINKED_REPORT and SR_PUBLIC_LINKED_REPORT, according to whether users should be able to create their own personal linked reports, or whether they should be able to create public linked reports.

As custom URL linked reports can generate any type of output, and may not even output directly to a browser, you may not put a custom URL linked report on the Home Page. If you have a need for this, it is possible to use the user custom programming feature to place a similar report on the Home Page.

Link to a Report

From the **Report** screen, select the **Create New Custom URL Report** option from the list, and press the Go button.

The following screen appears:

Link to Report screen

- To link to custom reports created via the API, enter a report title, a report description, and then enter the URL in the field provided
- The URL can be an absolute path to a URL external to ExtraView. In this case, the URL must begin with **http://**. If you are pointing to a relative path within ExtraView, only enter the option and action you want to pass. For example, to set up a report on an absolute URL that exists at a different server, the URL address for the report may look like this – http://www.myserver.com?param1=xxx¶m2=yyy To set up a relative URL to produce a report within ExtraView itself, may look like this - ?p_option=search.SearchReportDetailDisplay&p_action=doRunDetailed&id=10200 This report actually runs a detailed report looking for the ID of 10200
- Note the Limitations at the bottom of this page, to decide whether you want to open the report within a new browser tab / window
- Select any desired filters, noting that these will only be available to reports generated internally within ExtraView, unless a user custom code exit is used to process these and pass them to an external URL
- Click the **Run Report** button, or **Save** the report

Note: ExtraView will not validate the URL that you enter at the time you create a Custom URL report. It is the user's responsibility to check that valid URL's are entered. It is exceedingly likely that you will generate a runtime programming exception error if you enter an invalid URL.

Notice that you can turn the filters on the screen off and on, with the ▬ and ➕ buttons.

Limitations

Different browsers and different web sites place different restrictions on how content is accessed. This constraint might be exacerbated by ExtraView, according to whether you are trying to open the Custom URL report within an browser Iframe. This happens when the report is on ExtraView's Home Page, or when the report is opened within a Workspace. Some web sites will deny access to reports opened in this way, either because of their concern for security, or simply because it is their policy to not allow this mode of operation.

Note the prompt on the screen, **Open in new tab?** You can choose this option for your Custom URL report, so that ExtraView opens the report within a new browser tab or window, thereby satisfying the conditions set by the external URL. If all the URLs used by your users are within the same domain as ExtraView, this constraint is not liklely to affect them.

Service Level Agreement Reporting

Service Level Agreement management allows the administrator to set up the conditions to measure whether the handling of individual issues fall within predetermined conditions that you have set up to manage your customers. For example, you may have an obligation to respond to all customer issues within two hours, and an obligation to provide closure on an issue within two days. This utility allows you to set up these criteria. Under reporting, your users may use these criteria to prepare and run reports that show your organization's performance against these SLA's.

SLA definitions for organizations can differ widely, and different parts of an organization may need to create their own SLA terms for their individual customers.

ExtraView provides a scalable method to provide many SLA's within the product, but for simplicity an example follows that builds a global set of SLA conditions that look like this:

| | | Status Transition | |
Priority	New-Open	Open-Resolved	Resolved-Close
P1 Issues	< 2 hours	< 1 day	< 2 days
P2 Issues	< 4 hours	< 5 days	< 30 days
P3 Issues	< 1 day	No commitment	< 180 days

This implies that the STATUS field contains (at least) the values *New, Open, Resolved* and *Closed*. The PRIORITY field has the values P 1, P 2, and P 3.

Note: A key feature of measuring SLA's is that an issue may move between the different statuses many times, yet it is the total time in the status that is measured and reported. For example, a customer may report a P 2 issue via email, and it is initially created as an issue and placed in the *New* status via the EVMail capability. Your support technician moves the issue to an *Open* status in 3 hours, within the agreed time of 4 hours or less.

Three days later, the issue is moved from the *Open* to the *Resolved* status and handed back to the customer. This is within the agreed time for the SLA. However, some time later the customer reports back to your support technician that the resolution was not sufficient, and the issue is moved back to the *Open* status. In a further four days time the issue is handed back to the customer and the issue is once again moved to the *Resolve* status. The customer accepts this resolution.

The SLA that followed this workflow was not met, as the issue took a total of seven days to resolve, not just the three or four days for which the issue was *Open* on each occasion. The SLA measurement spans all the occasions the issue was in the *Open* status.

Following are instructions on how to set up this scenario to monitor all the conditions in the above table. The reporting tool allows end users to run reports that utilize SLA States defined by the administrator. The end user may not define their own SLA States.

From the **Site Configuration** menu in **Administration**, select **Service Level Agreement Management**. You will see a screen similar to:

SLA Management screen

Click the **Add** button. First, we use the upper part of the screen to create and add the SLA. For our example we can use the values in this screenshot to create our first SLA named **NEW_ISSUE_SLA**.

The NEW_ISSUE_SLA

Looking at the table at the top of this page, we are going to create three SLA's, one for **New** Issues, one for **Open** issues and one for **Resolved** issues. The **New** SLA State will have three SLA States, one for **P 1** issues, one for **P 2** issues and one for **P 3** issues.

We create the SLA in the top-half of the screen, providing a Fixed Name, Title, Business Area (if any) and Project (if any) as the minimum. There is the capability to enable/disable the SLA, provide a sort sequence and notes.

To define the three SLA States, we use the bottom-half of the screen. For each SLA State, we provide a Fixed Name, and Title. The **Drop-dead threshold** is the time in the status you will provide in the filter below, for this SLA State. At the same time that we implement the requirements for the SLA's we will optionally add an additional value. This is the **Warning threshold**. After this time, and before the drop-dead date / time for the SLA, reports will display the fact that the deadline is approaching.

Use the section beneath the Define SLA States to provide the filters for the SLA State you are defining. Given we are defining the SLA for **New** issues, and the first SLA State we are defining is for **P 1** issues, the filters should be *Priority = P 1* and *Status = New*, as shown below:

Defining an SLA State

Each time we define an SLA State as exemplified by the above paragraphs, we use the **Add to state list** button to save the SLA state. When we have added all the individual SLA States, you must also use the **Add** or **Update** button at the top and bottom of the screen to save the entire set of metadata, i.e. you are saving the overall SLA information plus the individual conditions that make up the SLA's. We can create the SLA with the following values:

State fixed name	State title	Warning threshold (hours)	Drop-dead threshold (hours)	Filters
P1	P1 Issues	1	2	Priority = P1 Status = New
P2	P2 Issues	3	4	Priority = P2 Status = New
P3	P3 Issues	20	24	Priority = P3 Status = New

Next, to implement the requirement, we will create an **OPEN_ISSUE_SLA** with the following metadata:

Fixed name = **OPEN_ISSUE_SLA**
Title = **Open Issue SLA**

State fixed name	State title	Warning threshold (hours)	Drop-dead threshold (hours)	Filters
P1	P1 Issues	18	24	Priority = P1 Status = Open
P2	P2 Issues	96	120	Priority = P2 Status = Open

Lastly, we implement the RESOLVE_ISSUE_SLA as follows:

Fixed name = **RESOLVED_ISSUE_SLA**
Title = **Resolved Issue SLA**

State fixed name	State title	Warning threshold (hours)	Drop-dead threshold (hours)	Filters
P1	P1 Issues	40	48	Priority = P1 Status = Resolved
P2	P2 Issues	650	720	Priority = P2 Status = Resolved
P3	P3 Issues	4000	4320	Priority = P3 Status = Resolved

When you have completed these additions, you have set up three SLA's to measure the conditions in the table at the beginning of this section, plus we have added warning times to aid users be aware of impending deadlines being exceeded.

Searching for Disabled Values

There are occasions when users may need to search for results, using disabled list values as filters. For example, the user may want to search for issues that are assigned to a user whose account is disabled.

This functionality is accessed from the advanced search mode. When in advanced search mode, and a list contains inactive or disabled entries, there will be an entry in the list of values, labeled * **Show disabled entries** *. When you select this entry, the screen is refreshed and you will see all the disabled as well as enabled values in the list. Note that you can remove the disabled values from the list by then pressing on the * **Do not show disabled entries** * list entry.

Record Selector

This feature allows a button to be placed on the menubar of a Quicklist or Column Report, and this button is used to select some arbitrary issues on the report, and to perform an operation on the selection. This screen shows the result of clicking the button **Turn On Record Selector** on the Quicklist report.

The record selector mechanism

The user may click on any of the checkboxes to select the records, or they may click on the **Click here to check or uncheck all the records on this page** checkbox. After checking the boxes, if the user selects the **Detailed Report** button or the **Group Issues** button then a Detailed Report appears with these issues, or the selected issues are used as input to an operation to group them together. If

the user is within a Workspace, then the selected issues may be dragged to another report, where they may be dropped, to provide an action such as changing one or more values within the dragged issues to the filter values of the panel where they are being dropped.

The record selector option requires the following setup –

Data dictionary	The Label field named EV_REPORT_SELECTION must exist in the data dictionary. This is preset for you
Permissions	Set the Read permission to **Yes** for each role that you want to use this feature, for the permission key named PR_RESOLUTION.EV_REPORT_SELECTION.
Layouts	On each of the Quicklist layouts that you want to use the feature, place the field EV_REPORT_SELECTION in row 1, column 1.
Column Reports	Within the Column Report editor, the user chooses to display the available buttons, then selects the button with the check mark (✓) as the first column on the report output

Mass Updates of Issues

It is not recommended that you give end users access to this facility, as it allows the mass update of issues. One wrong update could affect hundreds or thousands of issues and may be very difficult or impossible to undo. This function is most frequently used for tasks such as:

- Reassigning issues from one person to another
- Reassigning open issues from one release to the next
- Setting a field on many issues to a single specific value

Access to this feature is controlled by the security key named PR_RESOLUTION.MASS_UPDATE_ISSUES.

Assuming the permission is set to be able to use this feature, there will be a button labeled **Update** on Quicklist Reports, Detailed Reports and Custom Reports that you generate. This button allows the batch or mass update of a list of fields available to you.

Not all fields can be updated by this method. Only fields to which you have write access, and fields that are of display type List, Pop-up, Tab, Checkbox, Log Area, Text, Currency, Number, Decimal, Date, Day and User can be updated. The fields that are offered for update are those that exist on the *edit* layout in the current business area and project. You cannot update any field outside your current business area and project. However, if your query that selected the field to mass update contains records in other business areas and/or projects, then these will be updated, assuming the field is on the *edit* layout of the specific area / project.

The execution of mass updates is a background process within ExtraView. Once started, the process cannot be interrupted. This provides several advantages, in that ExtraView will prevent several mass update operations happening concurrently (this may lead to data integrity problems, and the corruption of data) and it gives a significant amount of insulation from problems such as a disruption of the communications network or the client computer crashing midway through the process. To use this feature:

- First, prepare a report
- Click on the **Update Issues** button on the menu bar of the report
- You may update from one to five fields with one pass of the mass update procedure. After you select a field, you can use the ➕ button to add an additional field to the update process. If you want to remove a field, you can use the ➖ button that appears on the second and subsequent fields
- For each field you add you will see a prompt asking for the new value for the field as well as a list of all the issues that are about to be updated. You can uncheck any issue from the list that you want to exempt from the update
- You must check the button **Generate Email** button if you want the standard notification sent for each issue being updated. The default for this is not to send email, as you may be generating a very large list of updates and hence email notifications.
- Click **Update all records**. Note that if you are updating a large number of issues, the process may take some time to complete, probably around one to three seconds for each issue.

Note the following when using Mass Update:

- If a field you are updating is a multi-valued field or part of a repeating record that can have many values, then the update will add the new value to the existing list of values for the field you are updating
- If you attempt to update a field to a value of * **None** *, ExtraView will perform a check to ensure this is a valid entry, dependent upon whether the field is required, or whether the field is subject to a FIELD/LAYOUT VISIBLE IF layout cell attribute. If the field is required, then the * **None** * entry will not appear in the list of values shown. If the field is subject to a FIELD/LAYOUT VISIBLE IF constraint, then an error will be returned if ExtraView attempts to update the field in a way that violates the FIELD/LAYOUT VISIBLE IF rule
- You may not be able to undo this mass update operation, so take care before pressing the **Update all records** button

- Mass update is not restricted by rules such as Visible If and Required If, etc. Issues will be updated and only enforced if you attempt to interactively update an individual record again. The principal reason for this is that mass update cannot stop during its execution to ask for additional input on an issue-by-issue basis. It is a background task. When you modify an individual record following the mass update, the rule will be enforced
- To conserve database resources, ExtraView limits this feature to updating a maximum of 5,000 issues in a single pass. If you need to update more than this number, simply run the selection multiple times and perform the update over again
- You may update issues across multiple business areas and projects provided the value to be updated does not violate the required and required if layout cell attributes of the current user role, current business area and current project of the user performing the operation. Also, you must not violate the allowed values relationship of the issue, within the current user role, current business area and current project of the user. If any of these conditions are encountered, you will see an error message stating why the issue cannot be updated
- If you include a field with a display type of Log Area in a mass update operation, the existing log area entries are untouched, and a new log area entry is added, corresponding to the text entered into the mass update field
- It is possible that you may update issues across multiple business areas and/or projects, and that in doing so you will attempt to mass update a field that does not exist on the *edit* layout for one or more areas / projects. When this happens, the value in that area / project will not be updated, and no error occurs
- The mass update function will not update all records if the query used to generate the list for updating used the advanced query interface, and used the or conjunction. There is a simple workaround, to create the list using each part of the filter without the or conjunction in turn, and perform the update
- Any user or administrator may only run one mass update operation at one time.

Example

This shows how you can reassign all open issues from one person to another, for a given product: Prepare a Quicklist report that uses Product, Assigned To and Status as filters. From the resulting report output, press the **Update** button. You will see a screen similar to:

Selecting a field to update

Select the Assigned To field from the list and the screen will redisplay, showing something similar to:

Mass update screen

From the select list labeled New Value for Assigned To, select the person you want to reassign the issues to. You can view any of the issues to check whether they should be part of the update, and you can uncheck any of the issues. You can also add one or more fields to the mass update by pressing the ➕ button.

If the field you are trying to update is dependent upon another field (for example a *module* field may be dependent upon *product*), then you will be prompted for the parent field, then the child field, to ensure that the relationships between the values are kept intact and only valid combinations are stored.

If you try to update a parent field that would invalidate the child records, for example trying to set the product field to a value that is not appropriate for a given module, you will receive an error for that issue, or issues, and these issues will not be updated.

When ready, you can press the **Update all records** button.

You will be asked to confirm that you want to update the issues.

The default is that email notification is not sent when performing a mass update of issues. However, you can turn this on, by clicking the **Generate Email** checkbox to on.
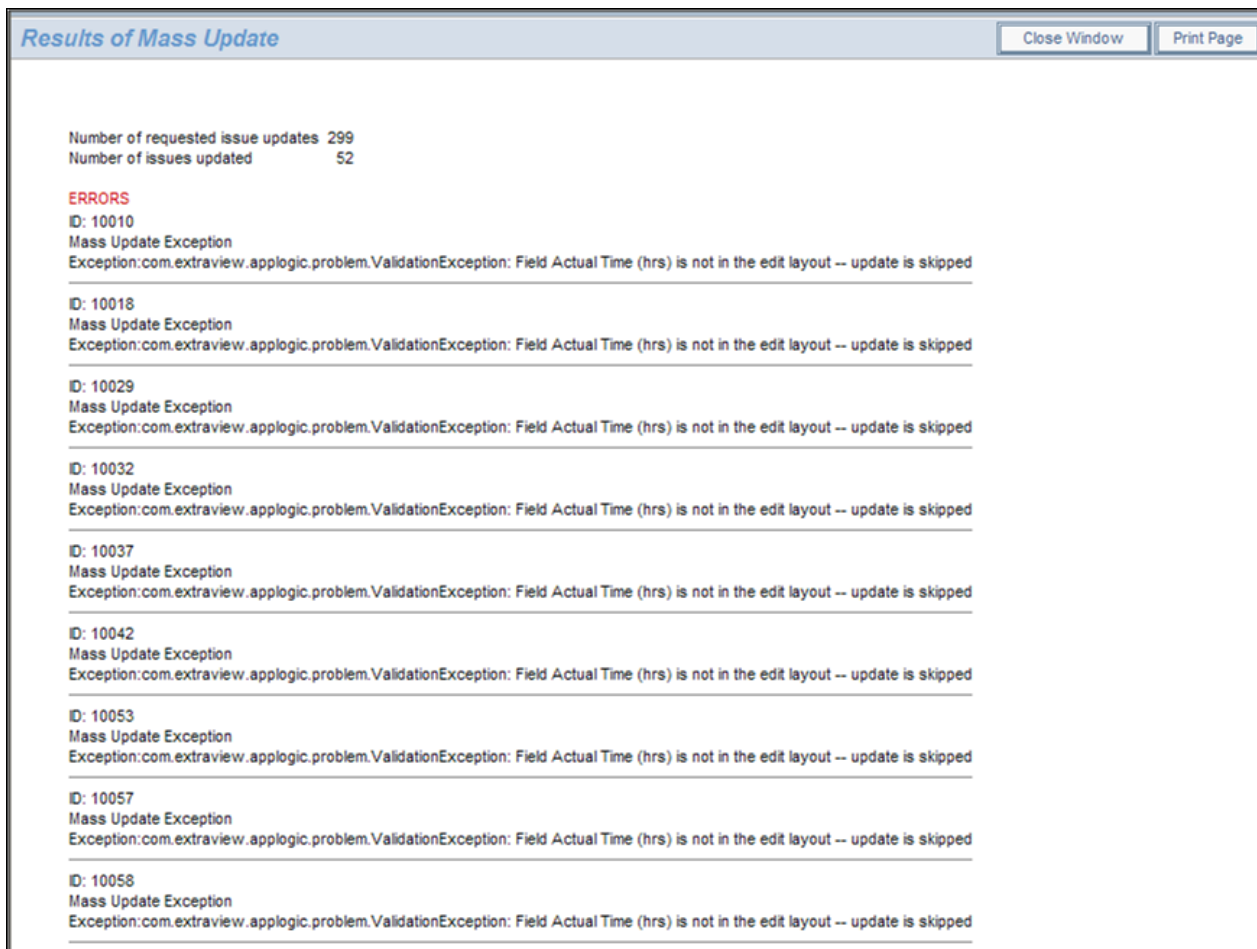
On the screen, there is a checkbox that allows you to alter the mass update capability to a mass clone utility. When you check this, you will see that you can select a new business area and a new project. When you perform the update, the issues are copied to the new project, rather than updated. When you undertake this function, you are asked to confirm the operation before it proceeds. See the next section of this guide for more details.

While the mass update is underway, you will see a progress window as follows:
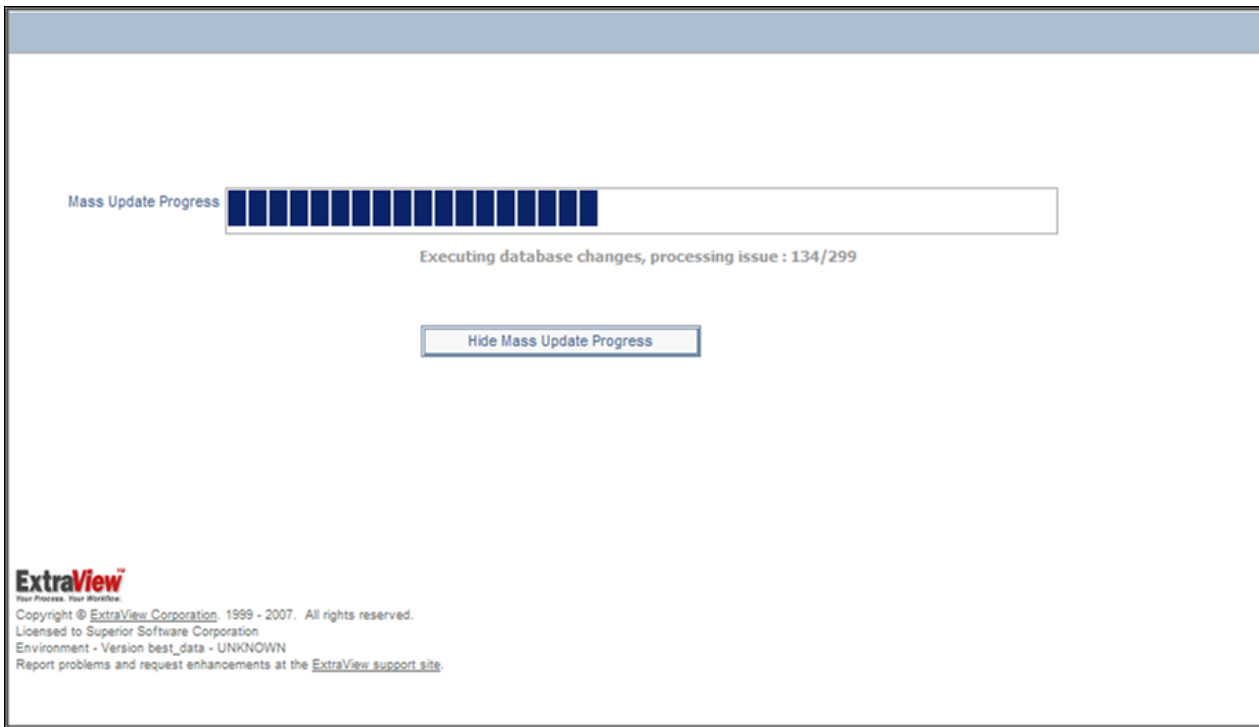
Mass update progress bar

Note: Pressing the button to hide the progress bar, or closing the window has no effect on the execution of the mass update. As a background process, it will continue. The progress bar is purely for informative purposes. Once the mass update is complete, and assuming you have not closed the progress window, you will see a summary of the mass update, including any errors encountered. The most common error is trying to perform a global mass update across several business areas, when the field you are trying to update is not on the *edit* screen for the business area and project to which the issue belongs.

Results of a mass update, showing errors

Cloning Issues via Mass Update

This procedure only clones issues from one Project in your installation to another Project. The destination project may be within a different Business Area. The operation of this feature is from within the Mass Update facility described in the last section. The user role must have both read and write access to the PROJECT fields (PR_ADD_PROBLEM.PROJECT and PR_RESOLUTION.PROJECT) in order to be able to perform this function.

Follow the guide in the previous section and select the Project field from the list of available fields to update. Your administrator must have given you permission to write to this field within your current Business Area and Project. When you choose the Project field, you will see a screen similar to the following:

Mass cloning of issues from one Project to another

- You must give your consent to the cloning of the issues displayed before proceeding.
- Select the destination Business Area and Project in which to place the newly created issues
- If you choose Project, you cannot select any additional fields to update beyond the Area field
- Proceed by clicking on the **Clone all records** button
- No email notification is generated from the mass clone operation, even if the Generate Email checkbox is checked.

Mass Deleting of Issues

Extreme caution should be used with this utility. Only user roles who have the authority to make mass changes which cannot be undone should be granted access to this utility.

The primary use for this feature is to allow organizations who have record retention policies to be in compliance with these policies. For example some organizations may have a policy that states that records more than 7 years old should be deleted.

Access to this feature is granted with the security permission key named PR_RESOLUTIO.MASS_DELETE_ISSUES. Again, it is strongly recommended that only user roles who have permission to make wholesale changes to your database should be given permission to this utility.

With permission, a button labeled **Mass Delete Issues** appears on the output of Quicklists and Column reports. This button invokes the utility and shows a screen similar to the following:

The same set of issues that were seen on the Quicklist or Column report are displayed, each with a checkbox allowing it to be selected or deselected for deletion. Note the checkbox at the bottom of the list at allows you to select or deselect all the issues with a single click of your mouse.

If you do not check the **Delete All History** box, the history of the issues will be retained within the database, and it may be possible, with the use of database scripts to recover the deleted issues. If you check this box, the issues cannot be recovered and are permanently deleted.

When you have made the appropriate selections for deletion, click the **Delete Selected Records** button on the menubar. You will be asked to confirm that it is OK to proceed. There will be a progress screen displayed. Once the deletion is complete, there will be a summary of the deletions made.

Configuring the Fields Displayed on the Screen

By default, the fields named VIEW_BUTTON, ID, AREA, PROJECT, and SHORT_DESCR appear on the review screen. These fields may be changed via the behavior setting named MASS_DELETE_TEMPLATE. This field contains a comma-delimited list of the fields that are displayed.

Scheduled Reports

With write permission to the key named SR_REPORT_SCHEDULE, members of user roles have the ability to generate reports on a timed, periodic basis. These reports are sent out via email and may also be added to the document repository for future reference. The task named SCHEDULED_REPORT_TASK must also be running.

Please view the end user documentation here for instructions on how to use the report scheduler, to send reports by email and to store

Document Repository

Reports that generate output that you send to the document repository are controlled with the following:

- The Report Scheduler Task must be running
- Users accessing the document repository must have permission to schedule reports via the role-based security permission key named SR_REPORT_SCHEDULE

- Users adding documents or deleting reports with the document repository must have write permission to the security permission key named SR_REPORT_REPOSITORY_ACCESS
- Users who view or download documents from the repository must have read permission to the security permission key named SR_REPORT_REPOSITORY_ACCESS
- Administrators who require access to the utilitiy named **Document Repository Manager**, must have read and write permission to the security permission key named CF_REPORT_DOCUMENT

The Document Repository Manager

The utility resides within the **Operational Tasks** administration menu.

Document Repository Administrative Utility

The utility allows you to perform the following tasks:

- Filter the documents by:
  - Created on or before a date
  - Last updated on or before a date
  - Created on or after a date
  - Last updated on or after date
- Select any subset of documents, whether filtered or not and delete these documents, irrespective of the owner of the document.

Controlling Large Queries

With a very large database, you may want to control how large reports are run, restricting users from doing repeated queries which may each try to return hundreds of thousands or more records. Use these behavior settings to provide some sensible constraints:

- ALLOW_UNLIMITED_SEARCH – This behavior setting can be used to turn off the ability to perform unrestricted queries on the database. When this setting is set to NO, it is used in conjunction with the next setting, LIMIT_QUERY_ROWS
- LIMIT_QUERY_ROWS – This sets a numeric limit on the number of rows returned by a single user's query. This has no effect if ALLOWED_UNLIMITED_SEARCH is set to YES
- MINIMUM_SEARCH_FIELDS – This setting is used to force a user to select a number of filters before their queries will be executed. If you specify a single number, then this specifies the number of filters in addition to the KEYWORD filter that must be provided. If you provide two numbers, separated by a comma, then the first number specifies the number of filters in addition to the KEYWORD filter that must be provided and the second number specifies the number of filters that must be provided when no KEYWORD filter is provided. The default is 0,0. For large databases, a setting of 2 or 3 for each number typically provides sufficient control, to ensure that only a small section of the database is searched at one time, and that users do not attempt to download millions of records or perform complex queries across millions of records.

  If the user has selected the advanced query mode, ExtraView only counts the conjunctions with an *and* conjunction. The other conjunctions do not restrict the query keyword query.

Localization of Reports

When you have localization turned on, and you are an administrator (i.e. your current role is the same as that defined in the behavior setting named ADMIN_OVERRIDE_ROLE), then there will be an additional prompt at the top of the Custom Report, the Summary Report and the Chart screens, when you are editing the report and as shown below:

Localizing a Report Description

The **Localize** button can be used to provide a localized description of the report for each locale you have defined. However, note that you cannot localize the Report Title. The reason for this is that each Report Title must be unique through the entire system.

Reports at a Point in Time

There are two special purpose filters, with data dictionary names of HIST_RANGE_START and HIST_RANGE_END, with default titles of **Historic data filter (start)** and **Historic data filter**. These allow the user to run a report as of a specific date / time, range of dates, in history. Simply use these fields as filter on a query and the results displayed will be shown as they existed at that point in time.

These filters have a special use, and may only be placed on a search layout for use in standard queries. There is no access to use this filter from advanced query screens.

It is important to note that the filters will only return reasonable results if the user sets HIST_RANGE_START to be less than or equal HIST_RANGE_END.

Other filters used in conjunction with a query that uses HIST_RANGE_START and HIST_RANGE_END are applied to the current values of these fields, not to the values that these issues had at any point in the past. It would not be logical to do otherwise, as the filter values may have changed many times during the history time range selected. One consequence of this is that it is never possible to return values for deleted issues, within the time period selected.

Updating Consecutive Issues from Reorts

Some workflows find it useful to be able to edit a number of records in sequence. For example, a user may want to run a report of their open issues, then edit each record in turn, but without needing to return to the report to click on the **Edit** button for each issue. An alternative is to use the Quickedit feature within the report itself.

To set up this feature, there is a permission key named PR_RESOLUTION.ALLOW_EDIT_NEXT_PREVIOUS. This key is operable when write permission is available for the user's current role, and the user executes a Quicklist or column report. When the user drills down into any issue with the **Edit** button on any issue, the *edit* screen will have two additional buttons on the menu bar, titled **Update & Prev** and **Update & Next**. When the user clicks one of these buttons, they will be taken directly to the previous or next issue following a successful update.

Ranking Fields on Reports

This is documented in the section on special purpose fields. Click here to access the page.

Privacy Groups

Privacy groups are designations that limit the visibility of issues, except those users who are members of the privacy group to which the issue belongs.

For example, your company may have six divisions, each of whom report issues using ExtraView. You may want to set up a structure, whereby each of the divisions report issues that remain private to each division, and are not shared with other divisions. Layered on top of this is the ability for a management or supervisory function that has access to all the issues, irrespective of the division that originated the issue. This can be implemented within ExtraView by setting up individual privacy groups for each division, restricting each member of a division to only enter issues for their own division (i.e. privacy group), and giving the management function access to all privacy groups.

To turn on the privacy group feature in your instance of ExtraView, use the behavior setting named ENABLE_PRIVACY_GROUPS. This is found in the Users behavior settings menu.

Note: If ENABLE_PRIVACY_GROUPS is set to NO, then issues may only be marked as PUBLIC or PRIVATE. If ENABLE_PRIVACY_GROUPS is set to YES, then issues may be marked as PUBLIC, PRIVATE or a member of a privacy group that you create. With the latter option, you may also elect to remove either the PUBLIC and / or PRIVATE designation from the list, as explained in the section following.

Note: There is a behavior setting named ENABLE_PRIVACY_GRP_OVERRIDE. If the value is set to YES, then internal users can see all issues regardless of the value of the PRIVACY field. Internal users are defined by the user's personal Company Name being identical to the company name defined by the behavior setting named COMPANY_NAME. If the value is set to NO, users may only see issues when they are a member of the privacy group to which the issue is assigned.

The precise rules for how issues are handled with respect to privacy groups are as follows:

- **Private issues**. When an issue is marked as private, then only users whose Company name setting in their user account settings is identical to the behavior setting named COMPANY_NAME will be able to view and update the issue. If the Privacy field does not appear on the Add Issue screen, then ExtraView will automatically set the PRIVACY field with its default value as specified in the data dictionary (usually PRIVATE). If there is no default, ExtraView always sets the value to PRIVATE. Security permission settings for individual fields, screens and functions still apply. Note that there is a behavior setting named ENABLE_COMPANY_NAME_ACCESS that overrides this behavior for specific circumstances. Please refer to the section in this guide on Company Name Security for full details.
- **Public issues**. When an issue is marked as public, then any user with an active account can potentially view and / or update the issue. The specific permissions to view and update the entire record or individual fields are subject to the security permissions for each object.
- **Issues within a privacy group**. As a user, you must be a member of the specific privacy group to view and update the record. Once again, security permissions for all objects still apply.

PUBLIC and PRIVATE Visibility

When using privacy groups, it is sometimes useful to be able to control whether the entries for PUBLIC and PRIVATE also appear within the select list boxes on the *add*, *edit* and *search* screens.

For example, you may always want a privacy group to be selected for every issue without exception.

This can be achieved through the settings of the following security permission keys. These can be set for each user role within ExtraView, in the standard way of setting security permissions.

Security permission key	Purpose
PR_ADD_PROBLEM. SHOW_PRIVATE_IN_PRIVACY_LIST	Write permission to this security key is needed to place the entry for Private in the select list for the Privacy field on the add issue screen. If this key is set to N, then issues cannot be marked as private for the user role.
PR_ADD_PROBLEM. SHOW_PUBLIC_IN_PRIVACY_LIST	Write permission to this security key is needed to place the entry for Public in the select list for the Privacy field on the add issue screen. If this key is set to N, then issues cannot be marked as public for the user role.
PR_RESOLUTION. SHOW_PRIVATE_IN_PRIVACY_LIST	Write permission to this security key is needed to place the entry for Private in the select list for the Privacy field on the edit issue and query screens. If this key is set to N, then issues cannot be marked as private or searched for the user role.
PR_RESOLUTION. SHOW_PUBLIC_IN_PRIVACY_LIST	Write permission to this security key is needed to place the entry for Public in the select list for the Privacy field on the edit issue and query screens. If this key is set to N, then issues cannot be marked as public or searched for the user role.

Creating New Privacy Groups

From the Site Configuration administration menu, click the **Privacy Groups** button. A screen similar to the following appears:

Privacy Groups screen

To create a new privacy group, click the **Add** button. The following screen appears:

Add New Groups screen

Enter a database name (no spaces or special characters) and a display name. If an interest list is allowed on the field, you can optionally create an interest list for this value and add a user. When ready, click the Add button to create your new value. Once the Privacy Group is created, the administrative user may assign individual users to the group by checking the appropriate check box on the Change a User's Details screen.

Change User's Details screen with Privacy Groups

User Roles

User Roles are the functional teams of your company or the external users that will be using ExtraView. User Roles are assigned specific privileges based on what you want each of them to be able to see and do. Example user roles may be:

- Administrator
- Customer

- Engineer
- QA
- Manager

There is no limit to the number of user roles that may exist within your ExtraView system. Additional user roles may be created at any time. Individual users may belong to any number of user roles. When a user is given the privilege of belonging to more than one user role, they are automatically given a link in the title bar of the screen that shows their current role. Following the link allows them to alter their role. A user may not change their role when in the add / edit issue process, or during the process of preparing a report as this change may significantly alter their permissions, and would lead to unpredictable behavior.

Altering the Current User Role

Users have the ability to change their role as long as they belong to more than one. This allows each user to wear a "different hat". By changing their role, they can perform different functions. For example, a user may be both an *Administrator* and a *QA Engineer*.

Home Page screen

1. Click on the list of user roles in the navigation bar and observe a list of the roles that your user may adopt
2. Select the role you would like to adopt by clicking on its name. Your role is immediately changed and your permissions will reflect the change. You may find that access to certain buttons on the navigation bar disappear or appear, or that fields no longer show up in the add/edit screens, or that buttons are added and fields appear in some screens; this is a direct result of changing your role.
3. When you change the user role to which you currently belong, ExtraView automatically adapts and presents the various screens and reports that have been customized for that user role.

Adding New User Roles

Click the **User Roles** entry under the Site Configuration tab within administration. Click the **Add** button on the User Roles Screen.

User Roles Summary screen

Add a User Role screen

- Click the **Update** button when done.

Multi-Language Setup

ExtraView allows each user to select their own language for the user interface of the product. In addition to localizing the system messages and prompts, the administrator may also localize their own metadata for the fields and layouts that they create. Note that no matter what languages you want to support on the user interface, ExtraView will handle any alphabet within the data that is entered within issues. Unicode is fully supported.

Setting up a multi-lingual setup requires a number of behavior settings to be set at their correct values. These settings may be found in the **Environment** behavior settings category.

Environment Settings	Purpose
DEFAULT_ATTACHMENT_CHARSET	The default character encoding for files being uploaded to ExtraView. This value is used to select the initial value presented to the administrator when creating a new user. UTF-8 is the default value and should be used for most installations. If the majority of your users use Japanese, you may consider an alternative such as Shift-JIS
DEFAULT_LANGUAGE	This is the default language for the installation and will be used as the default for new users. The value should be set to **en** for English. The two-character country codes are defined within ISO 3166-1. Enter the value in lower case
DEFAULT_REGION	This is the default region for the installation. Unless you are using multiple locales in your installation, and have created other locales, you should not alter this setting. A locale for the DEFAULT_REGION must exist before you alter this setting. Note that this setting is in upper case
DEFAULT_VARIANT	The default variant for the installation. This is not typically used
LOCALIZE_TITLES	This setting is used to turn off and on the localization buttons within administration. This is used when you are using ExtraView's multiple languages on the user interface. When this option is set to YES, a button with the title of **Localize** will appear beside all metadata titles and values that can be localized into different languages. Valid values are YES and NO
MULTI_LOCALE	ExtraView will behave as a single locale system, using the language specified in the behavior setting named DEFAULT_LANGUAGE when this value is NO. When it is set to YES, the administrator may add any number of additional language locales to the system, and provide localized messages and metadata for each locale. If you are only using a single language, do not set this value to YES, as there are performance impacts to running in a multi-locale mode

The following settings are sensitive if you are working within a language such as Japanese, where there are not necessarily any space characters between each word or character. Under these circumstances, the HTML language used in display browsers is not very forgiving and can give strange display results.

Display Settings	Purpose
FOLD_TEXT_POSITION	This specifies the character position at which to fold text lines in TEXT_AREA and LOG_AREA fields. You should not specify a number lower than 65. If you specify a high number, such as 99999, the text input will never be folded. This is important for double-byte languages where spaces may not appear between words, as it will force ExtraView to introduce line-breaks at appropriate intervals
FOLD_WORD_POSITION	This specifies the character position at which to break up a long word in TEXT_AREA and LOG_AREA fields. You should not specify a number lower than 65. If you specify a high number, such as 99999, the long word will never be broken up. This is important for double-byte languages where spaces may not appear between words, as it will force ExtraView to introduce line-breaks at appropriate intervals
REPORT_WITH_FIXED_WIDTH_FONT	This should always be set to NO when working with double-byte languages.

Home Page Shortcut Buttons

This feature allows any number of shortcut buttons to be placed on a layout that is part of the Home Page. The shortcuts either jump to an *add* screen or run a report.

Permissions to the shortcut buttons are controlled by the permission the user has to the underlying object. For example, if you are using a shortcut button to go to an *add* screen, then the user must have permission to go to the business area and project of the *add* screen. If you are using the shortcut button to run a report, then it is the permission the user has to that report that controls the visibility of the button.

Creating the Button Fields

Create a field in the data dictionary for each shortcut button that you require. To create a shortcut to an *add* screen, create a field where the fixed name has the form **BUTTON_ADD_name**. To create a shortcut to a report, the fixed name must have the form **BUTTON_RUN_name**. All fields should have a display type of **Custom**. The title of the button becomes the text that appears on the button. Make sure you provide security permissions for each role that you want to use each button. This is an example of a button in the data dictionary:


Shortcut Button in the Data Dictionary

Creating the Layout Type

Go to **Administration ⬥ Fields & Layouts ⬥ Create and Maintain Layout Types**. Add a new layout type. This layout type must have a fixed name of **HOME_PAGE**. The usage must be **Screen**.

Adding the Layout to the Home Page

Go to **Administration ⬥ Fields & Layouts ⬥ Design Center**. You may create a Home Page layout in the *** Global Area *** where it will be used across all business areas and projects, or using the standard inheritance mechanism, you can create different Home Page shortcut layouts in different business areas, projects or roles. Use the **Add a new layout for the entire system** select list to add the layout type you just created in the previous step.

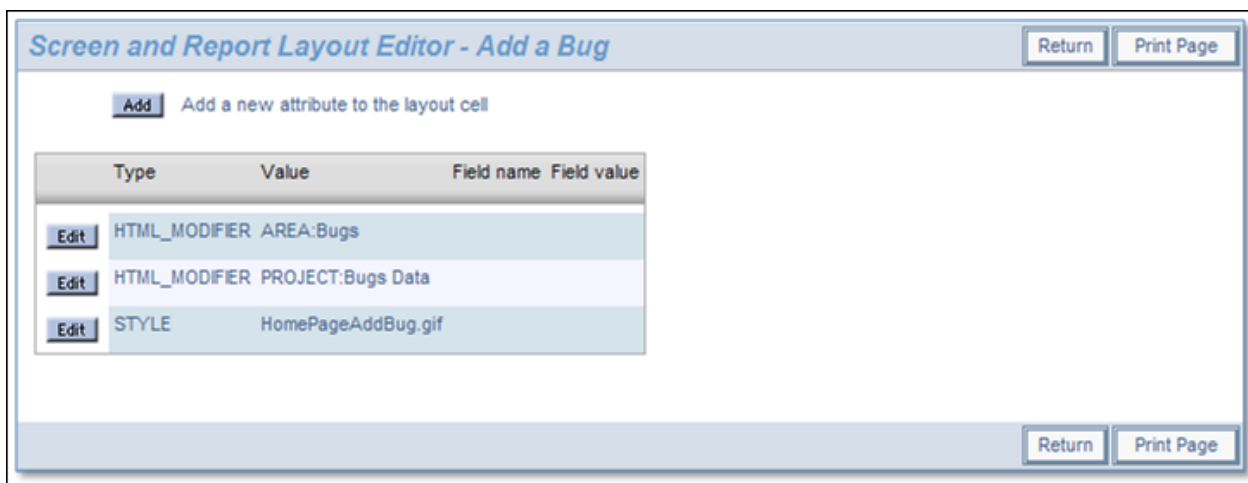After adding the layout, you are able to place the buttons onto the layout, similar to the screen shown here:
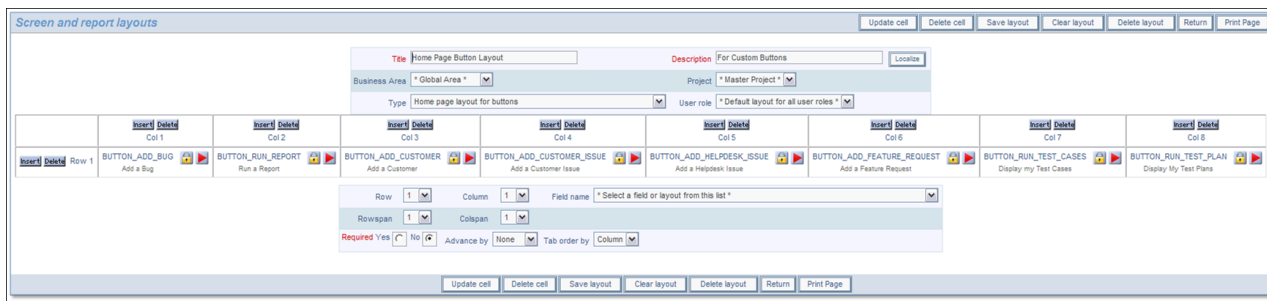
HOME_PAGE Layout

This screenshot creates a single row of shortcut buttons. You may create multiple rows of buttons. You must next add layout cell attributes to each button. These describe the action for the button as follows.

Shortcut Add Buttons

Attribute	Example	Purpose
FIELD HTML MODIFIER	AREA:BUGS	This points the button to add an issue in the **BUG** business area. The name of the business area provided must match the name of an existing business area to which the user has permission. To see the names of the business areas in your installation, go to **Administration ◆ Lists ◆ Business Areas**. This modifier is required.
FIELD HTML MODIFIER	PROJECT:BUGS_DATA	This points the button to add an issue in the **BUGS_DATA** project within the business area specified by the *AREA:* modifier above. This is required. Note that the Project fixed name must match the one into which you want to add data. Normally, this is **not** the Project with an ID of 0. If you are not certain of the Business Area and Project names, go to **Administration ◆ Lists ◆ Project**, select the Business Area and you will see a list of all the Projects within the Business Area.
FIELD STYLE	HomePageAddBug.gif	This is an optional attribute. If it is provided, the button will use this image for the button. The image is situated within the directory pointed to by the behavior setting named IMG_NAV_BAR_HOME. If you do not provide a STYLE attribute, then the button is rendered as a text style HTML button. Images for Home Page Shortcuts are not provided, but can easily be created with any button/image creation program. If ExtraView Corporation is hosting your installation, you do not have direct access to the file system of the server upload these images to the web server.
FIELD HTML MODIFIER	CSS:font-size:20pt;background-color:#FF8888	This is an optional attribute that overrides the built-in style of buttons, allowing you to set your own style. The style is expressed as CSS. Do not use this if you are using the **STYLE** attribute

Example of the attributes for an add shortcut button.

Shortcut Button Attributes

Shortcut Run Buttons

Attribute	Example	Purpose
FIELD HTML MODIFIER	REPORT:My Report	This points the button to run the report named **My Report**. The name must match the title of an existing report which the user has permission to run. This is required. Note that running Container reports and Custom URL reports from shortcut buttons is not supported.
FIELD STYLE	HomPageAddBug.gif	This is an optional attribute. If it is provided, the button will use this image for the button. The image is situated within the directory pointed to by the behavior setting named IMG_NAV_BAR_HOME. If you do not provide a STYLE attribute, then the button is rendered as a text style HTML button. Images for Home Page Shortcuts are not provided, but can easily be created with any button/image creation program. If ExtraView Corporation is hosting your installation, you do not have direct access to the file system of the server to upload these images to the web server.
FIELD HTML MODIFIER	CSS:font-size:20pt;background-color:#FF8888	This is an optional attribute that overrides the built-in style of buttons, allowing you to set your own style. The style is expressed as CSS. Do not use this if you are using the **STYLE** attribute

Example of the attributes for a run report shortcut button.
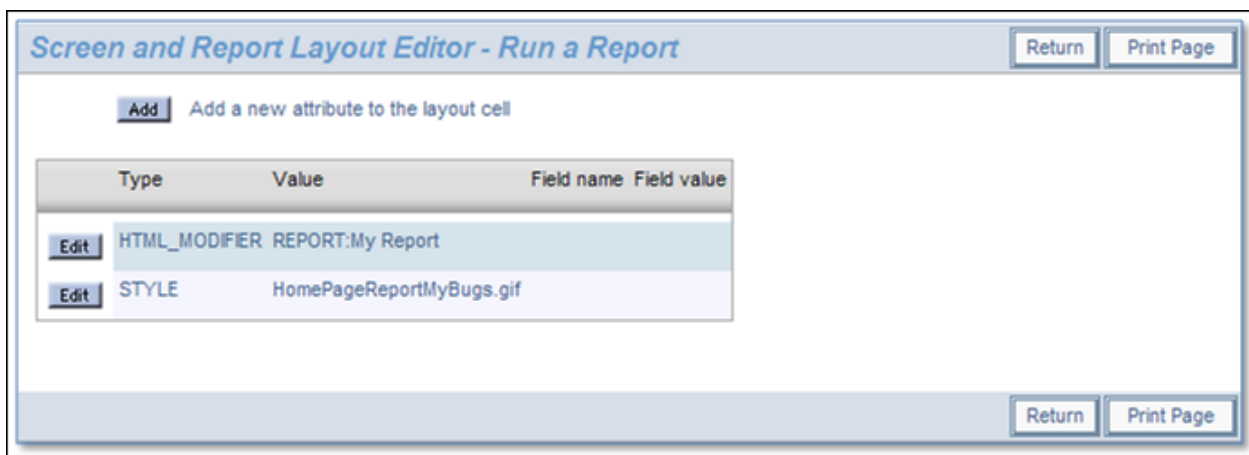
Shortcut Button Attributes

Customize the Home Page

Different users may have different customized Home Pages. The Home Page is composed of the following sections:

- A navigation bar with the following features:
  - Key ExtraView functions available as buttons. There are Home, Add, Query, Admin, Help and Sign Off buttons. Depending on the security permissions, not all these buttons may be visible to all user roles
  - An input area offering quick access to any issue for which the user knows the ID
  - **Account details** – the user's account settings and options are available from a link on the user's name if the user has permission to the security permission key named CF_PERSONAL_OPTIONS
  - **User role** – a list allowing the user to alter their role. This only appears if the user has the ability to change their role and they have permission to the security permission key named CF_ALLOW_CHANGE_ROLE
  - **Business area and project** – offers a list where a new default area and / or project may be selected. Once again, this is only displayed if the user has permission to change their Business Area or Project. The appearance of this entry on the navigation bar is controlled by the security permission keys named CF_AREA and CF_PROJECT
  - A list of reports that the user can run by simply selecting a report from the list. Note that both public reports and the user's personal reports are available. Some report types may not be run from this list. Access to this menu is controlled by the security permission key named SR_MENUBAR_REPORTS
- A sign on message, predefined by the system administrator

- An optional dashboard, created with user custom code, turned off and on for each user role with the security permission key named SR_DASHBOARD_ON_HOME_PAGE
- An optional search box for a knowledge base, turned off and on for each user role with the security permission key named SR_KB_ON_HOME_PAGE
- An optional set of shortcut buttons, created with a special layout type named HOME_PAGE.
- A list of up to three predefined reports
- A section that allows the user to choose which reports are seen on their Home Page. In addition, they can run any report they have permission to see. This report will appear in a separate window. This section will appear if the user has permission to use the Query screen and is controlled by the security permission screen named SR_SET_HOME_PAGE_REPORTS. You can use this facility to either allow a user to select their own Home Page reports, or to only allow the administrator to select the Home Page reports for any user.

Note: The Home Page will refresh automatically, according to the time (in seconds) set in the behavior setting named HOME_PAGE_REFRESH_SECONDS. This is found on the **Reporting & Query Settings** administration menu. Care should be taken by both the administrator and by users in composing their Home Page. Performance of the Home Page is essential in making sure users see good performance from the system. Suggestions for consideration are:

- If you or users have many reports (say more than a few hundred), then it is advised to turn access to the permission key named SR_SET_HOME_PAGE_REPORTS off
- Dashboard reports should only be configured that are extremely fast in execution
- Individual reports and charts should only be placed on the Home Page if they are also fairly fast. For example, summarizing a significant quantity of data in a chart on a daily basis over a year may take a lot longer and not offer much more information that the same chart displaying the data by month
- Keep the value of the behavior setting named LIMIT_HOMEPAGE_QUERY_ROWS to 20, thereby ensuring users do not repeatedly report hundreds or thousands of rows of data to their Home Page
- Set the value of MINIMUM_SEARCH_FIELDS to around 2 or 3, thereby ensuring that users must choose a number of filters in addition to keywords. This will help speed reports as filters provide indexes to the data tables, thus speeding the queries
- Consider setting the value of HOME_PAGE_REFRESH_SECONDS to more than 900, thus only refreshing the user's Home Pages on a less frequent basis.

LOGO

Bill Smith | Administrator
Bugs

Navigation Bar

Home | Add | Query | Admin | Help | Sign Off | Search for ID # | Go | Reports

Print Page

ExtraView Home

Quick Start
- Add your users
- Populate your lists
- Start adding issues

Resources & Documentation
- Quick Start Guide to Administration
- Frequently asked questions
- Free daily training
- Documentation

Sign On Message

Need More?
- Sophisticated configurations
- Product Tour
- Personalized demonstration
- Pricing

Search the ExtraView Knowledge Base

Product * Any *   Keywords   [Search]

Search Attachments? ☐   [Top Issues]

Knowledgebase Layout

Shortcuts   **Shortcut Buttons**
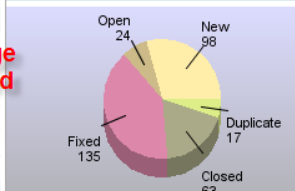
[Add a Feature Request] [Add a New Bug] [Add a New Customer] [Add a New Customer Issue] [View My Issues] [View My Team's Issues] [Customer List] [High Priority Issues]

Home Page Dashboard

Summary of Statuses	
Status	Total
New	98
Open	24
Fixed	135
Closed	63
Duplicate	17
Not Found	19
Grand Total	356

Open Issues by Business Area	
Area	Total
Bugs	9
Test Cases	1
Helpdesk	5
Customer Issues	7
Feature Requests	1
Knowledge Base	1
Grand Total	24

Home Page Dashboard Report

Key Statuses — Open 24, New 98, Duplicate 17, Closed 63, Fixed 135

Open Issues by Priority (chart) — P1, P2, P3, P4

My Open Issues

	ID #	Area	Product	Module	Last Modified	Priority ▲	Severity	Title
View Edit	10323	Bugs	Tracker	GUI	8/11/08 9:51 PM	P 1		new java file changes
View Edit	10226	Bugs	Tracker		8/11/08 9:51 PM	P 1	Low	Provision of TNCP problem
View Edit	10557	Bugs	Tracker	Database	4/17/09 9:08 PM	P 3		There is a defect in the admin module when a user makes an error
View Edit	10552	Bugs	Tracker	Database	8/14/09 9:51 PM	P 3	Low	The box label may be off center
View Edit	10526	Helpdesk	Tracker		8/11/08 1:03 PM	P 3		Authorization to provide a new telephone extension for new employee - Frank Bell
View Edit	10353	Customer Issues	Tracker	Processor	1/8/09 4:05 PM	P 3		Need to know how exclusive or's are processed internally
View Edit	10293	Customer Issues	Tracker		1/8/09 4:05 PM	P 3	Low	Customer reports a missing item in the most recent delivery
View Edit	10285	Customer Issues	Tracker Enterprise		1/8/09 4:05 PM	P 3	Low	Customer called to say he did not receive the spare part sent on 4/3/05

Home Page Column Report

Count: 8 records

Records 1 to 8 of 8

Bugs by Priority by Status

		Priority				
Status		P 1	P 2	P 3	P 4	Total
	New	4	5	13		22
	Not Yet Tested			1		1
	Open	4	2	2	1	9
	Fixed	11	21	23	6	61
	Closed	3	6	12	6	27
	Duplicate		3	3	1	7
	Not Found	1	2	4	1	8
	Total	23	39	58	15	135

Home Page Matrix Report

Select home page reports   **Home Page Report Selectors**

Public:Home Page Dashboard   Public:My Open Issues   Public:Bugs by Priority by Status

Sample Home Page screen

Custom Coding Extensions

If ExtraView Corporation is hosting your installation, you do not have direct access to the file system of the server to configure, alter or use this feature without contacting ExtraView support. One of ExtraView's most powerful features is the ability to extend and alter

its inbuilt functionality, by adding your own "user custom" code. This topic is covered in more detail in the <u>User Custom Guide</u>. It is covered briefly within this document.
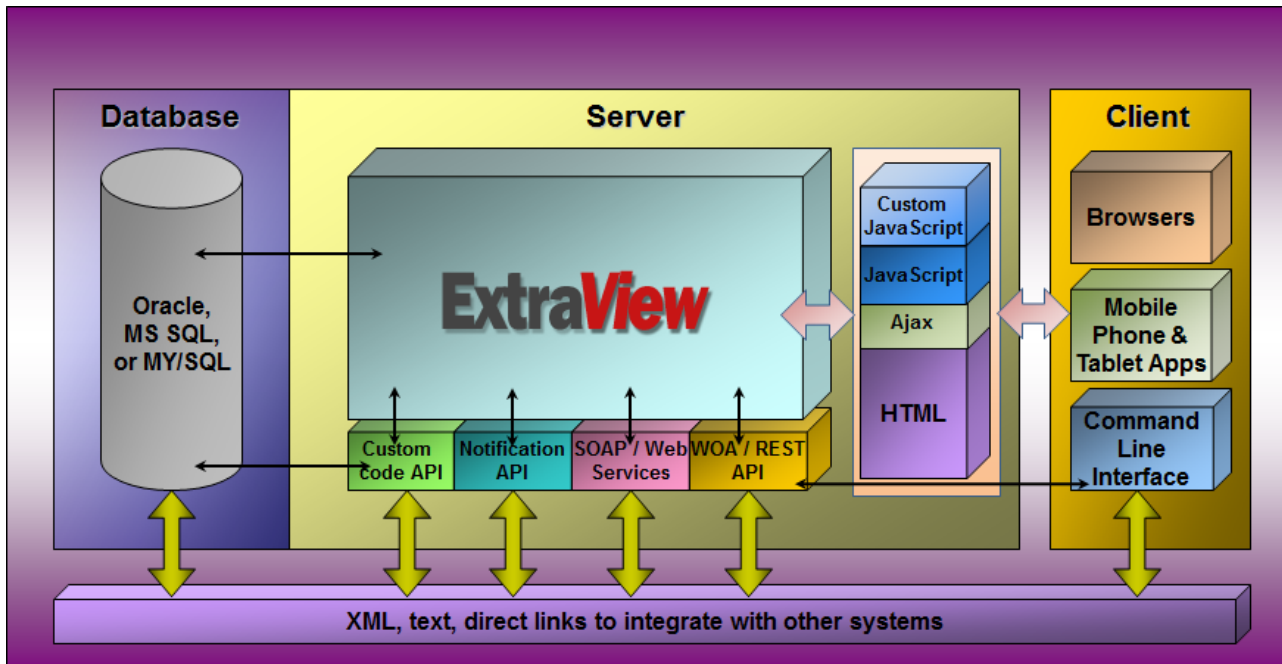
The ExtraView architecture

Java custom coding

Java custom coding supplements or alters the behavior of the standard ExtraView code. Within many of ExtraView's inbuilt functions, a code exit takes place to a method within a user custom module. If no user custom code exists within these methods, ExtraView continues its operation. If user custom code exists within the method, this will be executed. Within the UserCustom.java class provided within ExtraView, the programmer can inherit from this class and override the methods of interest. A sample of places where user custom coding can be inserted is:

- Before and after the display of an object, such as a screen
- Before and after the screen refresh of an object such as a screen
- To alter the contents of a list of data
- To alter the functionality when a button is pressed, such as the Relationship Group button, the Delete button or the Clone button
- To alter email notification functionality
- Before and after the updating of an issue
- Before the deletion of an issue
- In function calls made through the API

You will require a full Java development environment and JDK to create user custom Java code. It is also recommended that you use an IDE such as Borland's JBuilder or Eclipse.

Note: ExtraView strongly recommends that it be consulted about the user custom coding you wish to implement. ExtraView is a complex environment, and it takes both programming experience with the Java language as well as a thorough understanding of the internal structure of ExtraView to successfully design and build user custom extensions to ExtraView.

Note: If you are experiencing any errors in an ExtraView environment where you have installed user custom coding, we recommend that you check for the presence of the error with the user custom code removed, before you report the issue to ExtraView support.

JavaScript custom coding

JavaScript custom coding functions are typically called from an individual field of the *add* or *edit* screen. The JavaScript can perform many purposes, such as:

- Validation of the entry being made into the field by the user
- Complex validations that involve the comparison of many field values on the form
- Triggers to produce alerts to the user, by popping up messages according to the logic of the function

User custom JavaScript is placed in a single file named UserJavaScript.js. This file exists within your web server tree structure in a directory named /user_javascript.

JavaScript functions are downloaded into the client browser at the time the form is generated on the screen for the user. They are efficient, in that no call to the server is needed when you invoke a function. The code will execute within the browser.

Most JavaScript custom functions are defined within layout cell attributes for a field, using the FIELD HTML MODIFIER. The FIELD HTML MODIFIER calls the JavaScript function that you create in the specified place on the server. For example, if you want to check that an entry in a field named *product_code* on the *edit* screen is always sent to the server in upper case, you may do the following:

- On the field named *product_code* on the *edit* screen, define a layout cell attribute of type HTML Modifier and with a value of *onclick='checkUpperCase()';*
- Create a user custom JavaScript function within the user_javascript file, similar to:

```
<script language=JavaScript>
function checkUpperCase() {
  var s = document.editForm.product_code.value;
  document.editForm.product_code.value = s.toUpperCase();
}
</script>
```

Email Notification

ExtraView offers a number of email notification features that are designed to facilitate maximum flexibility in managing inter-group, cross-functional, and business-to-customer communication. As a general default, ExtraView automatically sends email to the person who Originated an issue, the person who is Assigned to the issue, and the person who is selected as the Owner of the issue. These can be altered or replaced using Email Rules.

Creating a Mailing List for an Issue

There are four types of users for email notification. All these users go through the same validation process as explained in this section.

1. **Interest list users**. These are the users qualified to be in the mailing list based on interest list conditions and subscription. Note that if the interest list uses filters with operators such as *changed to* and *changed from* then users that are added to the interest list based upon the results of these filters, then the mailing list on the *add* and *edit* issue screens will not display these users. This is because these users cannot be evaluated until the issue is updated. Of course, they will receive the notification upon the insertion or update to the issue.
2. **CC Mail users**. These are USER IDs or Email addresses entered manually into CC Mailbox
3. **Business Rule users**. Business rules may add additional users to the email list before it is processed. These users will be added unconditionally to the email list.
4. **Built-in users**. The following users are built-in users and they are automatically qualified to be in the mail list. They are also called standard email recipients. You may suppress this list with the behavior setting named SUPPRESS_STANDARD_EMAIL_LIST by setting it to a value of YES.

Data Dictionary Field Name

OWNER

ASSIGNED_TO

CONTACT

ORIGINATOR

LAST_CHANGE_USER

Previous OWNER

Previous ASSIGNED_TO

Previous CONTACT

Previous ORIGINATOR

      Release ASSIGNED_TO

      Release OWNER

      Module ASSIGNED_TO

      UDF list OWNER

      Module default OWNER if the behavior setting named EMAIL_MODULE_OWNER_ALWAYS = YES

An overall test is made to determine whether to send email notification in the following steps:

- The behavior setting named **EMAIL_NOTIFICATION** must have a value of YES
- The setting named **EMAIL_DIRECTORY** must have a valid path into which emails may be written, and the **BatchMail** task must be running
- Correct values must have been established for the settings named **EMAIL_FROM_USER_ID**, **EMAIL_FROM_USER_NAME**, **EMAIL_ADMINISTRATOR_ID** and **EMAIL_ADMINISTRATOR_USER_NAME**
- If the field named NOTIFICATION_GENERATE_EMAIL is not contained within the *add* or *edit* screen layout or the user role lacks permission to the field, then the value of the behavior setting named **GENERATE_EMAIL_BOX** is used to determine whether to generate the email notification. This setting must then have a value of CHECKED for notification to be sent.

When an issue is created or updated, ExtraView makes a determination of who is to be notified from the users in the above list, and the rules to process the mail notification are based on the following criteria.

- Check whether **EMAIL_CUSTOMER** = YES or NO. This comes from the check box on the *add* or *edit* screen. The value is always YES for Custom Email
- Check whether the behavior setting named **SUPPRESS_STANDARD_EMAIL_LIST** = YES or NO. Note that this setting has a default behavior setting value, but it may be controlled PROJECT by PROJECT on the List Management screen for each Project.
- Check whether each user's email address and alternative email address is valid – **VALID EMAIL ADDRESS** = YES or NO. The program checks whether each email address has @ sign within the address and has a "." at the 3rd or 4th position from the end. Examples of valid addresses are john@extraview.com and john@extraview.com.jp. If the address is not valid, processing does not halt, but the invalid address is dropped from the notification list
- A check is made to see whether the issue is available for viewing by each user. This is the **PRIVACY GROUP TEST** which will result in PASS or FAILED. This test is performed by ExtraView. The result is always PASS if the email being generated is an ad hoc email. The basic test results in PASS if the issues' privacy group is PUBLIC or PRIVATE or the user being tested is in the issue's privacy group
- The role of the user being tested is examined. The test results in PASS or FAILED. The result is always PASS if the behavior setting named LIMITED_USER_ROLE is not defined or is blank. The result is PASS if the user's current role is different from the value of the behavior setting LIMITED_USER_ROLE. The result is FAILED if the user's role is the same as that defined in LIMITED_USER_ROLE. If the result was FAILED, then by definition the user is a GUEST account and the following test is made – if the checkbox **EMAIL_CUSTOMER** on the *add* or *edit* screen is checked, then an email to the GUEST user is generated if one of the following conditions is true:
    - It is an ad-hoc mail that is being sent
    - The behavior setting named ALLOW_GUEST_MAIL is set to YES. Note that this behavior setting is not changeable by the administrator and is only available with a special license key from ExtraView. The reason for this is that access to this key could allow all users to bypass the ExtraView licensing scheme and have unlimited use of ExtraView.
- A check is made on whether the user is to **RECEIVE NOTIFICATION AT PRIMARY ADDRESS** = YES or NO. This comes from the user's personal options. This defines the initial address to which notification is sent
- A check is made on whether the user is to **RECEIVE NOTIFICATION AT ALTERNATE EMAIL ADDRESS** = YES or NO. This comes from the user's personal options. This defines the alternate address to which notification is sent
- Next, a check is made of the user's personal option **RECEIVE NOTIFICATION ON OWN UPDATES** = YES or NO. This checks whether the user who is inserting or updating the issue is to be removed from the notification list
- Lastly, a check of the behavior setting named **EMAIL_MODULE_OWNER_ALWAYS** = YES or NO is made. If the check returns YES, then the user ID who is attached to the field MODULE_ID is added to the notification list.

At the end of these checks, ExtraView has a list of the user's who are to be notified. The process then takes each of these users in turn and creates the outgoing email for the user, based upon their role and all security permissions.

Administrator-Controlled Email Features

- Turn system-wide email on or off
- Disable email generation control for user roles
- Control specific notification option for user roles while adding and updating issues. This may be cointrolled on a PROJECT by PROJECT basis

- Enable or disable email to external users
- Optionally assign a module owner who will automatically receive notification upon a new or updated issue
- Optionally set a product email address to notify a product manager upon a new or updated issue
- The administrator can set up conditions which, when met, will cause an action to occur. Most often, this is used to escalate issues when they have remained in a specific status for longer than your process requires
- The email subject line can be tailored to contain text or any fields within the current issue
- Optionally show the email recipient the CC list for the notification
- The administrator can define email templates as either text or HTML. These templates can contain as much or as little data as is required for the standard notification of issues to ExtraView users
- The administrator can optionally define a range of email templates that can be used to communicate with customers or users, merging data from the current issue with pre-defined text. These templates can be made available on a user role basis

See also the section on Scheduled Reports for information on how the administrator may control the email delivery of reports to users.

Email Options

Email addresses

Each user may have two email addresses within ExtraView, a primary email address and an alternative email address. The primary email address is maintained on the Personal Details tab of the user account maintenance screen. The alternative email address and the controls for receiving email are on the Notification Options tab of the user account maintenance screen.

The administrator or the end user can select options for the end user to receive email at their primary and alternative email addresses independently.

If the behavior setting named CHECK_EMAIL_ADDRESS_FORMAT has a value of YES, then the format of each email address is checked at the time it is entered or updated and an error message is returned if the format is not valid.

Turning System-Wide Email On or Off

1. From Administration menu, under the **Email Notification** tab, click on **Email Settings**.
2. Scroll down until you see EMAIL_NOTIFICATION.
3. Click **Edit** and change the value to YES, in order to turn on Email Notification, or set the value to NO to turn off Email Notification system-wide.
4. Click the **Update** button.

When EMAIL_NOTIFICATION is set to NO, no warnings are issued to users who enter or update issues, or generate emails in any way.

Disabling Access to the Email list for Specific User Roles

You can control access to the mailing list with the security permission keys named PR_ADD_PROBLEM.NOTIFICATION and PR_RESOLUTION.NOTIFICATION. If these keys are turned off for a combination of user role, business area and project, then the user will not have control over, or visibility of the email notification section of the *add / edit* screens.

Notification Permission Keys

These keys have both PR_ADD_PROBLEM and PR_RESOLUTION entries to control both the *add* and the *edit* screens.

Key	Purpose
NOTIFICATION	This is the master control for the display of the notification section on *add* and *edit* screens. You must have WRITE permission to these keys to observe the notification section on *add* and *edit* screens. Note that turning off this key is independent of the values of the fields and the security permission values of these keys. Just turning this display off does not alter the value of the EMAIL_SWITCH key. You can either set this key off as well, or use a business rule to inhibit the sending of email notifications from any screen within any Role / Business Area / Project combination.
CC_EMAIL	This key controls the visibility of the CC Email text entry box
CC_EMAIL_BUTTON	This permission key controls access to the CC Email button to the right of the CC Email text entry box
EMAIL_CUSTOMER	Giving permission to this key enables or disables access to the checkbox that turns notification on or off

to users in the DEFAULT_USER_ROLE. Note that this checkbox may be set programatically using a business rule

EMAIL_SWITCH	This controls access to the checkbox that has the prompt **Generate Email**
INTEREST_LIST	With this key you control access to the ability to allow users to add users to an interest list for the issue they are adding or updating
MAILING_LIST	This is the control for the list of users who will be notified upon inserting or updating the issue

Assign Module Owners

The System Administrator may assign *Module Owners*, such that changes on issues with given modules produce automatic email notification to the designated owner.

In addition, the module owner may be used to automatically populate the *assigned_to* field on the Add Issue screen, if the behavior setting named **LINK_MODULE_OWNER_ASSIGNED_TO** on the Workflow Settings screen is set to a value of YES. This allows you to create the Add Issue screen in such a way that when a module is selected from a list, the Assigned To is automatically set to be the module owner.

1. From the Administration menu, under the **Lists** tab or from the Data Dictionary, click **Module Names**
2. Locate the *Module* to which you would like to assign an *Owner*, and click the associated **Edit** button

Updating the module owner

* Scroll through the list of available users or else select the owner from the pop up text box
* Click the **Update** button.

Set Product Email Address

The Administrator may set an email address for specific *products*, so that ExtraView users associated with that *product* will receive automatic email notification about product-related issues. This is achieved from the Products list within the **Lists** tab of the administration menu.

Setting the Product email address

Notification Subject Line

ExtraView gives the Administrator the ability to customize the subject line of emails that are automatically sent when inserting and updating issues.

1. From the Administration menu, click on the **Behavior Settings** entry and then choose the **Email Settings** category
2. Scroll down until you see the behavior setting named EMAIL_SUBJECT_TEMPLATE, and click the associated **Edit** button
3. Type in the names of the fields you would like to display. Your custom email subject line can include any field from your ExtraView installation. These values are generated dynamically, based on the particular issue
4. If you would like values in your email subject line you must surround them with "$$". Normal static values can be typed in. See the example below:

Sample Text	Subject Line Output
$$ID$$ - $$SHORT_DESC$$	12345 – Problem with List Entries
$$ASSIGNED_TO$$ ($$PRODUCT_NAME$$; $$MODULE_ID$$) - This is an Email	rlloyd (Product X; Module Y) - This is an Email
Issue # $$ID$$ This is Assigned to $$ASSIGNED_TO$$	Issue # 12345 This is Assigned to rlloyd

Notification History

* Add new comment

There is a special field named NOTIFICATION_HISTORY which can only be placed on layouts with a type of HISTORY. This displays a record of which users were notified when the update to the issue was made. However, if you are using the behavior setting named HISTORY_DISPLAY with values of ABBREVIATED and/or SIDE_BY_SIDE, then the notification history is still shown, as part of each history entry.

CC Email Capability

This feature gives users the opportunity to send a one-time email to people who are not directly associated with a particular issue.

This feature is controlled by the Security Privilege Settings as follows. A different setting can be used for each user role within the system.

Security Key Name	Purpose
PR_RESOLUTION.CC_EMAIL	This controls the appearance of the CC Email entry box on the Edit Issue screen
PR_ADD_PROBLEM.CC_EMAIL	This controls the appearance of the CC Email entry box on the Add Issue screen
PR_RESOLUTION.CC_EMAIL_BUTTON	This controls the presence of a user popup button on the Edit Issue screen. This button gives access to a list of users within ExtraView, who can be mailed
PR_ADD_PROBLEM.CC_EMAIL_BUTTON	This controls the presence of a user popup button on the Add Issue screen. This button gives access to a list of users within ExtraView, who can be mailed

To utilize the CC Email functionality, type email addresses (separated by commas or semi colons) into the CC Email field prior to updating the issue, or select named users from the list by first clicking the people icon. Unless you have a license from ExtraView that allows your installation to send email to non-ExtraView users, email addresses entered must belong to registered and activated ExtraView users. Also note that if there is a privacy group restriction on the issue, the user whose name, ID or email address is entered must have permission to view the issue.

Email notification on the Add or Edit screen

Disabling Email Generation

Disable Automatic Email Generation

Each time a user adds or edits an issue, he or she has the opportunity to halt all email generation by un-checking the **Generate Email** checkbox at the bottom of the Add and Edit screens.

This is controlled by the following security permission keys

Security Key Name	Purpose
PR_RESOLUTION.EMAIL_SWITCH	This controls the appearance of the Generate Email checkbox on the Edit Issue screen
PR_ADD_PROBLEM.EMAIL_SWITCH	This controls the appearance of the Generate Email checkbox on the Add Issue screen

Email notification on the Add or Edit screen

Disable Generation of Email to External Users

Situations often arise where a customer may need to enter an issue, but you may not want the customer to see all of the different state changes that the issue goes through. When you add or update an issue, you have the option of halting email to external users. This is controlled by security permission keys for each user role as follows:

Security Key Name	Purpose
PR_RESOLUTION.EMAIL_CUSTOMER	This controls the appearance of the Include Guests checkbox on the Edit Issue screen
PR_ADD_PROBLEM. EMAIL_CUSTOMER	This controls the appearance of the Include Guests checkbox on the Add Issue screen

At the bottom of the Add and Edit screens you will see something similar to the screenshot below. The default setting is not to send email to external users. Note that the term "Customers" is replaced by your organizations own name for guests or customers.

Email notification on the Add or Edit screen

If you would like your external users to receive an email, click the **Include Customer users in notification** check box and continue to update the issue.

BatchMail Task

If ExtraView Corporation is hosting your installation, you do not have direct access to the file system of the server to configure, alter or use this feature without contacting ExtraView support.

When a notification is generated by ExtraView, it is written to a directory on the filesystem of the server. The location of the directory is controlled by the behavior setting named EMAIL_DIRECTORY.

This directory is managed by the BatchMail task or process. With ExtraView 6.0 and beyond, this task may be managed within the ExtraView administration utility to configure BatchMail. Previous to version 6.0, the BatchMail task was managed external to ExtraView.

For backwards compatibility, the task may still be operated externally. Please see this section in this guide for information on how this utility works.

Locale Handling

Notification and the User's Locale

When a user receives an email notification, they will receive it in their current locale irrespective of the locale of the person who updated the issue. For example, if you are on the interest list for an issue, and your locale is Japanese, you will receive the notification using Japanese as the language. If your locale is English, you will receive the notification in English.

User-Controlled Notification Features

- Turn on and off notification of user's own updates
- Select incoming email format to be text or HTML.
- Select from a fuller or briefer form of notification when receiving text
- Disable automatic email generation
- CC Email
- Send pre-formatted email, using templates set up by the administrator, to communicate information to customers or other users

Note: This guide describes the user-controlled email features along with the administration features, as a review of this functionality here helps to provide a more complete overview of email functionality in ExtraView.

Notify of Own Updates

This feature gives an individual user the option to disable automatic email to their self on issues that they add or update. The administrator can also apply this option on behalf of any user.

- As an individual user, click on your account name in the title bar to edit your account details, or from the User tab on the Administration menu click on the **User Account Maintenance** option and edit the user whose account you want to modify. Click on the **Notification Options** tab.

  A screen similar to the one below will appear:

Change a User's Details screen

- Click on the radio buttons by **Notify on own updates** to toggle the option on and off.
- Click the **Update User** button.

Select Email Format

This feature gives an individual user the option to choose whether to receive notifications in text or HTML formatted email. The administrator can also apply this option on behalf of any user. The choices are to display incoming email in one of four formats: HTML, Plain Text (full) , Plain Text (brief) and Plain Text (very brief).

Type	Purpose
HTML	Notification is sent using the layout named EMAIL_FULL, and is delivered in an HTML format
Plain Text (full)	Notification is sent using the layout named EMAIL_FULL, and is delivered as plain text format
Plain Text (brief)	Notification is sent using the layout named EMAIL_BRIEF and is delivered in a plain text format
Plain Text (very brief)	No layout is used for the notification, but a plain text format email is delivered, only showing the fields that changed in the insert or update operation of the issue. If a field value was modified, it is preceded with a * character. If a field value is inserted, it is preceded with a + character.

- As an individual user, click on your account name in the title bar to edit your account details, or from the **User** tab on the Administration menu click on the **User Account Maintenance** option and edit the user whose account you want to modify. Click on the **Notification Options** tab.

  A screen similar to the one below appears:

Change User Details screen

- The default email format is HTML. If you would like to see your email notifications in another format, select the desired format from the Email format list.
- Click the **Update User** button.

Below are examples of the different email formats:

Open [ExtraView-36474]-[P 3]-Execute notification using opt in/out rules, per new Interest List and E...

File   Edit   View   Insert   Format   Tools   Actions   Help

Reply | Reply to All | Forward |

From: ExtraView Support [support@extraview.com]   Sent: Fri 2/24/2006 11:08 AM
To: carl@sesame.com
Cc:
Subject: Open [ExtraView-36474]-[P 3]-Execute notification using opt in/out rules, per new Interest List and Escalation Rules

Edit	Issue #	Created	Last Modified
	36474	February 8, 2006 11:38:51 AM PST	February 24, 2006 11:07:28 AM PST
Area	**Status**	**Days Open**	**Changed by**
Defects	Open	16	Allan Rofer
Category	**Cust. Sup. Issue ID #**	**Customer**	
Feature Enhancement		Vodafone	
Title			
Execute notification using opt in/out rules, per new Interest List and Escalation Rules			
Product	**Module**	**Originator**	**Assigned To**
ExtraView	Administration	Dan Fineman	Ali Goksel
Found in Release	**Severity**	**Browser**	**Priority**
4.3.5		All Platforms	P 3
Found In Build #	**Database**	**OS**	**Duplicate Bug ID**
	All		
Requested Release	**Committed Release**	**Branch Status**	**Fixed-in build #**
4.3.6	5.0	Open	
Planned Time (days)	**Planned due date**	**Phase**	**Spec. Ref #**
	April 3, 2006	3	6.2
Description			
Execute notification under new paradigm, involving opt-in / opt-out by users for Escalation Rules and Interest Lists.			

cc: Carl Koppel ; Allan Rofer ; Cindy Norman ;
Ali Goksel ; Dan Fineman

HTML Email in Microsoft Outlook

```
    Subject: Open  [#19419]: Category doesn't seem tied to Product anymore
       Date: Wed, 2 May 2001 13:57:53 -0700 (PDT)
       From: "ExtraView" <extraview-user@customer.com>
         To: <support@sesame.com>




ExtraView Notification for Bug # 19419

*Synopsis: Category doesn't seem tied to Product anymore

Link: http://www.extraview.net/extraviewsql/SE_Signon.FrameSet?p_case_id=19419

*Product: Dev Tools
*Originator: rlloyd
*Owner: rlloyd
*Changed By: dwong
*Priority: 1
 Severity:
*View: Private
*Last Modified: 02-MAY-2001 13:57
*Category: Software
 Alt ID:
 Created: 02-MAY-01

*Module: ExtraView
 Component:
*Platform: all
 OS:
 Clarify ID:
 Test Case ID:
 Test Case Location:
 Problem Reproducible in SQA:

*Version Open: 3.1.2.1
*Status: Open
 Version Closed:
 Disposition:

*Description:
If you select Product=EV, Category=Software, you will see two modules called
"EV".  Go into Admin...Modules and one is Hardware, one is Software.


 Comments:



--------------------------------------------------------------------------
ExtraView - Copyright Sesame Technology 1999, 2000. All rights reserved.
```

Plain Text (Full)

```
    Subject: Open  [#19419]: Category doesn't seem tied to Product anymore
       Date: Wed, 2 May 2001 13:57:53 -0700 (PDT)
       From: "ExtraView" <extraview-user@customer.com>
         To: <support@sesame.com>



ExtraView Notification for Bug # 19419

*Synopsis: Category doesn't seem tied to Product anymore

Link: http://www.extraview.net/extraviewsql/SE_Signon.FrameSet?p_case_id=19419

*Product: Dev Tools
*Originator: rlloyd
*Owner: rlloyd
*Priority: 1
 Severity:
|

----------------------------------------------------------------------------
ExtraView - Copyright Sesame Technology 1999, 2000. All rights reserved.
```

Plain Text (Brief)

Note: These standard email formats can be modified using the layout editor and modifying the EMAIL_FULL or EMAIL_BRIEF layouts.

Adding Issues with EVMail

Many organizations want to use email to submit new issues to ExtraView, either from their own employees or from their customers. ExtraView enables this capability via a task named EVMail. This task may be configured to allow ExtraView to receive and process emails that both insert and update issues. See the section entitled Email Poller Task for instructions on how to configure the EVMail task. EVMail can perform the following tasks:

- Add new issues to your database
- Update existing issues

See the section entitled [Configuring EVMail](#) for instructions on how to setup and configure the EVMail task.

Workspace Configuration

Workspaces have several configuration points of which you should be aware.

Security Permissions

MENU_WORKSPACE	This key provide the overall control of access to workspaces, role by role. If the user role does not have write access to this key, the **Workspace** navigation bar entry does not appear, and there is no way for the user to access workspaces
CF_ALLOW_RETURN_FROM_WORKSPACE	Read access to this permission key places an entry in the workspace menu with the title **Close Workspace**. Without this permissoin, the user cannot return to the traditional ExtraView user interface. If the user's profile has selected their start page as **Worskpace** and this permission is not provided, the user is locked into the workspace environment
SR_PUBLIC_WORKSPACE	If the user role has write permission to this key, they are allowed to save & delete public workspaces created by other users. Without this permission, the user may only delete workspaces that they have created.

Saving Workspaces

Each workspace is governed by three factors:

- Its user role for which it is created
- Its Business Area within which it is saved. This becomes the default Business Area for operations within the workspace
- Public or Private. If you create a public workspace, other users may use this, and if they have write access to SR_PUBLIC_WORKSPACE, they may overwrite and/or delete it. If the workspace is private, only the owner has permission to access it.

There is no practical limit to the number of workspaces that can be configured, although it is recommended that you do not have too many public workspaces, as users may be confused as to the different purpose of each saved workspace.

LDAP & SSO Servers

If ExtraView Corporation is hosting your installation, you do not have direct access to the file system of the server to configure, alter or use this feature without contacting ExtraView support.

ExtraView may be configured to work with both Lightweight Directory Access Protocol (LDAP) and Single Sign On (SSO) servers. LDAP servers include Microsoft's implementation named Active Directory. It is not necessary to configure both LDAP and SSO at the same time, although this can be done.

We strongly recommend that you have access to a resource that is skilled in administering LDAP and/or SSO to set up these features.

You may connect directly to an LDAP server, or you may connect to an LDAP server via SSO. Typically, it is slightly easier to configure the combination of LDAP and SSO servers, as opposed to configuring only an LDAP server.

This section discusses connecting ExtraView with both SSO and LDAP first, and then discusses a direct connection to LDAP without SSO.

Neither LDAP nor SSO is part of ExtraView, but are separate applications that ExtraView may integrate with. There are many "flavors" and implementations of both LDAP and SSO and they may be configured in many different ways by different organizations. While ExtraView may connect to and use LDAP and SSO, the configuration is often different from one installation to another. ExtraView's professional services team can help with the integration of ExtraView to your LDAP and SSO servers, but this is not part of the standard installation, and may be a separate, chargeable event.

Setting up LDAP and SSO implies that you will set up the configuration in both behavior settings and within the Configuration.properties file external to ExtraView.

Note that there is also a hybrid authentication mode, where ExtraView can authenticate against an LDAP server and also use its own internal authentication mechanism. This is ideal in an environment where you want to use the LDAP server for internal users, but use ExtraView's authentication for your customer users.

LDAP Synchronization

There are two background tasks that can be used to enhance the LDAP server performance. Only one of these should be configured at one time, and for most purposes, this should be the task titled **LDAP Background Task** and not the **LDAP Synchronization Task**.

The background task configuration is described on the page here.

SSO Connections

The Single Sign On facility allows another application to control user access to ExtraView. When this is enabled through ExtraView's behavior settings (see above), the SSO application is entirely responsible for the authentication of each user. Once the authentication is complete, the SSO application forwards the authenticated information to ExtraView, and ExtraView automatically signs on the user. If necessary, and subject to any licensing restrictions, a new user will be created within ExtraView.

When ExtraView is configured to use SSO as the user authentication mechanism, the user points their browser to the SSO sign on page. ExtraView extracts the user ID and other pertinent data from the HTTP request header and automatically logs the user into ExtraView – no ExtraView sign on page is displayed. During the sign on process, ExtraView will access all of the user's information in the LDAP Server, assuming this is configured. A new ExtraView user will be created, if the user does not exist. If the user exists in the ExtraView database, their record is updated to ensure synchronization with the LDAP server.

SSO Header Mapping

The "login" HTTP header from SSO contains the authenticated user's information. ExtraView administrative data held in the security_user table defines the headers that are used by ExtraView and where in ExtraView each field is stored. All of these fields

map to the individual fields that contain user data. This mapping is used in conjunction with the LDAP user data, bypassing the need for an administrator to add a new user specifying this information.

In the ExtraView configuration file (Configuration.properties), these fields are mapped to match the host header data as shown in the following example:

```
###############################
## SSO HEADER MAPPING      ##
###############################
SSO_PRIMARYKEY = USER_NAME
SSO_SURNAME = SURNAME
SSO_GIVENNAME = GIVEN_NAME
SSO_EMAIL = EMAIL_ADDRESS
SSO_STREET = STREET
SSO_CITY = CITY
SSO_STATE = STATE
SSO_POSTALCODE = POSTALCODE
SSO_COUNTRY = COUNTRY
SSO_PHONE = TELEPHONE_NUMBER
SSO_MOBILE = MOBILE_NUMBER
SSO_PAGER = PAGER_NUMBER
```

As well as the fields such as USER_NAME and CITY in the example above, you can also map the user defined fields in the ExtraView security_user table. There are five of those, named USER_FIELD_1 through USER_FIELD_5.

The SSO_DN_USER_ID_ATTRIBUTE

SSO_DN_USER_ID_ATTRIBUTE - short for Distinguished Name User ID Attribute - indicates two behaviors:

1. The SSO header is in Distinguished Name format, e.g., cn=abc,dn=def,ou=ghi
2. The attribute of the user ID within the DN is the value of the SSO_DN_USER_ID_ATTRIBUTE followed by an optional instance number.. For example, if SSO_DN_USER_ID_ATTRIBUTE=cn, then the user ID would be abc based on the example in the previous point.

Another example is:

cn=2055092,cn=Users,dc=dsd,dc=fmcna,dc=com

In this case, SSO_DN_USER_ID_ATTRIBUTE=cn would still work to establish 2055092 as the user_id.

If the DN is:

cn=Users,cn= 2055092,dc=dsd,dc=fmcna,dc=com

then the SSO_DN_USER_ID_ATTRIBUTE=cn#2 would establish the 2055092 as the user_id (the #2 indicates that the second instance of the attribute should be used). Note that cn#1 and cn act the same when used as configuration values for SSO_DN_USER_ID_ATTRIBUTE.

ExtraView API in a SSO Environment

When you require accessing the ExtraView API, and a Single Sign On server is in place, special needs exist to ensure that all API calls are correctly authenticated, and that every access is from an authorized source. ExtraView uses the following logic to establish whether a call to the API is authentic, when a Single Sign On server is in place:

1. The API call is examined and if the parameters user_id and password exist, these are used to authenticate the user
2. If the first step does not result in a valid user, the headers returned from a SSO connection are examined. If these contain a valid user ID and password, these are used to authenticate the user
3. If there is still no authenticated user, and if the behavior setting ALLOW_ANONYMOUS_API_ACCESS is set to YES, then an anonymous connection is established, using the permissions of the role of the user that is set in the behavior setting ANONYMOUS_API_USER_ID.

Direct LDAP Connections

When ExtraView is configured to work directly to an LDAP server, the following functions are enabled, all without any need for custom programming within ExtraView:

- Access to an unlimited number of LDAP fields
- Customized mapping of one or more fields in ExtraView to one or more fields LDAP fields
- Customized pre-population of mapped fields on the ExtraView *Add* and *Edit* Screens
- Customized pre-population of fields through a popup link
- The upsert of data to the ExtraView user table, based on a configuration value. Upsert is a combination of insert and update. If a record exists, it is updated, if it does not exist, it is inserted.

Setup Access Details to your LDAP Server

Normally, you will set up the *LDAP Background Task* within the *Task Manager* administration utility before beginning configuration of your LDAP server.

There are several behavior settings that control access to your LDAP server. These should be configured first, to provide the basic connection from ExtraView to your LDAP server. These settings are:

Setting	Purpose
LDAP_USER_LOOKUP	This is the basic on/off switch for LDAP. Set this to YES if you intend to use an LDAP server for user lookup
LDAP_DEFAULT_AREA	This is the ID of the business area to associate with a new user who is added to ExtraView via the LDAP server. To find this number, look at the data dictionary field AREA, and click the **List** button. By each business area you will see the ID. It is essential that you provide a valid entry for this setting.
LDAP_DEFAULT_PROJECT	This is the ID of the project within the business area to associate with a new user who is added to ExtraView via the LDAP server. To find this number, look at the data dictionary field PROJECT, and click the **List** button. Select the business area and you will see a list of the IDs associated with each project. Normally you will choose the *DATA* project within the business area. It is essential that you provide a valid entry for this setting.
LDAP_HOST	This is the URL to your LDAP server. It is essential that you provide a valid entry for this setting.
LDAP_MANAGER	The "Security Principal" or user accessing LDAP
LDAP_PSWRD	The password to the LDAP server
LDAP_ROOT	The root directory of the LDAP server or search base, e.g. ou=blahWorker, o=blah.com. It is essential that you provide a valid entry for this setting.
LDAP_ALLOWED_STALE_INTERVAL	The minimum number of minutes between LDAP upsert operations on a single user. This stops ExtraView accessing the LDAP server with every operation that requires user information in order to optimize performance. After this interval, a fresh check of the LDAP server is made. The LDAP Background Task should be running for this to work
LDAP_SEARCH_FILTER	LDAP filters may be defined in the behavior setting or may be defined in the Configuration.properties file. The behavior setting takes precedence over the Configuration.properties entry. LDAP filters are defined in RFC 2254. As an example, if you wanted to add a filter to only retrieve records with mycompany within the email address, you could set this as the filter: (mail=*mycompany*) The parentheses are essential.
LDAP_UPSERT	This setting controls upserting LDAP information to ExtraView. THe possible values are YES or NO. When this value is NO, LDAP will not be used to add or update ExtraView user information. A valid LDAP user that is not already an ExtraView user will not be able to log in to ExtraView
LDAP_UPSERT_DEFAULT_USER_ROLE	If this setting contains a valid role, then this is the role a user is given when the LDAP upsert takes place. If this setting does not contain a value, then the role defined in the behavior setting named LIMITED_USER_ROLE is used instead.
LDAP_USER_LOOKUP	When this behavior setting is set to YES, whenever a user performs an operation to lookup the details of another user, ExtraView will ask the LDAP server for the

information. At the same time this is done, the information for the user within
ExtraView, will be synchronized with the information within the LDAP record

Setting LDAP Fields in the Configuration.properties File

In the configuration file of the application server (Configuration.properties), there is a parameter named LDAP_FIELDS. These are
the meta-names of the mappings to be used with LDAP. These must be comma separated, listed on one line. The following fields are
mandatory and must be provided as the fields for the upsert operation:

LDAP_FIELDS =   LDAP_PRIMARYKEY,

LDAP_SURNAME,

LDAP_GIVENNAME,

LDAP_COMMONNAME,

LDAP_EMAIL,

LDAP_STREET,

LDAP_CITY,

LDAP_STATE,

LDAP_POSTALCODE,

LDAP_COUNTRY,

LDAP_PHONE

Other fields may be added. They should follow the same naming convention. These represent the fields for which information will be
retrieved. For example, you might add more fields such as:

LDAP_MOBILE,

LDAP_PAGER,

LDAP_COMPANYNAME,

LDAP_DEPARTMENT,

LDAP_TITLE

LDAP_ADDITIONAL_EMAIL

LDAP_ADDRESS_LINE2

LDAP_PHONE

LDAP_HOME_PHONE

LDAP_COMPANYNAME

LDAP_FAX

LDAP_USER_DEFINED_1

LDAP_USER_DEFINED_2

LDAP_USER_DEFINED_3

LDAP_USER_DEFINED_4

LDAP_USER_DEFINED_5

LDAP_USER_DEFINED_6

LDAP_USER_DEFINED_7

LDAP_USER_DEFINED_8

LDAP_USER_DEFINED_9

                    LDAP_USER_DEFINED_10

Map the Fields in LDAP_FIELDS to Values in the LDAP Directory

These fields are mapped with a 1:1 relationship, similar to the following example:

LDAP_PRIMARYKEY	=	employeenumber
LDAP_LOGINID	=	uid
LDAP_SURNAME	=	sn
LDAP_GIVENNAME	=	givenname
LDAP_COMMONNAME	=	cn
LDAP_EMAIL	=	mail
LDAP_STREET	=	street
LDAP_CITY	=	l
LDAP_STATE	=	st
LDAP_POSTALCODE	=	postalcode
LDAP_COUNTRY	=	postaladdress
LDAP_PHONE	=	telephonenumber
LDAP_MOBILE	=	mobile
LDAP_PAGER	=	pager
LDAP_COMPANYNAME	=	displayname
LDAP_DEPARTMENT	=	department
LDAP_TITLE	=	title

Map the Fields in LDAP_FIELDS to Values in ExtraView

There can be more than one ExtraView field mapped to an LDAP_FIELD. You must prefix the name of the LDAP field meta-name with the characters EV_. For example, LDAP_PRIMARYKEY becomes EV_LDAP_PRIMARYKEY. You should separate multiple values with a comma.

EV_LDAP_PRIMARYKEY	=	USR_ID
EV_LDAP_SURNAME	=	USR_LNAME
EV_LDAP_GIVENNAME	=	USR_FNAME
EV_LDAP_COMMONNAME	=	USR_NAME
EV_LDAP_EMAIL	=	USR_EMAIL
EV_LDAP_STREET	=	
EV_LDAP_CITY	=	USR_CITY
EV_LDAP_STATE	=	USR_STATE
EV_LDAP_POSTALCODE	=	
EV_LDAP_COUNTRY	=	
EV_LDAP_PHONE	=	USR_PHONE,USR_PHONE2
EV_LDAP_MOBILE	=	USR_MOBILE
EV_LDAP_PAGER	=	

```
EV_LDAP_COMPANYNAME        =

EV_LDAP_DEPARTMENT          =   USR_DEPT

EV_LDAP_TITLE               =
```

List Fields to be Pre-populated

Ensure each of these is placed on ONE line:

Add Screen

This is a comma-separated list of ExtraView field names that are to be filled in on *add* screens, from values in the LDAP, and based on a search for a user. Example:

ADD_SCREEN_LDAP_FIELDS = USR_NAME, USR_TITLE, USR_DEPT, USR_EMAIL, USR_PHONE, USR_PHONE2, USR_MOBILE, USR_FAX, USR_CITY, USR_STATE, USR_BUILDING

Edit Screen

This is a comma-separated list of ExtraView field names that are to be filled in on *edit* screens, from values in the LDAP, and based on a search for a user. Example:

EDIT_SCREEN_LDAP_FIELDS = USR_NAME, USR_TITLE, USR_DEPT,USR_EMAIL, USR_PHONE,USR_PHONE2

Searching on the Add and Edit Screens

This is a comma-separated list of ExtraView field names whose values are to be used as the search key for filling the field values on the *add* or *edit* screens. Example:

LDAP_USER_FIELD_NAMES = USR_NAME, USR_TITLE

Specify if you want to Update ExtraView User Information with the Latest Information from the LDAP Server

If you specify YES, then when each user signs in to ExtraView, the code will upsert that user's information to the ExtraView user table, directly from the LDAP directory.

LDAP_UPSERT = YES

If LDAP_UPSERT is NO, a user will not be able to sign on to ExtraView if they have not already been added to the system. The default value for this behavior is YES, so new users will be upserted and will be signed on.

Extending the Query Capability

When you have configured your LDAP connection to ExtraView, and you click on the search icon by a user field, a popup allows you to search the LDAP directory for users and to return one or more user's details into the form on the screen. It is sometimes useful to configure the fields used for the query.

This is accomplished with the LDAP_SEARCH_FIELDS property within the Configuration.properties file. This is a comma-separated list of LDAP field names. The LDAP search may be filtered by these additional attributes.

For example, entering the following on a single line within the Configuration.properties file:

LDAP_SEARCH_FIELDS = SURNAME,
GIVENNAME,
PRIMARYKEY,
PHONE, COMPANYNAME

extends the search for users on user pop-ups to use the PHONE and the COMPANYNAME fields in addition to the standard fields.

User Name Lookup

There is an optional field within the Configuration.properties files named LDAP_EDIT_FIELDS_CHANGE_ON_REFRESH. When there is a parent user field mapped to an LDAP field and this is placed on an *edit* screen and this field is set to a value of NO, the child fields that are mapped are not altered when the parent field is changed. This allows the values of a record on the LDAP server to be placed on an *edit* screen and not be altered once they have been read from the LDAP server. The default value for this property is YES, allowing the child fields to be refreshed when the parent is altered.

When LDAP is configured for user lookup, and LDAP_LOGINID is configured to map to the same LDAP name as LDAP_PRIMARYKEY, the values returned for an LDAP search will contain the mapping: LDAP_PRIMARYKEY --> SECURITY_USER_ID and LDAP_LOGINID_KEY --> LOGIN_ID for the user. The primary effect of this is on user custom code that expects the LOGIN_ID to be the value returned for LDAP_LOGINID. Note that this is not the result.

User ID's

Some LDAP installations are configured to support User ID's that contain space characters. ExtraView does not support space characters within a User ID. It is considered bad practice to use User ID's that contain space characters. User ID's that contain space characters on the LDAP server cannot be mapped to User ID's in ExtraView.

Upsert of Users

If the Configuration.properties file contains the entry LDAP_UPSERT = YES, then you may specify which fields within ExtraView are updated with the upsert operation. The entry LDAP_UPSERT should be followed with an entry named UPSERT_LDAP_FIELDS. This will include all or a subset of the fields in the following list:

UPSERT_LDAP_FIELDS = LDAP_COMMONNAME, LDAP_GIVENNAME, LDAP_SURNAME, LDAP_EMAIL, LDAP_ADDITIONAL_EMAIL, LDAP_STREET, LDAP_ADDRESS_LINE2, LDAP_PHONE, LDAP_HOME_PHONE, LDAP_MOBILE, LDAP_PAGER, LDAP_COMPANYNAME, LDAP_POSTALCODE, LDAP_STATE, LDAP_CITY, LDAP_COUNTRY, LDAP_FAX, LDAP_USER_DEFINED_1, LDAP_USER_DEFINED_2, LDAP_USER_DEFINED_3, LDAP_USER_DEFINED_4, LDAP_USER_DEFINED_5, LDAP_TITLE

If the UPSERT_LDAP_FIELDS statement is not included then all the fields are assumed to be updated upon the upsert operation.

Timeouts

It is important that the connection from ExtraView to the LDAP server is extremely fast. A poor LDAP connection will introduce unacceptable performance problems into the environment. To guard against LDAP connections that do not return information in a timely manner, there are two timeouts:

TCP connection timeout. If there is general network slowness, there is a two-minute timeout period.

LDAP read timeout. This is set to one minute, and guards against a response not being received from the LDAP server. Note that Java 1.6 (or greater) is required for this timeout to work.

Pre-population of Fields from the LDAP Server

Assuming that you have set up the above configuration parameters, when you load the ExtraView *add* screen, the fields you selected for ADD_SCREEN_LDAP_FIELDS in Configuration.properties will pre-populate with values from the LDAP directory, according to the LDAP mappings configured in the Configuration.properties file.

In the example below, the circled fields are all pre-populated from the LDAP entry based on the current user (Campbell, Rob).

Example of fields pre-populated from an LDAP server

Popup Link Configuration

After you have decided which fields to put on your *Add* and *Edit* screen layouts, you can pick one (usually a primary identifier such as the user's name) to have a URL link popup window beside it. This popup window will allow for dynamic searching of the LDAP directory. It will also populate multiple fields on the *Add* or *Edit* screen with the values for the primary identifier. ExtraView fields will be populated from the LDAP directory based on the mappings you set up in Configuration.properties.

First, configure the link with the appropriate URL, within the Data Dictionary. Full instructions for this are in the Data Dictionary section of this administration guide.

Data Dictionary entry where the popup link is configured

- Set Display as URL to Yes.
- Enter the first part of the URL, exactly as shown. Do not complete the field entry yet

  ?p_action=doDisplay&p_option=security.SearchLDAPDisplay

- Append to the URL the list of fields you want populated in the following pattern, but using the fields for your installation:

  &FIELD=USR_NAME&FIELD=USR_TITLE&FIELD=USR_DEPT &FIELD=USR_EMAIL&FIELD=USR_PHONE
  &FIELD=USR_PHONE2&FIELD=USR_MOBILE &FIELD=USR_FAX&FIELD=USR_CITY&FIELD=USR_STATE
  &FIELD=USR_BUILDING

  where the pattern for each field is *&FIELD=DataDictionaryName*

  The final URL will look something like this:

  ?p_action=doDisplay&p_option=security.SearchLDAPDisplay
  &FIELD=USR_NAME&FIELD=USR_TITLE&FIELD=USR_DEPT &FIELD=USR_EMAIL&FIELD=USR_PHONE
  &FIELD=USR_PHONE2&FIELD=USR_MOBILE &FIELD=USR_FAX&FIELD=USR_CITY&FIELD=USR_STATE
  &FIELD=USR_BUILDING

Popup Link Usage

When you click on the popup link beside the primary identifier, ExtraView will open a new window, where you can search for users by typing in search criteria. This is shown in the following screen shot.

Searching for users in the LDAP directory

After using the search criteria, click on one of the results in the ID field. The fields on the parent *add* or *edit* screen will be populated with all the fields you requested.

LDAPS Connections

LDAPS Connections (LDAP over SSL)

The objective of configuring ExtraView for LDAPS is to install the LDAP server SSL certificate into the JVM that is used by the application server hosting the ExtraView web application in order for the client (ExtraView webapp) to be able to authenticate directly to the LDAP server.

Before starting the process to configure ExtraView to work in this mode, ensure the LDAP server has already been configured for secure LDAP with a SSL certificate, and the secure LDAP port has been confirmed to be active. Consult your LDAP server documentation for configuring your LDAP server for secure LDAP with a SSL certificate.

It is beyond the scope of this document and beyond the scope of your ExtraView support team to explain this further, as much depends on your underlying environment. There is good online help available within Oracle's web site.

Java Configuration

1. Download the InstallCert.java file and compile it with your JDK
2. Run the InstallCert program as follows:

   java InstallCert <host>:<port>

where <host> is the hostname of your LDAP server, and <port> is the LDAPS port (e.g., 636) configured on your LDAP server

3. You will see some Java error text:

   javax.net.ssl.SSLHandshakeException: sun.security.validator.ValidatorException: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target

   and be prompted to select a certificate to add to the trusted keystore

4. Select the certificate to add to the keystore and the InstallCert program will display the entire certificate. Also, be aware that the InstallCert program creates the file jssecacerts in the current directory
5. To verify that the certificate has been added, re-run the InstallCert program as in step 2, and instead of the Java error text you should see the following message:

   No errors, certificate is already trusted

6. Lastly, you need to add the jssecacerts file to your JDK by copying it to the JAVA_HOME/jre/lib/security/ directory.

LDAP Connections with SSO

ExtraView can be configured to use an LDAP directory for details of users. This is optional, and is controlled by the same behavior setting that signifies whether SSO is turned off or on (SSO_STATE). See the section on Single Sign On within this guide for more information.

LDAP Server Information

Once again, see above for the ExtraView behavior settings that control the connection. In addition, the ExtraView configuration file Configuration.properties specifies the parameters required to access the LDAP server.

There is an assumption that only a single LDAP server will be configured. The following information also assumes that the administrator is familiar with the configuration of LDAP servers.

In the ExtraView configuration file, the following fields will be mapped to matched the host LDAP data structures:

```
##########################
## LDAP SEARCH MAPPING ##
##########################
LDAP_PRIMARYKEY = primary_key
LDAP_SURNAME = surname
LDAP_GIVENNAME = given_name
LDAP_COMMONNAME = common_name
LDAP_DISTINGUISHEDNAME = distinguishedname
LDAP_EMAIL = email_address
LDAP_STREET = street_address
LDAP_CITY = city
LDAP_STATE = state
LDAP_POSTALCODE = postal_code
LDAP_COUNTRY = country_name
LDAP_PHONE = telephone_number
LDAP_MOBILE = mobile_number
LDAP_PAGER = pager_number
LDAP_COMPANYNAME = your_companyname
```

ExtraView maps the following LDAP fields, if they are configured for use and if they are accessible in LDAP by ExtraView.

ExtraView Field	LDAP Field	Mapped	Comments
User ID	LDAP_PRIMARYKEY	Yes	Used for login and user authentication
First name	LDAP_GIVENNAME	Yes	
Last name	LDAP_SURNAME	Yes	

Password	LDAP_PSWRD	Yes	Used for login and user authentication. Note that the password is used only when inserting a new user via LDAP connection, and only if the CUSTOM_AUTHENTICATION behavior setting is set to NO.
Expire password	NA	No	
User Roles	NA	Optional	May use rule mapping to map this field
Privacy Groups	NA	Optional	May use rule mapping to map this field
Set Default Area	NA	Optional	May use rule mapping to map this field
Set Default Project	NA	Optional	May use rule mapping to map this field
Email address	LDAP_EMAIL	Yes	
Date Format	NA	No	
Time in 24 Hour Format	NA	No	
Drilldown Report format	NA	No	
Time zone	NA	Yes	
Notify on own updates	NA	No	
Job title	NA	Yes	
Company name	LDAP_COMPANYNAME	Yes	This is not used in the mapping
Address	LDAP_STREET	Yes	
City	LDAP_CITY	Yes	
State / Province	LDAP_STATE	Yes	
Zip / Postal Code	LDAP_POSTALCODE	Yes	
Country	LDAP_COUNTRY	Yes	
Work phone	LDAP_PHONE	Yes	
Home phone	NA	Yes	
Cell phone	LDAP_MOBILE	Yes	
Fax	LDAP_FAX	Yes	
Pager	LDAP_PAGER	Yes	

Any UDF field	A distinguished Name	Yes	
User Expired	NA	Yes	Value is used but not saved locally within ExtraView
Session expiration limit	NA	No	

It is supported to map the same LDAP schema name to multiple ExtraView configuration properties, for example:

LDAP_PRIMARYKEY = uid LDAP_EMAIL = uid

Note: The fields in bold in the above table are typically mandatory in ExtraView.

Note: Most of the ExtraView fields are accessible through the user administration screens.

Note: ExtraView user role information is not stored in the LDAP server.

LDAP and SSO Example

The following is an excerpt from a Configuration.properties file showing the entries to connect to both an LDAP and SSO server.

```
... ... ... ... ... ...
#########################
## SSO HEADER MAPPING ##
#########################
SSO_PRIMARYKEY = SM_USER
SSO_SURNAME = SSO_SURNAME
SSO_GIVENNAME = SSO_GIVENNAME
SSO_EMAIL = SSO_MAIL
SSO_STREET = SSO_STREET
SSO_CITY = SSO_CITY
SSO_STATE = SSO_STATE
SSO_POSTALCODE = SSO_POSTALCODE
SSO_COUNTRY = SSO_COUNTRY
SSO_PHONE = SSO_TELEPHONENUMBER
SSO_MOBILE = SSO_MOBILE
SSO_PAGER = SSO_PAGER
LDAP_FIELDS =
LDAP_PRIMARYKEY,LDAP_SURNAME,LDAP_GIVENNAME,LDAP_COMMONNAME,LDAP_EMAIL,LDAP_STREET,
LDAP_CITY,LDAP_STATE,LDAP_POSTALCODE,LDAP_COUNTRY,LDAP_PHONE,LDAP_MOBILE,LDAP_PAGER,
LDAP_COMPANYNAME,LDAP_DEPARTMENT,LDAP_TITLE
#########################
## LDAP SEARCH MAPPING ##
#########################
LDAP_PRIMARYKEY = employeenumber
LDAP_SURNAME = sn
LDAP_GIVENNAME = givenname
LDAP_COMMONNAME = cn
LDAP_EMAIL = mail
LDAP_STREET = street
LDAP_CITY = l
LDAP_STATE = st
LDAP_POSTALCODE = postalcode
LDAP_COUNTRY = postaladdress
LDAP_PHONE = telephonenumber
LDAP_MOBILE = mobile
LDAP_PAGER = pager
LDAP_COMPANYNAME = displayname
LDAP_DEPARTMENT = department
LDAP_TITLE = title
```

```
EV_LDAP_PRIMARYKEY = USR_ID
EV_LDAP_SURNAME = USR_LNAME
EV_LDAP_GIVENNAME = USR_FNAME
EV_LDAP_COMMONNAME = USR_NAME
EV_LDAP_EMAIL = USR_EMAIL
EV_LDAP_STREET =
EV_LDAP_CITY = USR_CITY
EV_LDAP_STATE = USR_STATE
EV_LDAP_POSTALCODE =
EV_LDAP_COUNTRY =
EV_LDAP_PHONE = USR_PHONE,USR_PHONE2
EV_LDAP_MOBILE = USR_MOBILE
EV_LDAP_PAGER =
EV_LDAP_COMPANYNAME =
EV_LDAP_DEPARTMENT = USR_DEPT
EV_LDAP_TITLE =
... ... ...
... ... ...
```

Updating ExtraView User Information

When the LDAP_UPSERT = YES functionality is set in Configuration.properties, a user's information will be updated in ExtraView with the latest information found in the LDAP directory. This occurs when the user signs into ExtraView, and without any action on the user's part.

With this feature, the ExtraView administrator needs only to create the account for a new user, using the SAME unique id as the LDAP_PRIMARYKEY in the LDAP directory. Then upon sign on, these users have their account information in ExtraView updated. All fields that are mapped are updated, and kept up to date automatically.

Note: This feature should only be used if you have licensed ExtraView with concurrent license usage, as users can continually add new licenses to the user table.

ExtraView Help

Navigation Bar Help

The ExtraView application includes a comprehensive HTML-based help system that you can access at any time by clicking the **Help** button on the navigator frame. In addition, many tool tips and context-sensitive links are defined throughout the application.

When you place the mouse cursor over a screen label that has a tool tip, a small window will appear next to your mouse cursor with a definition of what this label does. These labels allow you to define help tips for your users. If you press the mouse button over a label, you will be taken to a specific page within the help system. If you do not have a specific page defined within the screen name Administrative section, you will be taken to the Help Index page. This page consists of links to detailed information about the system.

ExtraView End-User Help Index

Version 10.1

This is the end-user Help Index for the ExtraView issue tracking and workflow management system. Use these links to navigate to topics in the help system. Topics may also be accessed by clicking on field name links as found throughout ExtraView's various interface screens.

Note that while the features and functions described here are generic, the "look and feel" of the screens may not match your installation's specific configuration. Please contact your System Administrator with questions.

For complete end-user and administrative documentation on ExtraView, please visit the ExtraView Documentation center.

Help Topic List

- Before You Get Started
- User Sign On
- User Interfaces
  - Standard Interface
    - Home Page and Navigation
    - Changing your Personal Options
  - Workspace Interface
    - Workspace Menu
    - Panels
- Adding a New Issue
- Editing an Existing Issue
- Queries
  - Expanded Queries
  - Advanced Queries
- Reports
  - Column Reports
  - Summary Reports
  - Matrix Reports
  - Aging Reports
  - Calendar Reports
  - Taskboard Reports
  - Planning Reports
  - Charts
  - Dashboard Reports
  - Custom URL Reports
  - Container Reports / Layouts
- Email Notification Features
- Field-Type Behaviors
- Glossary
- ExtraView Corporation, the Author of ExtraView

Copyright © ExtraView Corporation, 1999-2016. All rights reserved.

ExtraView Help screen

Defining Your Own Help System

It is straightforward to develop your own help system within ExtraView. This can be completely separate from ExtraView's inbuilt system, or it can completely replace the inbuilt system.

Default Help System Paths

The standard Help system HTML files are stored in a directory with the path of en_US/help, within the installation directory where your ExtraView installation resides. The images embedded within the Help system are stored by default in the path en_US/images. Note that the menu buttons for the Help system are stored in the standard images directory for the system.

Defining the Path to your own Help System

If you develop a complete help system, it is recommended that you set the behavior setting named HELP_HOME in the Installation Defaults menu within the ExtraView Configuration menu within Administration.

You can then set a regular index.html file within this directory, and set up a complete system from there.

When you press the **Help** button on the main menu, this page is accessed, in a new window.

Accessing your own Help System

You can obviously build the help file with internal links to navigate around the pages. You can also use the Help URL within each field of the Data Dictionary to provide context sensitive help on each field that appears on each screen such as the *Add Issue* and *Edit Issue* screens. You can also use the standard html bookmark convention to provide a drilldown from the Data Dictionary to a place within a help page.

Field-Level Help

The behavior setting named ALLOW_HELP controls the link that is placed on field labels on the *add* and *edit* screens. When this is set to YES, then when a user clicks on the field label, they are taken to the help system and to the page specified in the **Help URL** of the field in the data dictionary. If there is no value in the **Help URL**, the user is taken to the index page of the help system.

Menubar Help

All menubars may have a context sensitive help button which takes the user to the appropriate documentation page within this site.e **Help URL**, then the user is taken to the index page of the help system.
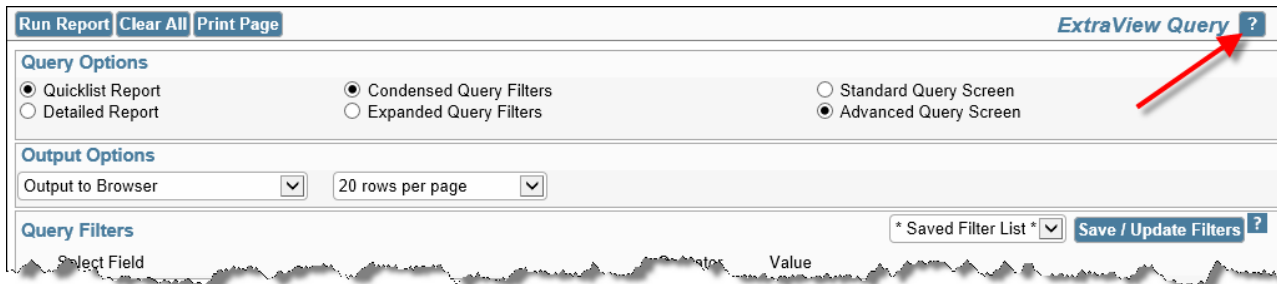
Whilst this button always appears within Administration screens, the behavior setting named ALLOW_HELP can be set to NO to remove the button from end-user accessible screens.

Adobe PDF Report Templates & Mapping

You may configure one or more buttons on an *edit* screen to merge data with an Adobe PDF document, using any field data from the current issue (and related issues) that are displayed on the *edit* screen. The Adobe PDF document is created outside ExtraView using tools from Adobe. It is possible to further customize the contents of PDF document using custom code techniques - see the User Custom Guide. You will require access to, and a good working knowledge of Adobe Acrobat and/or Adobe LiveCycle Designer to create PDF forms that can be utilized with ExtraView.

The setup involves three basic steps:

- Create a template by uploading a PDF file, and mapping fields within the PDF file to fields within ExtraView. Note that fields that have the same name within ExtraView and within the PDF document are mapped automatically. Also note, from the example at the bottom of the page that Adobe forms use a particularly verbose means of identifying a field. The Adobe tools show the definitions of the fields which include their position within the document. The appearance of a typical field is something like:

  ```
  topmostSubform[0].BodyPage[0].Page1[0].BIRTH[0]
  ```

  Consult the Adobe documentation for more information.

- The PDF document file uses forms with fields, in a similar manner to which ExtraView uses layouts with fields

- Create and configure a button field to trigger the creation of the merged PDF document

- Place the button field on an *edit* screen layout.

The resulting Adobe PDF document is added into the ExtraView issue as an attachment or into the contents of a document type field.

Creating the Template

The template is created with the administration utility named **Adobe PDF Report Mapping**, found on the **Operational Tasks** administration menu. Within the utility, press the **Add** button to begin the process.

Creating the PDF template

Simply provide the title to the template, and browse to and select your PDF document. Then click on the **Create Template & Upload File** button. You are then taken to the following screen:

Mapping fields with the PDF template

Options

- **Save Mappings / Upload New File** - This allows you to select a new file from your local computer and replace it the PDF stored on the server. Use the **Browse** button to select the file then press this button
- **Delete Template** - Delete the template and all related PDF files
- **View Original PDF File** - This allows you to view the original PDF that was uploaded. You may save this file for any purpose
- **Preview PDF Document** - Preview the mapped document.

General Points about Entering Properties

- You use the Properties box to provide the mappings
- All comments are on lines that being with a # character. These lines are ignored when processing the mappings
- Fields that have the same name within ExtraView as within the PDF document are mapped automatically for you. There is no need to include these in the mappings, unless you want to map them to different fields
- Use a \ character to escape a space character which follows
- The two rows on the screen labelled **PDF Fields & Values** and **ExtraView Fields & Values** are "helper" fields. They display the fields and their values that can be mapped. A button allows you to copy these fields and values to the computer clipboard so that you can paste them into the Properties. Of course, you can simply type the fields and values as appropriate into the Properties
- For multi-valued list fields, repeating row fields, and related record fields mapped to PDF text fields, the multiple values are concatenated with commas
- Multi-valued list fields can be mapped to multiple PDF checkbox or radio fields
- The properties file is checked for syntax & content errors when the **Save Mappings / Upload New File** button is clicked. Errors are listed on the screen, and the mapping template is not saved to the database when there are any errors.

Property Mapping

There are several key sections to the properties. You will see some examples as comments within the properties.

- **Field name mapping**
  This is the primary method to map fields from ExtraView into a PDF document. Fields that have the same names in ExtraView and in the PDF document are mapped automatically. The basic construct of a mapping entry is:

  ```
  ev_field|<ExtraView_field> = pdf_field|<PDF_field>
  ```

  For example, if you want to map the ExtraView named **SHORT_DESCR** to a PDF field named **TITLE** you would use: ev_field|SHORT_DESCR = pdf_field|TITLE If you want to map a single ExtraView field to multiple fields within a PDF document, use a the label parameter as shown here:

  ev_field|<ExtraView_field>|<label_1> = pdf_field|<PDF_field_1> ev_field|<ExtraView_field>|<label_2> = pdf_field|<PDF_field_2>

  The `<label_n>` is any string that makes the left-hand side of the expression unique. For example, if you have the **SHORT_DESCR** field in ExtraView and you want to use that as a heading on a PDF with two pages with a heading field of **hdr1** on page 1 and **hdr2** on page 2, you would use:

  ev_field|SHORT_DESCR|1 = pdf_field|hdr1 ev_field|SHORT_DESCR|2 = pdf_field|hdr2

  In addition to ExtraView's fields, you may also map the following:

  - SYSDATE - The current date and time
  - SYSDAY - The current date
  - USER - The current user preparing the report
- **Field value mapping**
  This section maps the values of fields in ExtraView to the values of fields in the PDF document. Once again, if the value of the field is identical in both ExtraView and the PDF document, then there is no need to provide a mapping. Further, mappings are typically provided only for enumerated list type fields, with display types such as List, Popup, Checkbox, Radio Button, Tab, etc. Value mappings for text, date and numeric type fields are not required. The syntax to map a value is as follows:

  ev_value|<ExtraView_field>|<value> = pdf_value|<PDF_field>|<value>

  For example, we might want to map the value of **Not Fixed** as a value in the ExtraView field named **STATUS** to a checkbox field in a PDF document, where the checkbox field is named **Not finished_checkbox**:

  ev_value|STATUS|Not\ Fixed = pdf_field|Not\ Finished_checkbox|On

  Note the use of the \ character to escape the space character. It is possible to map the one ExtraView field value to multiple fields within the PDF form. The syntax is as follows:

ev_value|<ExtraView_field>|<value>|<label> = pdf_value|<PDF_field>|<value>

- **Filtered field name mapping**
  Defining filtered field mapping entries is similar to field name mapping entries. They provide a mechanism to use repeating rows and related issues for a specific issue that you are using to prepare a PDF document. The filter itself is defined in the section **Filter definition for repeating rows and/or related records**. You must define an entry in that section for every filter you use in this section. The syntax for a filtered field mapping is as follows:

  filter_ev_field|<Filter>|<ExtraView_field> = pdf_field|<PDF_field>

  Use either of the above two forms, depending on whether you are mapping to a single or to multiple fields in the PDF document. For example, to apply a filter named **myProjectFilter** which refers to a field named **ADDRESS** within a related issue you might have this entry for the mapping:

  filter_ev_field|myProjectFilter|ADDRESS = pdf_field|Address

- **Filtered field value mapping**
  This provides a means to map ExtraView field values where the field is on a repeating row or a related issue. The filter itself is defined in the section **Filter definition for repeating rows and/or related records**. You must define an entry in that section for every filter you use in this section. The syntax for a filtered field value mapping is as follows:

  filter_ev_value|<Filter>|<ExtraView_field>|<value> = pdf_value|<PDF_field>|<value>

  For example, we might want to map the value of **Not Fixed** as a value in the ExtraView field named **STATUS** of a related issue to a checkbox field in a PDF document, where the checkbox field is named **Not finished_checkbox**. We want to use the relationship defined by the filter named **myProjectFilter**:

  ev_value|myProjectFilter|STATUS|Not\ Fixed = pdf_field|Not\ Finished_checkbox|On

  It is possible to map the one ExtraView filtered field value to multiple fields within the PDF form. The syntax is as follows:

  filter_ev_value|<Filter>|<ExtraView_field>|<value>|<label> = pdf_value|<PDF_field>|<value>

- **Default date format**
  This section defines a default date format to be used to place all ExtraView *date* and *day* display type fields on the PDF document. The format used is known as the Java Simple Date Format. You can see a definition of the date formatting here. An example of using the default date format is:

  DEFAULT_DATE_FORMAT = MM/dd/yyyy

- **Date format for specific fields**
  This section defines a date format to be use for a specific PDF document field that contains a date. The format used is known as the Java Simple Date Format. You can see a definition of the date formatting here. An example of using the default date format is:

  DEFAULT_DATE_FORMAT = MM/dd/yyyy

- **Document, image fields and file attachment mapping**
  You are able to map document field, image field display types as well as issue attachments from ExtraView issues to PDF documents. The syntax to map fields from the current record is:

  ev_doc_img|<ExtraView_field>|<label> = pdf_page|<page_number> ev_attachments|<label> = pdf_page|

  The **pdf_page|<page_number>** parameters indicate where in the PDF document that the document or image will be inserted. As well as mapping these fields from the current issue, you can map them from repeating records and related issues, using the filtered field mechanism. The syntax is:

  filter_ev_doc_img|<Filter>|<ExtraView_field>|<label> = pdf_page|<page_number> filter_ev_attachments|<Filter>|<label> = pdf_page|

An example might be to map the same image in an ExtraView image field named **LOGO_IMAGE** at the beginning and end of a document:

ev_doc_img|LOGO_IMAGE|start= pdf_page|0 ev_doc_img|LOGO_IMAGE|finish = pdf_page|10

- **Filter definition for repeating rows and/or related records**
  The filter expressions that you can build are similar to the links you build with the **LINK** directive in [Business Rules](). Filter expressions are additive as they are defined, and separated by commas. The basic syntax is:

  FILTER|<Filter_name> = <expression>

  This example shows how two filters can be used:

  FILTER|personFilter = AREA = Legal, PROJECT = Personnel, PERSON = USER

  You may use the special variables **$$SYSDATE$$**, **$$SYSDAY$$**, **$$USER$$**, and **$$ID$$** to match the current date, current day, current user, and the ID of the current issue respectively. Similar to the **LINK** business rule directive, you use the **RG_NAME** and **RG_TYPE** attributes to define filters for related issues. The syntax is:

  FILTER|<Filter_name>|RG_NAME = <relationship group name> FILTER|<Filter_name>|RG_TYPE = <relationship type>

  The **RG_NAME** is the name of an existing relationship group. The **RG_TYPE** may be one of **CHILDREN**, **GRANDCHILDREN**, **GRANDPARENTS**, **PARENTS**, **RELATED**, **MEMBERS**, **LINKED**, or **SIBLINGS**. If you do not specify **RG_NAME** or **RG_TYPE** for a filter, the default relationship group is used, with an **RG_TYPE** of **LINKED**. The filter can be used to define how to display the results, when many records are returned by the filter. Typically, you will want to display the results horizontally or vertically. The syntax for the filter is:

  FILTER|<Filter_name>|DELIMITER = delim

  where delim is the string used to delimit the records. The default delimiter is a comma. If you want to display the results vertically you will use the delimiter \n to introduce a newline character between each result. The typical entries you will define for related issues that are to be mapped to a PDF document are:

  FILTER|<Filter_name> = <expression> FILTER|<Filter_name>|RG_NAME = <relationship group name> FILTER|<Filter_name>|RG_TYPE = <relationship type>

  An example of a filter definition for use within related issue mappings might be:

  FILTER|myProjectFilter = AREA = Customers, PROJECT = Customer Data FILTER|myProjectFilter|RG_NAME = CUSTOMER_ISSUES FILTER|myProjectFilter|RG_TYPE = CHILDREN

  When handling repeating rows, you can use the **ROW_NUM** attribute to match a specific row. The syntax is:

  FILTER|<Filter_name>|ROW_NUM = <integer>

Creating the Button Field

The button field is created within the data dictionary as a User Defined Field with the following properties:

Property	Purpose
Fixed Name	The name you use to create the field must begin with **GENPDF_** and end with **_BTN**. For example, a valid name is GENPDF_SALESREPORT_BTN
Title	You normally will not want to see a title with the button field you are generating, therefore use a space character for the title
Display Type	This must be **Custom**

Default Value	This is the name of the template you created in the first step. It is important to use the exact name you created
Help Text	This is text that will be placed on the button and will be visible to the user. For example, you might put **Generate PDF** as the text
Help URL	This entry should either be the word **ATTACHMENT** to store the generated PDF document as an attachment, or the data dictionary name of a document field to store the generated PDF document in a document field.

Configuring the Layout

Simply place the field you created in the previous step on the edit screen layout where you want to see the button. You may apply the usual layout cell attributes such as FIELD VISIBLE IF to the field on the layout.

Sample Configuration

```
ev_field|MANUFACTURER_REPORT_NUMBER = pdf_field|topmostSubform[0].BodyPage[0].Page1[0].Mfr_Report[0]
ev_field|UF_IMPORTER_REPORT_NUM = pdf_field|topmostSubform[0].BodyPage[0].Page1[0].UF_Importer_Report[0]
#
# Section A
#
ev_field|PATIENT_IDENTIFIER = pdf_field|topmostSubform[0].BodyPage[0].Page1[0].Patient_Identifier[0]
ev_field|PATIENT_AGE = pdf_field|topmostSubform[0].BodyPage[0].Page1[0].AGE[0]
ev_field|PATIENT_DOB = pdf_field|topmostSubform[0].BodyPage[0].Page1[0].BIRTH[0]
ev_field|PATIENT_WEIGHT = pdf_field|topmostSubform[0].BodyPage[0].Page1[0].Weight_lbs[0]
ev_field|PATIENT_GENDER|1 = pdf_field|topmostSubform[0].BodyPage[0].Page1[0].Male[0]
ev_field|PATIENT_GENDER|2 = pdf_field|topmostSubform[0].BodyPage[0].Page1[0].Female[0]
#
# Section B
#
ev_field|ADVERSE = pdf_field|topmostSubform[0].BodyPage[0].Page1[0].ADVERSE[0]
ev_field|PRODUCT_PROBLEM = pdf_field|topmostSubform[0].BodyPage[0].Page1[0].PRODUCT[0]
ev_field|OUTCOME|1 = pdf_field|topmostSubform[0].BodyPage[0].Page1[0].Death[0]
ev_field|OUTCOME|2 = pdf_field|topmostSubform[0].BodyPage[0].Page1[0].DISABILITY[0]
ev_field|OUTCOME|3 = pdf_field|topmostSubform[0].BodyPage[0].Page1[0].LIFE_THREATENING[0]
ev_field|OUTCOME|4 = pdf_field|topmostSubform[0].BodyPage[0].Page1[0].CONGENITAL[0]
ev_field|OUTCOME|5 = pdf_field|topmostSubform[0].BodyPage[0].Page1[0].HOSPITAL[0]
ev_field|OUTCOME|6 = pdf_field|topmostSubform[0].BodyPage[0].Page1[0].Other_Event[0]
ev_field|OUTCOME|7 = pdf_field|topmostSubform[0].BodyPage[0].Page1[0].REQUIRED[0]
ev_field|PATIENT_DOD = pdf_field|topmostSubform[0].BodyPage[0].Page1[0].Death_Date[0]
ev_field|EVENT_DATE = pdf_field|topmostSubform[0].BodyPage[0].Page1[0].EVENT[0]
ev_field|DATE_REPORT = pdf_field|topmostSubform[0].BodyPage[0].Page1[0].REPORT[0]
ev_field|DESCRIPTION = pdf_field|topmostSubform[0].BodyPage[0].Page1[0].DESCRIBE[0]
ev_field|RELEVANT_TESTS = pdf_field|topmostSubform[0].BodyPage[0].Page1[0].RELEVANT[0]
ev_field|OTHER_RELEVANT_HISTORY = pdf_field|topmostSubform[0].BodyPage[0].Page1[0].OTHER_RELEVANT[0]
ev_field|OTHER_REMARKS = pdf_field|topmostSubform[0].BodyPage[0].Page3Cont[0].concomitant[0]
#
```

Creating Alternative Navigation Bars

ExtraView comes with a built-in navigation bar which is used for both the standard and workspace interfaces. The theme of the navigation bar is selectable, allowing you to change the graphical style and color of the buttons. Although you may change the theme of the built-in navigation bar, you may not change its functionality. Using the utility described on this page, you may create additional navigation bars for any or all user roles that the administrator has defined and use these to override the built-in navigation bar. The built-in navigation bar appears like the following screen shots, when you have selected the pale blue theme:
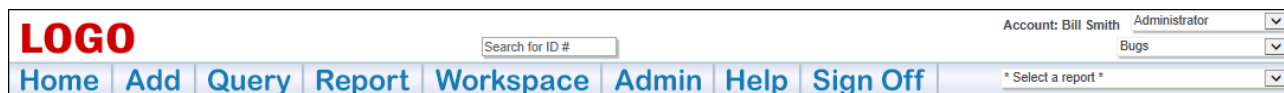
Built-in Standard Navigation Bar

Built-in Workspace Navigation Bar

New navigation bars are able to be created for use by one or more user roles in your system. You may define navigation bars that have significantly more entries than the built-in navigation bar, with purpose-built buttons that control access to most ExtraView functions and with access to web pages outside of Extraview. You may also define hierarchical-style navigation bars with sub-menus being available in a traditional menu-style form.

Access to the administration utility is provided with the security permission key named CF_NAV_BAR. Note that the ADMIN user account always uses the inbuilt navigation bar, and never uses a navigation bar that you define. This prevents a situation where you design a navigation bar that inhibits access to the administrative functions, and effectively locks all users out of administration.

The administration utility that allows the definition and modification of different navigation bars is accessed through **Admin --> System Configuration --> Navigation Bar**. This initially takes you to a screen like this:


Navigation Bar Administration

The key attributes about designing your own navigation bar are as follows:

- You may define up to seven sub-menu levels for any navigation bar button, but for usability it is suggested that you do not go to more than three levels deep.
- Any button on the navigation bar may access a function when it is clicked on by the user, or it may provide a sub-menu when it is clicked. If a sub-menu is selected, the user can click on any entry in that sub-menu to execute a function, or to open up another sub-menu at a lower level.

Customized Navigation Bar with Sub-Menu

- Use the **NAVIGATION BAR BUTTON/MENU ITEM PROPERTIES** tab to define the properties for each navigation button, or each drop-down menu item placed within the individual navigation bar button
- Two blank images are provided within the directory structure of the existing buttons, named ButtonTemplateOff.png and ButtonTemplateOn.png. These may be used as placeholders while you are designing your own buttons for the top level of the navigation bar. They are used as the default names for the buttons you create, but you can change these defaults to any name you use for buttons you create.
- The original Adobe Photoshop files used to create the buttons are provided within the installed directory structure at *extraview_installed_directory*/locales/en_US/images_nav_bar. You can use Photoshop with the file for your theme, to create matching buttons for each of the themes provided. You can also create your own navigation bar theme from scratch (see here). Note that the navigation bar buttons for all workspace themes are contained in a single file, named nav_bar_workspace.psd.
- Any button may be designed with any text or graphic on the button itself. The button also has a title which is used when the user places their mouse over the button.
- Any button you create may be made visible on the standard interface alone, the workspace interface alone, or on both interfaces. You can create new navigation buttons using the appropriate icon on the user interface:
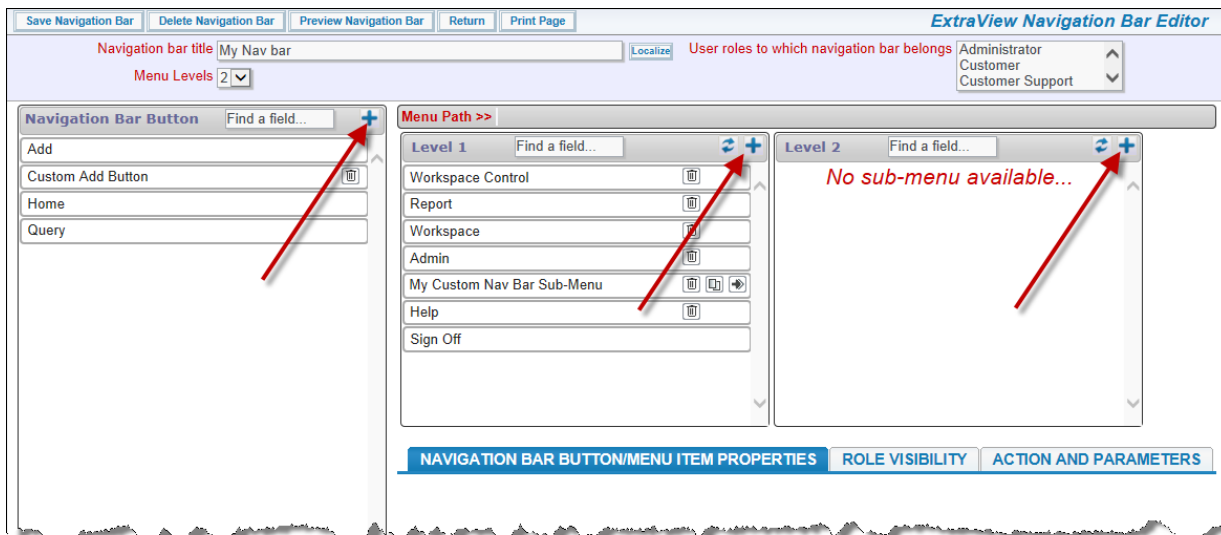
Adding a New Navigation Bar Button
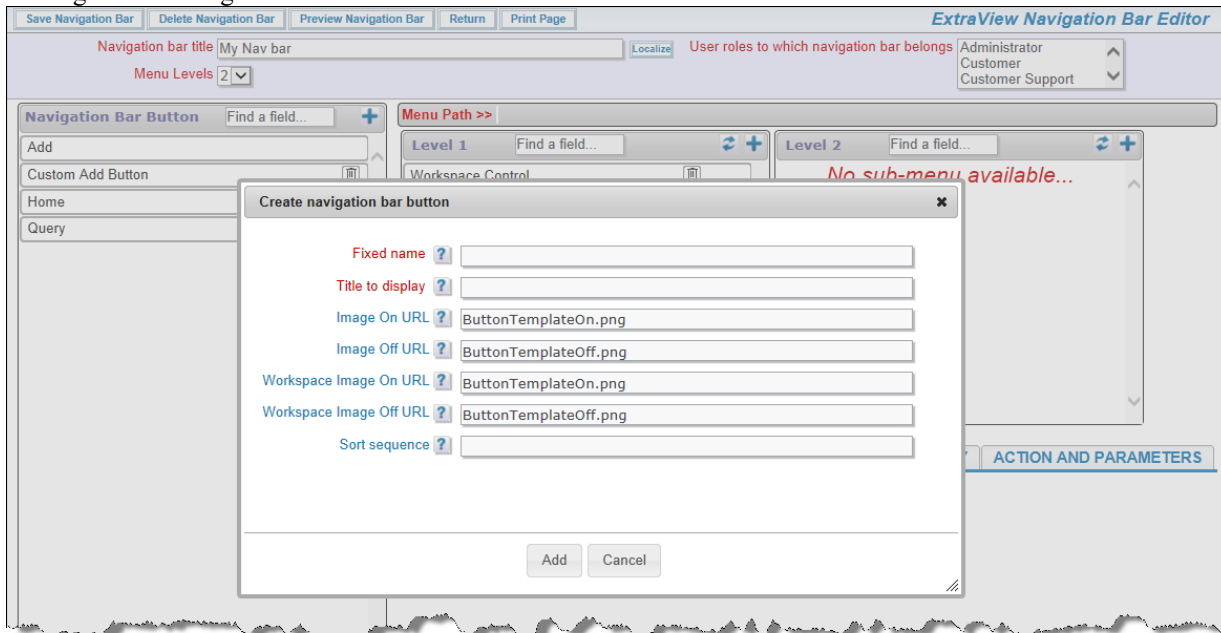
Adding a New Navigation Bar Button

- There are four possible images for each navigation bar button. Two are used on the standard interface, two on the workspace interface. Only two images are needed if you are only using the button on one interface. The buttons are of the "rollover" type, with one image being used when the button is visible, but no mouse over it, and one image being used when the mouse is over the button
- The navigation bar buttons and sub-menu items may be made visible or invisible for any of the roles to which a navigation bar belongs. You use the **ROLE VISIBILITY** tab for this
- Each navigation bar button or sub-menu item has an action type that defines what happens when you click on the button or sub-menu item. The available actions on the **ACTION AND PARAMETERS** tab are:
  - **Sub-Menu**: This entry creates a sub-menu where additional actions are defined. Sub-menus may point to further sub-menus, nesting to a maximum of seven levels
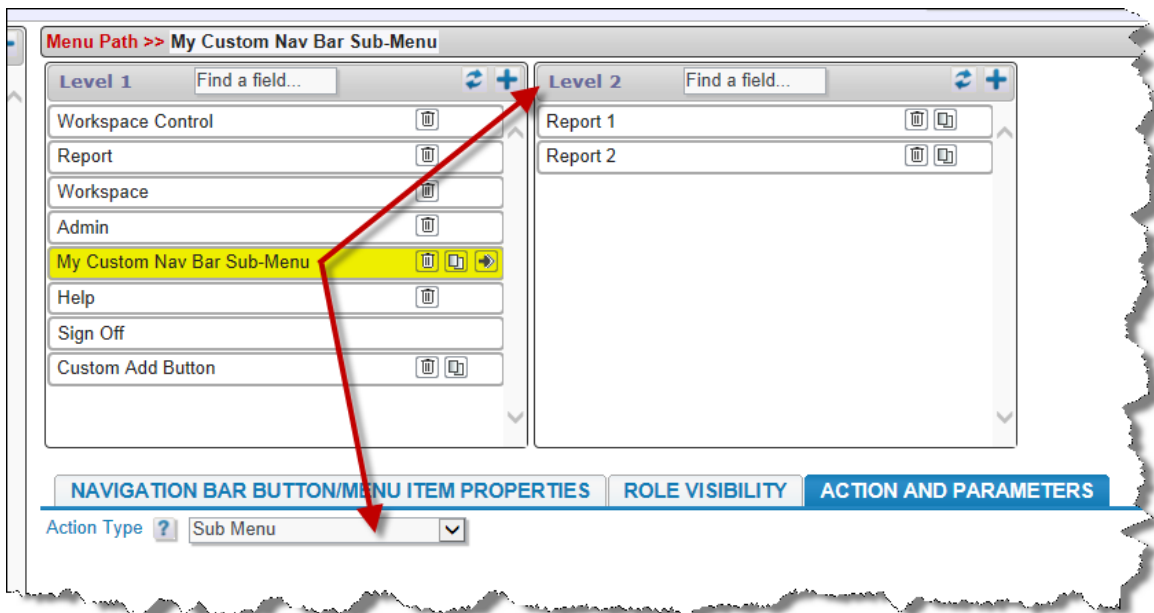
Custom Sub-Menu

- ○ **Open URL**: This action allows you to enter any valid URL. A new browser window or tab with the contents of the URL will be opened when the user selects a menu entry with this action
- ○ **Add Issue**: This action opens an *Add Issue* screen or panel within the ExtraView window or workspace. In addition to opening the *Add Issue* screen, you must select a Business Area and Project for the screen. You also have the ability to choose any number of field names and initial values which will be used to populate the *Add Issue* screen. As you start to type a name, the utility will pop up a small window with matches for the name you are entering. Typically, you will provide a field name and a value, and these will be used to populate the *Add Issue* screen when the user presses the button
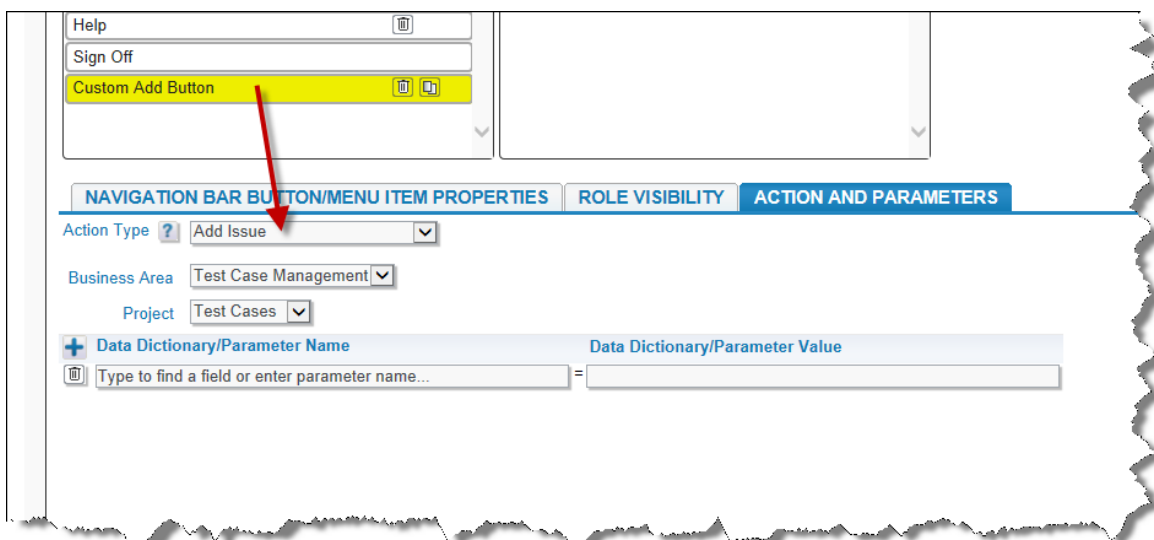
Custom Add Button

- ○ Not all field display types are supported for use with fields that can be used to populate values onto the *Add* screen. Valid field display types are: Checkbox, Currency, Date, Day, Decimal, List, Number, Pop-up, Radio button, Tab, Text Area, Text Field, and User. There are also restrictions in that the following fields may not be used to populate the screen: ID, PARENT_ID, AREA, PROJECT, ITEM_ID, EV_REPORT_SELECTION, RELATIONSHIP_GROUP_TXNS, NOTIFICATION_HISTORY, ALT_ID, all ATTACHMENT fields, ITEM_ID, MINI_HISTORY, HIST_RANGE_START, and HIST_RANGE_END. In addition to using the fields in your database as actions, you may also use the parameters that are passed to the screen form. In practice, the parameter that may be of most use is to define the name of **add_confirm_page** with a value of no. This will inhibit the screen from moving to the **Add Confirmation** screen after adding the issue
- ○ **Run Report**: You can select a public report for the menu, to run when the menu entry is chosen. Note that only public reports are available for placement on a navigation bar. In addition to the report, you can select the output type and the page size

- **Custom Action**: A custom action allows the definition of a user custom JavaScript method to call when the button is pressed by the user. As you would expect, this action must be supported by JavaScript code in the UserCustomJavaScript.js file
- **ExtraView Home**: This action purposes the button to open the standard interface Home Page screen when the button is pressed by the user. This action only operates in the standard interface. It has no effect within a workspace interface
- **ExtraView Add**: This action opens an *Add Issue* screen or panel. Unlike the **Add Issue** action, the screen is opened to the user's current Business Area and Project. This provides the functionality of the standard, in-built *Add* button
- **ExtraView Query**: This action opens the standard, in-built ExtraView Query screen
- **ExtraView Report**: This action opens the standard, in-built ExtraView Report screen
- **Open in Edit**. ExtraView supports a mode where new issues are created and inserted into the database before presenting the input screen to the user. The issue thus has an ID (and perhaps an ALT_ID) before the user sees the screen. This action allows a new issue to be created and for the screen to be opened to the user in edit mode
- **ExtraView Workspace**: This action opens the default standard, in-built Workspace for the user. This action is only operable on the standard interface
- **ExtraView Admin**: This action places the button that accesses the standard, in-built Administration screens on the navigation bar
- **ExtraView Help**: This action places the button that accesses the standard, in-built Help screen on the navigation bar
- **ExtraView Sign Off**: This action places the button that operates the standard, in-built Sign Off button on the navigation bar. You should always place this button on the top-level menu
- **ExtraView Workspace Menu**: This action is only operable within a workspace. It places the standard, in-built Workspace menu on the navigation bar.
- You should always place a Sign Off button on navigation bars that you design. It is also recommended that you place a Home Page button on the standard interface navigation bar and a Workspace menu button on the workspace navigation bar

Triggering Rules from a Navigation Bar

This technique only works with **Add** buttons on navigation bars. You can trigger a rule action using a business rule similar to this:

```
if (SCREEN_NAME = 'ADD' && MY_ADD_NAV_BUTTON= 'Add Task' ) {
  PRIORITY = 'P 2';   CATEGORY = 'Task';
}
```

If the user chooses the **Add Task** navigation bar item, then the PRIORITY and CATEGORY fields are set as shown above.
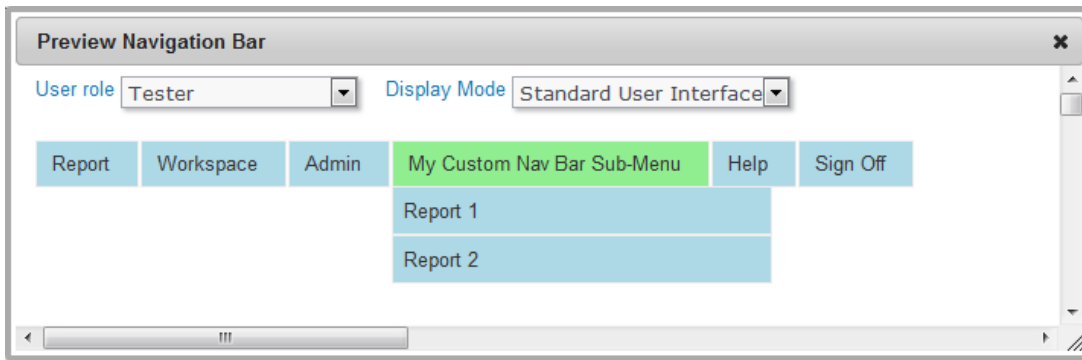
Tips to Create a Navigation Bar

- When you first create a new navigation bar, use the multi-select list with the title **User roles to which navigation bar belongs** to select all the user roles with which you want to use the navigation bar
- You have control over which buttons within the navigation bar appear for each of the selected user roles selected using the **Role Visibility** tab within each entry on the menu. Thus, you can define a navigation bar for all or many roles, with all the buttons you require, and then use the **Role Visibility** checkboxes to turn individual entries off and on for each role
- You also have control over which buttons appear within the standard user interface and within the workspace interface using the **Display mode** selector within the properties
- The built-in ExtraView navigation bar uses the MENU_XXX security permission keys to control visibility. For example, MENU_ADD_PROBLEM controls the **Add** button on the built-in navigation bar
- When you create your own navigation bar, the MENU_XXX permission keys are used for built-in buttons and permission keys named MENU.XXX are made available for your own user-defined buttons. For example, when you add built-in buttons named MY_REPORT_NAV, MY_QUERY_NAV, and MY_ADMIN_NAV to your navigation bar, you will have the following permission keys available, to control the button role visibility:
  - MENU.MY_REPORT_NAV
  - MENU.MY_REPORT_NAV
  - MENU.MY_ADMIN_NAV

  You can control the permission to your buttons either through the utility documented on this page, or via the **Grant Security Privileges** administration utility.

- If you use the **Sort sequence** to reorder a list, use the **Refresh** button to re-display the menu
- When designing a navigation bar, you can use the **Preview Navigation Bar** menu bar button to check exactly what will be seen for any user role and for both the standard user interface and the workspace interface. Here is an example:

Preview Mode

- Plan your new navigation bar from the top level downwards, and have a graphic designer use Photoshop to create the new buttons before completing the configuration of the navigation bar.

Initial Setup Menu

Uploading New License Keys

When you first install ExtraView, or you purchase additional ExtraView licenses or make other changes to your ExtraView license you import the new license with the utility titled **Upload New License Activation Key**. If you were sent your license activation key via email, save the attachment that contains the license key to your computer, remembering the location where you are saving it. When installing the product for the first time, the following screen comes up automatically, after you accept the end-user license agreement. If you are updating your license file for any other reason, navigate to **Admin**, **Initial Setup**, **Upload License Activation Key** to see the screen.
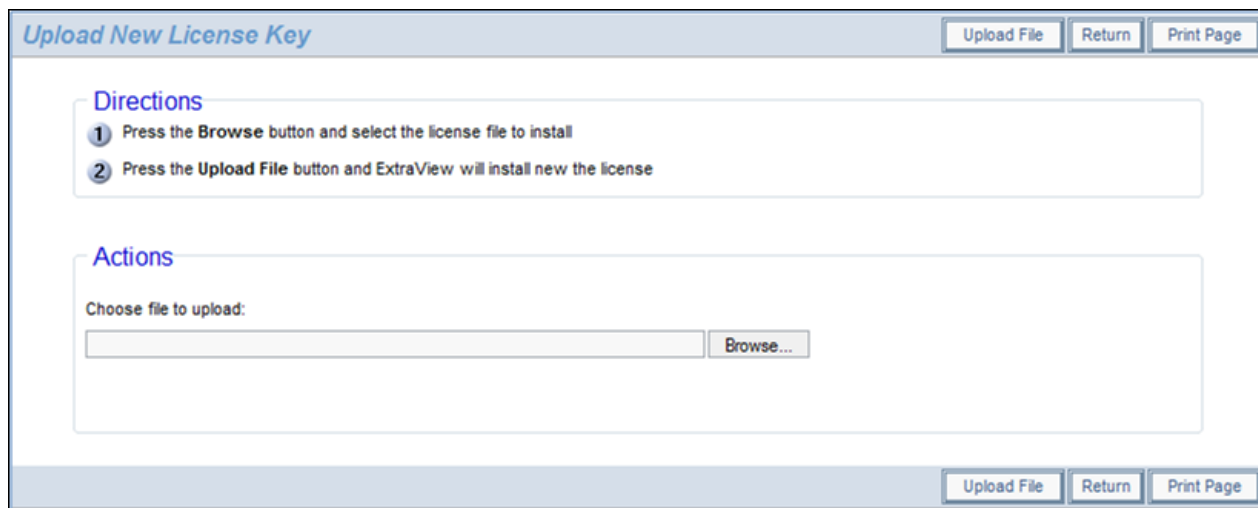
Importing a new license key

Simply use the **Browse** button, navigate to where you saved the the license activation key file provided by ExtraView and then press **Upload File**. You will see a confirmation screen indicating the parameters of the activation key you uploaded. Your new license key takes immediate effect.

Sign On Message

Many organizations use the Sign on Message screen as a bulletin board for system-wide messaging to the members of the organization. You may use all the capabilities of HTML to customize your own Sign on Message.

From the Administration screen, under the **Initial Setup Menu**, click on **Sign On Message**.

The following screen appears:
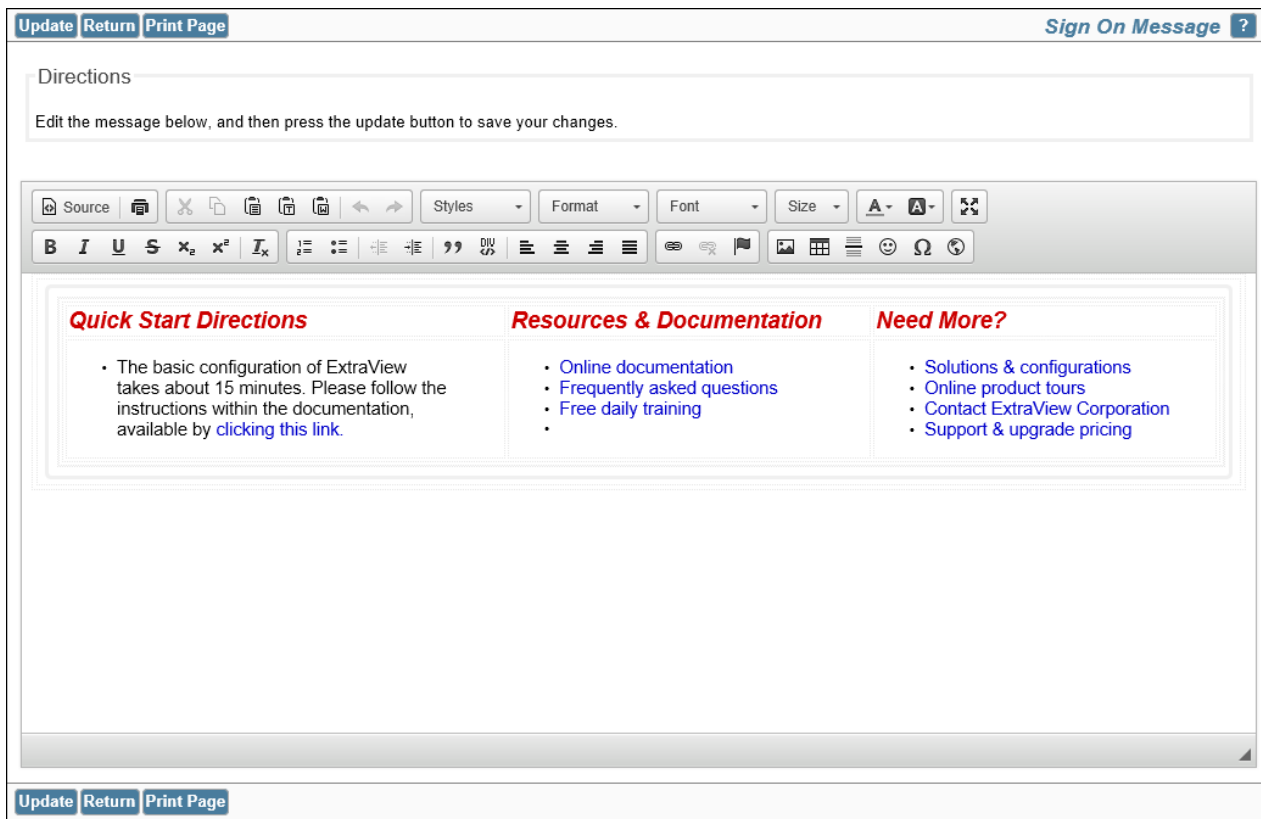
Sign On Message screen

With Internet Explorer as your browser, the HTML Area utility is used to enter and edit your sign on message. See Appendix F for a guide to using this utility to edit your message. If you are using a different browser, the message is edited as direct HTML within the edit box presented to you.

You can modify the Sign On Message in any manner, utilizing valid HTML. This screen allows you to display HTML on the home screen at the top of the page. When you are finished, press the **Update** button.

Within the sign on message, it is recommended that you use CSS styles as opposed to HTML to style the of the message to be displayed. This will ensure that styles you want for your characters will take effect. Some HTML styling of the text may not appear because of the precedence of styles used in the overall document.

As an example, you should use Here is some text as opposed to Here is some text

Below is an example of a Home screen with the Sign On Message:

Home Page screen

The sign on message is limited to a maximum length of 4,000 characters. This includes all the HTML tags and attributes. If you attempt to enter more than this number of characters, an error is generated, and you will need to reduce the size of the message. However, it is possible to include references to JavaScript functions in the UserJavaScript.js file. From there you can generate additional HTML with document.write statements.

It is the administrator's responsibility to ensure that the HTML inserted into the sign on page is valid. It is possible (for example by opening an *<A . . .>* tag without a corresponding *</A>*) to introduce a Sign On Message that will effectively lock all users out of the system. Should this happen, you must sign on to ExtraView with the *admin* user account and password. This bypasses the sign on message and allows you to edit the message and remove any poor HTML.

Uploading a Company Logo

- Add new comment

The company logo that appears in the top left hand corner of the navigation bar can be replaced with your own image. Click on the link named **Upload Your Company Logo** within the Initial Setup Menu admin screen. You will see the following:

Uploading your company logo

When you click the **Add** button, a dialog box appears, allowing you to choose a file from your local computer to upload.

If you upload a logo that is too large for the available space on the navigation bar, ExtraView will resize the logo so that it fits in the space. If your navigation bar is set to the horizontal direction, then the logo is reduced in size so that its height is equal to the MENU_SIZE.

If your navigation bar is set to the vertical direction, the logo is reduced in size so that its width is equal to the MENU_SIZE. If your application server and your web server are on different physical servers, you will need to have a mount point within the application server to ensure that the logo you upload is served up from the application server, not the web server.

Only files of type gif may be uploaded.

Note: The new company logo image may not appear immediately on your page, especially if the new one you uploaded is of identical size to the previous one. The reason for this is that the browser caches the image, and does not always detect that it should download a fresh copy. Please click on the Home button on the navigation bar to see the new logo in situ.

The same logo is placed on the sign on screen. There may be some occasions when, for aesthetic reasons, you want to place a different logo on the sign on screen. This is possible, utilizing two behavior settings, under the Display Settings menu. You will find two behavior settings there:

SIGN_ON_SCREEN_LOGO	This is the path to the logo used for the sign on screen logo. By default, this points to ../images/CompanyLogo.gif. This is the location of the company logo that is uploaded with this utility. You can point to any other valid location with an absolute or relative URL. You must place this logo on the file system of the server yourself. This allows you to place any logo image on your sign on screen.
SIGN_ON_SCREEN_LOGO_STYLE	This setting is typically used to place the logo in SIGN_ON_SCREEN_LOGO at a specific location on the sign on screen. The default is to use a style of position: absolute; left:10px;top:10px which places the logo 10 pixels from the top left-hand corner of the screen.

Note: You cannot use this feature if you are running ExtraView from within a deployed War file. Simply copy the logo file to the correct path on the server.

Note: The logo you upload will also be used in Workspaces. It will be resized automatically to fit the space in the Workspace screen.

User Interface Themes

This administrative function provides a shorthand method to set all the individual font, color and image settings with a single selection. This is the recommended method to select your user interface. Select **User Interface Themes** from the menu and you will see this screen:
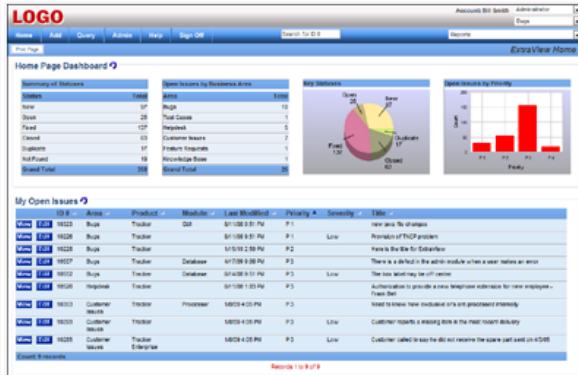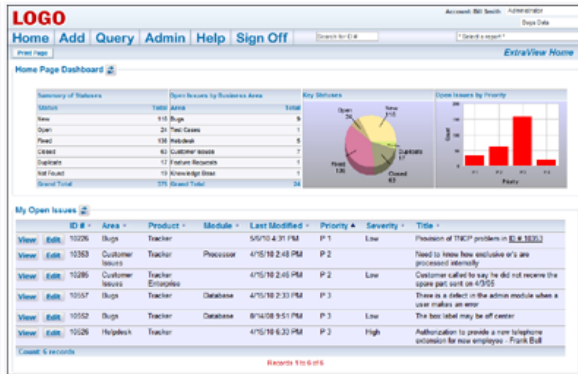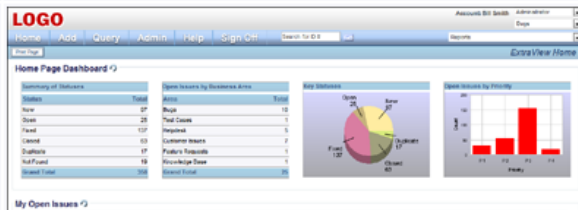
Selecting a user interface theme

Scroll down the available themes and use the checkbox to select the theme that you wish to use for your installation. Next, click on the **Update** button to make your changes. You should sign off and sign on again for all the changes to take place.

Manage Quickfind

This screen is used to set up and configure Quickfind, a mechanism that greatly increases the speed of text searches through the ExtraView database. This utility defines exactly what text will be indexed, and where the indexes will be stored. The Quickfind indexing mechanism is based upon the Apache Foundation Lucene technology.

Note that documents and attachments larger than 16MB are not indexed.

Information on the Quickfind Synchronization task is found here.

The basic setup of Quickfind is as follows:

- Check the **Task Manager** to ensure the Quickfind Synchronize Task (FULL_TEXT_SYNCHRONIZE) is **not** running. If it is running, you must stop it before proceeding. See the page here for instructions on starting and stopping the task.

- The basic setup and configuration of Quickfind is accomplished on the **Admin**, **Initial Setup**, **Manage Quickfind Settings** screen
- Enable Quickfind from the screen by setting the **Enable Quickfind** list to Yes
- Setup the path to the location where the index files will be stored on your file system. If you will use the default path as explained below, you can skip this step. All application servers in your installation must have read and write access to the index file location. The index location is also the value stored in the behavior setting named QUICKFIND_INDEX_LOCATION. Further, you must make sure you have sufficient disk space to store the indexes as they grow.
    - The default path to your indexes is quickfind_index. Note that this is using a pathname relative to your WEB-INF folder. Any pathname relative to WEB-INF may be used, except when you are running ExtraView within a WAR file environment, in which case you must use an absolute pathname
    - You may choose to enter an absolute pathname such as C:\ExtraView\QuickfindIndex within a Windows environment, or /usr/ExtraView/QuickfindIndex within a Linux environment
    - If the location is on an NFS mount, then you should include this in the index location by using the convention nfs:pathname
- The prompt **Allow Quickfind on user defined text fields** determines whether the text within all the user defined fields should be indexed. The recommendation is that you set this to *Yes*
- If you are upgrading an existing database, run the external utility named FullTextIndexSetup as described below, to create the initial indexes from the issue data and the file attachments that already exist. If this is a new installation, you can skip this step
- Finally, return to the **Task Manager** screen to add and / or start the Quickfind Synchronize Task (FULL_TEXT_SYNCHRONIZE).

The Quickfind management utility

Observe the dates on which files were last indexed, and the count of files still to be indexed. The button named **Estimate Storage Requirements** gives an approximate estimate of how much storage is required with the current database, to store the indexes. Make sure you have sufficient storage available at all times, as your database grows in size. On rare occasions, it might be necessary to manipulate the content types and the character set mappings of information to be indexed. It is not recommended that you alter these settings.

FullTextIndexSetup

If you are enabling Quickfind on an existing installation, you should index all the existing information by using the external program named FullTextIndexSetup. This should be accomplished with the task named Quickfind Synchronization Task (FULL_TEXT_SYNCHRONIZE) turned off, or the ExtraView application stopped. Some upgrades may also require the indexes to be rebuilt, using the same FullTextIndexSetup program.

The FullTextIndexSetup utility is extremely quick on a new database, but may take some time on a very large existing database. It is difficult to predict exactly how long this will take on a large database, as there are many factors such as the processing speed, memory, amount of text, and the amount of attachments that all have an impact. However, it is unlikely that the process will take more than a few hours. Your users can continue working during this period, and the search results will improve as the process continues. Our recommendation is to start the process after the majority of users leave work for the day.

The utility is found in the directory named WEB-INF/data.

The syntax to run this utility is:

For Windows platforms - FullTextIndexSetup.bat JAVA_HOME TOMCAT_HOME EV_BASE

    where JAVA_HOME is the path to your Java, TOMCAT_HOME is the path to your application server and
    EV_BASE is the path to the ExtraView installation

or for Linux platforms - FullTextIndexSetup.sh evj

    where evj is the path to the ExtraView installation
    It is likely that you will have environment variables set for JAVA_HOME and TOMCAT_HOME,
    but these can be set in the shell script file if needed.

Optional parameters for FullTextIndexSetup

-report filename - generate a report in file filename. By default, the report will be written into the ExtraView log file

-testOnly - this allows a dry run migration of the attachments without any modification to the database or the repository. Errors will be reported

-directory - the directory to use for the full text indexes if you are using Microsoft SQL Server. This must be created prior to running the script

-tablespace - the tablespace for the full text indexes, if you are using Oracle. This must be created prior to running the script.

What is Indexed for Quickfind Keyword Searches

The following are "indexed" as part of the Quickfind operation. Indexing is done in the talisk named FULL_TEXT_SYNCHRONIZE, or through the FullTextIndexSetup operation, which is run as a command-line operation (see above).

- Inbuilt Fields
  - the SHORT_DESCR field
  - If the behavior setting named QUICKFIND_INDEX_USERS is YES, the following user fields are indexed:
    - OWNER
    - ASSIGNED_TO
    - LAST_UPDATED_BY_USER
    - ORIGINATOR
    - CONTACT
    - All MODULEs: ASSIGNED_TO field
    - User Defined Field USER type fields
  - All User Defined Fields with a display type of TEXTFIELD
  - All large User Defined Fields with a display type of TEXTAREA, LOGAREA, PRINT_TEXT
  - The User Defined Fields with a display type of HTMLAREA, after removing the HTML tags
- DOCUMENT fields
  - Document description
  - File name
  - Document content according to MIME type and content size:
    - Size must be less than 16MB
    - MIME type must not be any kind of video, audio, or image
    - The MIME type is mapped to an extractor – see MIME type extractors below
- IMAGE fields
  - Image description
  - File name
- Attachments - these are only searched when the **Search Attachment** checkbox is checked
  - Attachment description
  - Attachment content according to MIME type and content size:
    - Size must be less than 16MB

- MIME type must not be any kind of video, audio, or image
- MIME type is mapped to an extractor – see MIME type extractors below

MIME Type Extractors

There are two text extractors:

1. PDF Text Extractor: uses iText PdfReader object to tokenize the PDF strings. Document content will NOT be indexed if it is marked as being "encrypted"
2. Office Text Extractor: uses one of the POI extractors appropriate to the type (Word, Excel, Powerpoint, Outlook, Publisher, or Visio). Note: if Excel text extractor fails, it tries to extract text assuming it is a comma-separated or tab-separated text file with an Excel MIME type
3. Other MIME type documents are indexed as text

Notes

As stated above, Quickfind utilizes the Apache Lucene software to provide the indexing mechanism. One support issue is that if your Java Virtual Machine, or your application server (Apache Tomcat or whatever) crashes for any reason, then the indexes may be left in a locked state on the server. The lock files are kept in the directory specified by the **org.apache.lucene.lockdir** system property if it is set, or by default in the directory specified by the **java.io.tmpdir** system property (on Unix boxes this is usually **/var/tmp** or **/tmp**). If for some reason **java.io.tmpdir** is not set, then the directory path you specified to create your index is used. Lock files have names that start with *lucene-* followed by an MD5 hash of the index directory path. If you are certain that a lock file is not in use, you can delete it manually.

For full information on the use of Quickfind, see the page titled Keyword Searching.

Configuring your own User Interface

The ExtraView user interface is composed of the following components:

- **The Navigation Bar.** This may appear horizontally at the top of the screen or may appear vertically down the left-hand edge of the screen. This always contains the main navigation buttons and a drilldown box to search for issues.
- **The Menu Bar.** This appears at the top of the screen within the main frame. If there is a horizontal navigation bar, then it appears beneath the navigation bar, else it appears at the top of the screen within the main frame.
- **Background and Text Colors.** In conjunction with the Cascading Style Sheets, these allow for the setting of the main frame colors including the text.
- **Images.** This is the range of images used as buttons within the main frame, as well as graphics used on the various screens.
- **Main Screen Image Sets.** This section shows the settings used within the preset themes. Normally you can set one of these image sets to get close to your desired interface.
- **Sign On Screen.** This explains how to add your own embellishments to the user sign on screen.
- **Cascading Style Sheets.** This provides an explanation of the different style sheets used within ExtraView.

Navigation Bar

The navigation bar is a set of horizontal buttons and links that appear at the top of the browser window. The navigation buttons that are presented may be different, according to a number of criteria:

- Buttons may be made visible / invisible by using security permission keys for user roles
- The button styles are determined by the user interface theme
- Different navigation bars may be set up for different roles within the system. With this feature, different navigation bars within the system may also have different button styles for the different roles
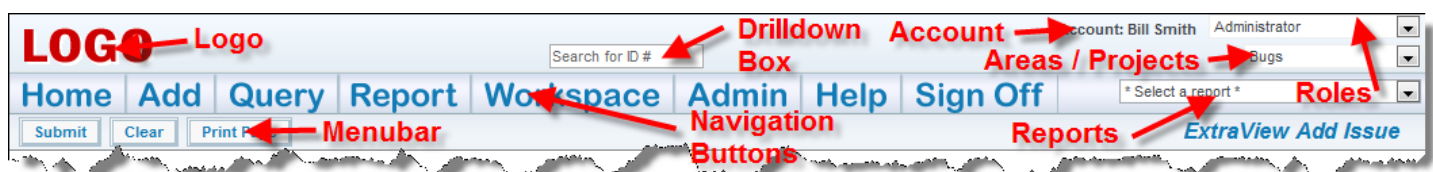
The components of the navigation bar are:

- Your company logo. The company logo image (CompanyLogo.gif) should be sized to fit correctly within your selected user interface theme
- Navigation buttons. The **Home**, **Help**, and **Sign Off** buttons are always present. The **Add**, **Query**, **Report**, **Workspace** and **Admin** buttons are present if the user's current role has permission to see and use these buttons
- The user's name with a link to their personal account information
- A reports list from which the user may execute reports to which they have permission
- A list from which the user can alter their current role
- A list from which the user can alter their current business area and / or project
- A drilldown box into which a user can enter one or more ID's and drilldown into these issues
- A contextual menubar beneath the main navigation buttons. The buttons that appear on the menubar are dependent upon the page you are visiting. The menubar also has the name of the screen. The position of the buttons and the screen name can be switched with the behavior setting named MENU_BUTTON_POSITION.

To aid the configuration of the navigation bar, there are several behavior settings in the Display category, as follows:

Behavior setting	Purpose
NAV_BAR_DRILLDOWN_BOX_STYLE	The CSS style to be applied to the table containing the drilldown box on the navigation bar. This is only used when the MENU_DIRECTION is set to HORIZONTAL with horizontal style navigation bars and is used to alter the position of the drilldown box for different styles of navigation bar. Most frequently, you can position the drilldown box in an absolute position on the navigation bar, but you can also use effects such as altering the background-color to change the presentation of the control
NAV_BAR_LOGO_STYLE	This style is used to position the CompanyLogo.gif on the navigation bar, when the MENU_DIRECTION is set to HORIZONTAL
NAV_BAR_STYLE	This allows you to apply a CSS style to the navigation bar buttons.
NAV_BAR_GO_BUTTON	This places a Go button onto the navigation bar by the drilldown box
LOGO_STYLE	This is an optional CSS style that is applied to the ExtraViewLogo.gif image on the navigation bar if the MENU_DIRECTION is set to HORIZONTAL.
MENU_TEXT_COLOR	This allows you to select the color of text on the navigation bar, to ensure a contrasting color to the background

The following security permission keys control the ability for a user role to edit / alter their access to the options in the title bar:

Menu Bar Entry	Control
Account	The CF_PERSONAL_OPTIONS controls access to the user's account details and options. If the user's role has only read access to this key, then the user name will appear in the title bar, but they are not able to drill down into the personal options screen. Write permission is needed in order to be able to drill down into the personal options screen
Reports	The SR_MENUBAR_REPORTS security permission key controls the presence of this entry on the menu bar. It is worth noting that if you have an installation where each user has access to many hundreds of reports, there is a noticeable performance issue in generating this reports menu, each time the user hits a button that navigates to a new screen. In this environment, the recommendation is to turn this permission key off, and have the users reach reports via the Query button on the navigation bar. This one additional mouse click may save them one or more seconds each time they move to a new screen.
Role	If the user has only a single role defined, this title bar entry will not appear. If the user may adopt two or more roles, this entry appears and the user can drill down to a screen to alter their current role
Area and Project	The security permission keys named CF_AREA and CF_PROJECT control access to whether the user's role may alter their current business area and project

Background and Text Colors

The following screen shots give the names of many of the behavior settings that can be altered within the Display category of behavior settings.

Color Names

Color Names

Images

If ExtraView Corporation is hosting your installation, you do not have direct access to the file system of the server to configure, alter or use this feature without contacting ExtraView support.

Navigation and action buttons within ExtraView are created and stored as png files. They reside in two *images* directories, which are defined by the paths in the IMG_HOME and the IMG_NAV_BAR_HOME behavior settings. The IMG_NAV_BAR_HOME setting contains all the images required by the standard interface navigation bar. There is a sub-directory of this directory, named **workspace**, which contains a set of images with the same name, but used for the Workspace interface. The remaining images are stored in IMG_HOME. Any image can be replaced with an alternative or you may create a new directory and place a complete set of images within the new directory. You then define the path to the new directory in the IMG_HOME and / or the IMG_NAV_BAR_HOME behavior settings. Care should be taken not to provide images that are too large, and cause your browser to display them with unsatisfactory results. The following table shows the standard images that can be altered by the user, and their suggested size (width x height).

Note that the file named CompanyLogo.gif does not reside within the IMG_HOME or the IMG_NAV_BAR_HOME location, but sits directly under the directory named **/images**. This allows your company logo to sit in one place, but be accessible to any set of issues you select. Also note that there is an administrative function that allows you to upload the file CompanyLogo.gif, without needing access to the file system of the server.

If you want to create your own navigation bar image buttons, the original Adobe Photoshop files used to create the buttons for all themes are provided in the installed ExtraView directory, at *extraview_installed_directory*/web/en_US/images_nav_bar. Note that the navigation bar buttons for all workspace themes are stored in a single file named nav_bar_workspace.psd.

Navigation Bar Images in the IMG_NAV_BAR_HOME Directory

Note again that under this directory is a directory named **workspace**, which is where the navigation bar buttons for the workspace reside. Navigation bar buttons within the workspace are recommended to have a size of 60 pixels wide, by 40 pixels high.

Sample Image	Filename	Size	Purpose
Home	bHomeOff.png	V=100x25 H=80 x 30	The off state for the homepage button
Home	bHomeOn.png	V=100x25 H=80 x 30	The on state for the homepage button
Admin	bSystemConfigOff.png	V=100x25 H=80 x 30	The off state for the administration button
Admin	bSystemConfigOn.png	V=100x25 H=80 x 30	The on state for the administration button
Add	bAddProblemOff.png	V=100x25 H=80 x 30	The off state for the add issue button
Add	bAddProblemOn.png	V=100x25 H=80 x 30	The on state for the add issue button
Query	bSearchOff.png	V=100x25 H=80 x 30	The off state for the search/report button
Query	bSearchOn.png	V=100x25 H=80 x 30	The on state for the search/report button
Help	bHelpOff.png	V=100x25 H=80 x 30	The off state for the help button
Help	bHelpOn.png	V=100x25 H=80 x 30	The on state for the help button
Sign Off	bLogoffOff.png	V=100x25 H=80 x 30	The off state of the sign off button
	BannerBackground.png	V=any reasonable size x 85 H=130 x	Image background for the main menu

any reasonable size

ExtraView	ExtraViewLogo.gif	V=any reasonable size x 37 H=100 x any reasonable size	The ExtraView logo that appears on the navigation bar	
Go	GoButton.png	V=any reasonable size x 37 H=100 x any reasonable size	Optional button displayed by the drilldown box on the navigation bar	

Other Images in the IMG_HOME Directory

Sample Image	Filename	Size	Purpose
Add	AddButton.png	40 x 20	This is a general use add button used throughout
Desc	DescButton.png	40 x 20	Used to access the metadata for file attachments
Del	DeleteButton.png	40 x 20	The delete button on the layout editor screen or on a report
Delete	DeleteButtonWide.png	60 x 20	The delete button on the layout editor screen or on a report
Drag	DnDButton.png	40 x 20	Used as a handle for drag objects
Edit	EditButton.png	40 x 20	This is a general use edit button used throughout
Hide Categories	HideCategories.png	137 x 20	Used within administration screen lists
Hide Filters	HideFilters.png	137 x 20	Used within administration screen lists
Hist	HistoryButton.png	40 x 20	Used to access History from a report
Insert	InsertButton.png	40 x 20	The insert button on the layout editor screen
List	ListButton.png	40 x 20	This is a general purpose list button
Map	MapButton.png	40 x 20	Used in the import functions
QEdit	QuickEditButton.png	40 x 20	Used to access the Quickedit mode
Remove	RemoveButton.png	60 x 20	Used to remove issues from relationship groups
Save	SaveButton.png	40 x 20	General purpose
Set GUI	SetGUITheme.png	60 x 20	Used to set the theme within administration
Show Categories	ShowCategories.png	137 x 20	Used within administration screen lists
Show Filters	ShowFilters.png	137 x 20	Used within administration screen lists
Updt	UpdateButton.png	40 x 20	Update button in metadata screens
Upload	UploadButton.png	60 x 20	Used to upload files
View	ViewButton.png	40 x 20	The button used to view a detailed report

	NextButton.png	18 x 18	The button used to select the next page of a report
	PreviousButton.png	18 x 18	The button used to select the previous page of a report
	Calendar.png	19 x 18	Button used to access the calendar
	ColorPicker.png	19 x 18	Used to indicate the color-picker popup
	LinkButton.png	19 x 18	The button used to provide on-screen links to other functions
	PermissionsButton.png	19 x 18	Provides access to security permissions
	QmarkButton.png	19 x 18	Button used to access a pop-up list of users
	timerOff.png	19 x 18	Used for the timer field
	timerOn.png	19 x 18	Used for the timer field
	Style.png	19 x 18	Access to the style of an object
	AgingIcon.gif	19 x 18	Icon used to signify a report is an aging report
	ChartIcon.gif	15 x 15	Icon used to signify a report is a chart
	ContainerIcon.gif	15 x 15	Icon used to signify a report is a container report
	ExternalReportIcon.gif	15 x 15	Icon used to signify a report is an external report
	MatrixIcon.gif	15 x 15	Icon used to signify a report is a matrix report
	PageLayoutIcon.gif	15 x 15	Icon used to signify a report is a layout
	ReportIcon.gif	15 x 15	Icon used to signify a report is a columnar report
	StatisticsReportIcon.gif	15 x 15	Icon used to signify a report is a statistics report
	SummaryIcon.gif	15 x 15	Icon used to signify a report is a summary report
	AgingIconBig.gif	19 x 19	Icon used to signify a report is an aging report
	ChartIconBig.gif	19 x 19	Icon used to signify a report is a chart
	ContainerIconBig.gif	19 x 19	Icon used to signify a report is a container report
	ExternalReportIconBig.gif	19 x 19	Icon used to signify a report is an external report
	MatrixIconBig.gif	19 x 19	Icon used to signify a report is a matrix report
	PageLayoutIconBig.gif	19 x 19	Icon used to signify a report is a layout
	ReportIconBig.gif	19 x 19	Icon used to signify a report is a columnar report
	StatisticsReportIconBig.gif	19 x 19	Icon used to signify a report is a statistics report
	SummaryIconBig.gif	19 x 19	Icon used to signify a report is a summary report

	GrowButton.png	10 x 10	Button used to enlarge a text area when editing
	ShrinkButton.png	10 x 10	Button used to reduce the size of a text area when editing
	smallbullet.gif	10 x 10	General purpose small button
	ArrowSelectOff.png	12 x 12	Icon used to signify report columns that can be sorted
	ArrowSelectOn.png	12 x 12	Icon used to show the column currently used as the descending sort key for a report
	B0.gif...B9.png	20 x 20	Ten Separate gifs with the numerals 0 to 9, used in many places
	Arrow.gif	11 x 9	Used on the menubar to indicate entries with sub-menus
	promote.gif	15 x 15	Used in reports to indicate a group heading that may be promoted
	SelectorOff.gif	15 x 15	Used as the off indicator for the record selector
	SelectorOn.gif	15 x 15	Used as the on indicator for the record selector
	SortAscend.gif	15 x 15	Used to indicate ascending sort
	SortDescend.gif	15 x 15	Used to indicate descending sort
Add	TreeAddButton.gif	40 x 15	Tree button to add elements
Edit	TreeEditButton.png	40 x 15	Tree button to edit elements
Delete	TreeDeleteButton.png	40 x 15	Tree button to delete elements
View	TreeViewButton.png	40 x 15	Tree button to view elements
Opt-in	TreeOptInButton.png	40 x 15	Tree button to opt-in
Opt-out	TreeOptOutButton.png	40 x 15	Tree button to opt-out
	ChartLayout.gif	270 x 160	Part of PDF preparation
	GroupChartPic.gif	270 x 160	Part of PDF preparation
	GroupColDGPic.gif	270 x 160	Part of PDF preparation
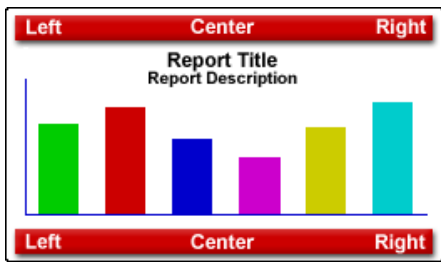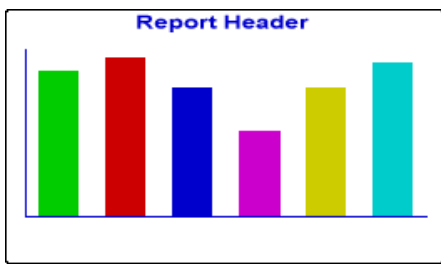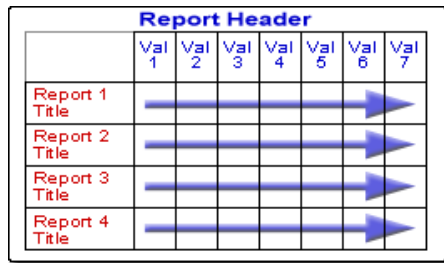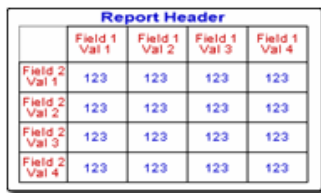
	GroupColPic.gif	270 x 160	Part of PDF preparation
	GroupColReportPic.gif	270 x 160	Part of PDF preparation
	GroupHeadersPic.gif	270 x 160	Part of PDF preparation
	GroupRowDGBPic.gif	270 x 160	Part of PDF preparation
	GroupRowPic.gif	270 x 160	Part of PDF preparation
	ReportLayout.gif	270 x 160	Part of PDF preparation
	ReportContainer.gif	150 x 100	Depicts continter report layout
	AreaProjectFlow.gif	500 x 250	Exemplifies the Area / Project structure
	TransitionArrow.gif	30 x 20	Used as part of the Status Transition diagram
	TransitionArrow.gif	20 x 20	Used as part of the Status Transition diagram
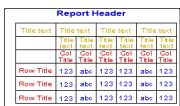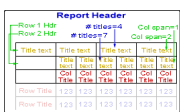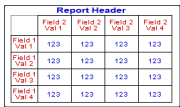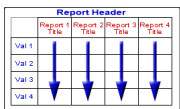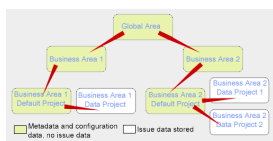
Main Screen Image Sets

To select any of these, alter the behavior setting named IMG_HOME in the Environment Settings menu beneath the Administration Advanced tab, and set the behavior settings to those shown. Consider using the themes to select a color set, as all settings are made for you.

Pale Blue Theme

These images offer theme of pale blue colors with light blue buttons.

Setting	Value
IMG_HOME	../images/images_light_blue/
BG_ALT_COLOR	#F7F7FB
BG_COLOR	#F0F1FC
BORDER_COLOR	#1292EA
LABEL_COLOR	#333366
MOUSEOVER_COLOR	#FAE992
REPORT_HEADER_FOOTER_COLOR	#ADBFD0
REPORT_SUMMARY_TOTAL_COLOR	#ADBFD0
SCREEN_HEADER_FOOTER_BG_COLOR	#D5DEE9
TAB_FONT_OFF_COLOR	#333366
TAB_FONT_ON_COLOR	#FFFFFF
TAB_OFF_COLOR	ADBFD0
TAB_ON_COLOR	3F5A7A
WINDOW_BG_COLOR	#FFFFFF

Dark Blue Theme

These images offer theme of rich blue colors with dark blue buttons.

Setting	Value
IMG_HOME	../images/images_dark_blue/
BG_ALT_COLOR	#EBF1F9
BG_COLOR	#E0EDF9
BORDER_COLOR	#1292EA
LABEL_COLOR	#405E87
MOUSEOVER_COLOR	#EFEFEF
REPORT_HEADER_FOOTER_COLOR	#B4C6F2
REPORT_SUMMARY_TOTAL_COLOR	#B4C6F2
SCREEN_HEADER_FOOTER_BG_COLOR	#D5DEE9
TAB_FONT_OFF_COLOR	#615499
TAB_FONT_ON_COLOR	#FFFFFF
TAB_OFF_COLOR	E0EDF9
TAB_ON_COLOR	3F5A7A
WINDOW_BG_COLOR	#FFFFFF

Light Blue Theme

These colors are a mix of light blues, with contrasting highlights.

Setting	Value
IMG_HOME	../images/images_light_blue/
BG_ALT_COLOR	#D5E3EB
BG_COLOR	#F5F5FF
BORDER_COLOR	#1292EA
LABEL_COLOR	#405E87
MOUSEOVER_COLOR	#E0E8F3
REPORT_HEADER_FOOTER_COLOR	#ADBFD0
REPORT_SUMMARY_TOTAL_COLOR	#ADBFD0
SCREEN_HEADER_FOOTER_BG_COLOR	#D5DEE9
TAB_FONT_OFF_COLOR	#FFFFFF
TAB_FONT_ON_COLOR	#FFFFFF
TAB_OFF_COLOR	ADBFD0
TAB_ON_COLOR	3F5A7A
WINDOW_BG_COLOR	#FFFFFF

Green Theme

A palette of green tones.

Setting	Value
IMG_HOME	../images/images_green/
BG_ALT_COLOR	#E8F7E7
BG_COLOR	#FEFFFE
BORDER_COLOR	#1292EA
LABEL_COLOR	#389538
MOUSEOVER_COLOR	#D4F0B4
REPORT_HEADER_FOOTER_COLOR	#98DB98
REPORT_SUMMARY_TOTAL_COLOR	#98DB98
SCREEN_HEADER_FOOTER_BG_COLOR	#DDE8A0
TAB_FONT_OFF_COLOR	#FFFFFF
TAB_FONT_ON_COLOR	#FFFFFF
TAB_OFF_COLOR	99CC99
TAB_ON_COLOR	389538
WINDOW_BG_COLOR	#FFFFFF

Grey Theme

This theme provides a mixture of grey tones complemented by dark red highlights.

Setting	Value
IMG_HOME	../images/images_grey/
BG_ALT_COLOR	#FAFAFA
BG_COLOR	#F6F6F6
BORDER_COLOR	#1292EA
LABEL_COLOR	#444444
MOUSEOVER_COLOR	#E7DBDD
REPORT_HEADER_FOOTER_COLOR	#BABABA
REPORT_SUMMARY_TOTAL_COLOR	#BABABA
SCREEN_HEADER_FOOTER_BG_COLOR	#DDDDDD
TAB_FONT_OFF_COLOR	#000000
TAB_FONT_ON_COLOR	#FFFFFF
TAB_OFF_COLOR	CCCCCC
TAB_ON_COLOR	990000
WINDOW_BG_COLOR	#FFFFFF

Sign On Screen

Consider this example sign on screen:

It is composed of the following areas:

These areas are configurable in the following way:

Sign On Logo Area

This is a single image whose location is defined by the behavior setting named SIGN_ON_SCREEN_LOGO. The default for this is the image named CompanyLogo.gif, but you can set this to any valid image file. You can apply a CSS style to the image with the behavior setting named SIGN_ON_SCREEN_LOGO_STYLE. Typically you can use this style to provide an absolute position to the image

Promo Area

This is an optional area defined by a screen type field in the data dictionary named PROMO. This field contains HTML which can be used to create any valid entry on the sign on screen. Note, you can insert JavaScript into the screen with this field

Sign On Area

This area is generated internally by ExtraView and its presentation may not be altered. However, it's background color is set by the display theme chosen

Display Area 1

This is an optional area that is insert onto the sign on screen within an IFRAME. It may contain any valid HTML. The HTML is located on a server at the location defined in the behavior setting named DISPLAY_SIGNON_URL_1. The size of the IFRAME area

is 500 pixels wide by 600 pixels high. The behavior setting named DISPLAY_SIGNON_PROD_INFO must be set to a value of YES to turn this feature on

Display Area 2

This is an optional area that is insert onto the sign on screen within an IFRAME. It may contain any valid HTML. The HTML is located on a server at the location defined in the behavior setting named DISPLAY_SIGNON_URL_2. The size of the IFRAME area is 350 pixels wide by 300 pixels high. The behavior setting named DISPLAY_SIGNON_PROD_INFO must be set to a value of YES to turn this feature on

Cascading Style Sheets

If ExtraView Corporation is hosting your installation, you do not have direct access to the file system of the server to configure, alter or use this feature without contacting ExtraView support.

Each user can select between one of three font sizes as a personal option. These are named small, medium and large. When the user selects one of these, ExtraView loads a corresponding cascading style sheet (CSS). The style sheets installed by ExtraView are within your web server environment, defaulting to the directory named IMG_HOME/stylesheets. This contains a number of sections with font sizes and other style parameters.

If you want to create your own version of the style sheets, it is typical that you want to do this in conjunction with creating your own image set for all buttons. It is recommended that you retain the same structure of the existing image sets and stylesheets by creating a new directory and subdirectory within locales/en_US/images and copy an existing image set into this directory. You use the IMG_HOME behavior setting to point to this new directory, and then make all your changes within there. In this way future upgrades of ExtraView will not write over your files, although you may need to extend these files if new entries, images or stylesheet settings are part of the upgrade. Note that if you are using or creating localized versions of the image sets, you will need also to place the stylesheets in other locale directories. Your stylesheets/ directory must contain the following files:

File	Purpose
small.css	The main stylesheet when the user selects the *small* text personal option
medium.css	The main stylesheet when the user selects the *medium text personal option*
large.css	The main stylesheet when the user selects the *large text personal option*
dnd.css	The style used to control the presentation of drag and drop objects
menustyle.css	The style of the menu used at the top of most screens
popupwindow.css	This style us used for window popups that are generated from within the browser, as opposed to being generated from the server
popupwindowsafari.css	This is similar to popupwindow.css, but is only used for the Apple Safari browser
tooltip.css	The style of popul tooltips that apper on many screens

Typically you will not add new or alter existing style names, but you can alter the existing styles. Also, you must be careful in altering the existing styles as some of these are used for accurate positioning or sizing of objects on ExtraView screens. As a rule of thumb, you can alter colors safely, but you should be careful about altering all other style information.

Consult a good HTML or Cascading Style Sheet manual for full details of options available to set up alternate styles.

Import / Export Menu

This section of administration deals with the import and export of data and metadata with ExtraView. The available functions are:

- Metadata export - Exports the configuration, or part of the configuration of the installation to a local file
- Metadata import - Imports a previously exported file into an installation
- File Import - Issue Data - Imports issue data from a CSV or TSV file
- File Import - User Information - Imports and creates users from a CSV or TSV file
- File Import - Data Dictionary - Imports and create new user defined fields in the data dictionary from a tab-delimited or comma-delimited file

Note: It is strongly recommend that you backup your entire database using its standard features, before you use the import facility. Importing metadata, layouts, reports and user data is irreversible, and it is possible that failures can occur during the process.

Note than exporting metadata is different from backing up your entire database. This should be performed with the tools that are provided by your database vendor, or third party tools that are specifically developed for the task.

Data within ExtraView is split into two parts, metadata and item data. Metadata is the supporting data for the application, such as information supplied as part of the product, data you supply such as the entire list of available product names, the entire list of possible status values, and the screen layouts you design. Item, or issue, data is the physical data relating to the items being tracked. This will contain the attributes of a specific issue such as its ID, its specific product name, specific status currently assigned, and all the history information relating to the issue.

There are several reasons to export and import just the metadata, within an ExtraView instance:

- To produce a backup of all or part of the metadata in a form that can be restored. This is separate from performing a backup of the database instance itself, as the metadata is exported in XML format.
- To transfer the metadata from a test or staging environment to a production database. Upon transfer, operations such as update or update and merge can be selected.
- To obtain the data in a standardized format (XML) that can be used to interface to other systems.

When exporting the metadata, there is an import overall option to understand. You may check the option box titled **Minimize the user information exported**. Doing so serves the purpose of leaving the data associated with users undisturbed on the production server. User data such as passwords and each user's personal options is dynamic and you most often want to leave the production version of the user information alone.

It helps, but is not essential, to have an understanding of the ExtraView schema when using the metadata transfer utility. When you export information, all the tables that are associated with the family are also exported, including subordinate information. When you import information, there are three modes, as follows:

- **Update** – modify existing objects of same name and type from the export image
- **Update / Merge** – this is a combination of update plus a merging of new data items in the export file.
- **Localization Merge** – import only the messages for all languages, which were exported as a family

Note: For most common import purposes, you should use the Update / Merge facility, as this is most likely to produce the desired results of adding new data, whilst updating existing data.

When an object is loaded from the saved disk image to the instance, all necessary referential integrity and uniqueness constraints are maintained. Maintaining uniqueness constraints necessarily involves incrementing or modifying the sequence values for specific objects in the instance appropriately. Maintaining referential integrity constraints implies that multiple tables may be updated, because rows on which the new values depend must be added, if they are not already there.

When deleting objects in the target database, all referential integrity constraints are maintained, using a cascade deletion policy. That is, if an import results in the removal of an object, the metadata object references to that object must be removed as well. This strategy is designed to ensure information is unlikely to be lost, since dependent objects are likely be restored during the import.

When adding new metadata, additional rules may be imposed on the data relationships for user problem data. In these cases, all existing issue data is checked against the new relationships.

When you move the business rules from one instance to another as part of the export / import process, the rules are never merged with the existing rules in the target database. The old rules are always overwritten with the new rules.

Export Family Information

Family	Tables exported	
All Metadata	ALLOWED_FUNCTIONS	LAYOUT_TYPE
	ALLOWED_VALUE_TYPE	MODULE
	ALLOWED_VALUES	MODULE_TYPE
	APPLICATION_DEFAULT	OUTPUT_TYPE
	AREA	PRIORITY
	CATEGORY	PRIVACY_GROUP

CHART_TYPE	PRIVACY_GROUP_USER
CUSTOMER	PRODUCT
DATA_DICTIONARY	PRODUCT_LINE
ESCALATION_RULE	PRODUCT_PRODUCT_LINE
ESCALATION_RULE_ACTION	PRODUCT_RELEASE
ESCALATION_RULE_ELEMENT	PROJECT
ESCALATION_RULE_USER	RELATIONSHIP_GROUP
EV_FILE_IMPORT	RESOLUTION
EV_HIERARCHY	SECURITY_GROUP
EV_HIERARCHY_LEVEL	SECURITY_GROUP_USER
EV_LIST_MAP	SECURITY_MODULE
EV_RULE_TEXT	SECURITY_PERMISSION
EV_TEMPLATE	SECURITY_USER
EV_TEXT_LOOKUP	SEVERITY_LEVEL
INTEREST_LIST	START_PAGE
INTEREST_LIST_ELEMENT	STATUS
INTEREST_LIST_USER	STATUS_RULE
ITEM_GROUP_TYPE	TITLE_MAP
ITEM_TYPE	UDF
LAYOUT	UDF_LIST
LAYOUT_ELEMENT	
LAYOUT_ELEMENT_ATTRIBUTE	

All metadata minus reports

ALLOWED_FUNCTIONS	LAYOUT_TYPE
ALLOWED_VALUE_TYPE	MODULE
ALLOWED_VALUES	MODULE_TYPE
APPLICATION_DEFAULT	OUTPUT_TYPE
AREA	PRIORITY
CATEGORY	PRIVACY_GROUP
CHART_TYPE	PRIVACY_GROUP_USER
CUSTOMER	PRODUCT
DATA_DICTIONARY	PRODUCT_LINE
ESCALATION_RULE	PRODUCT_PRODUCT_LINE
ESCALATION_RULE_ACTION	PRODUCT_RELEASE
ESCALATION_RULE_ELEMENT	PROJECT
ESCALATION_RULE_USER	RELATIONSHIP_GROUP
EV_FILE_IMPORT	RESOLUTION
EV_HIERARCHY	SECURITY_GROUP
EV_HIERARCHY_LEVEL	SECURITY_GROUP_USER
EV_LIST_MAP	SECURITY_MODULE

EV_RULE_TEXT	SECURITY_PERMISSION
EV_TEMPLATE	SECURITY_USER
EV_TEXT_LOOKUP	SEVERITY_LEVEL
INTEREST_LIST	START_PAGE
INTEREST_LIST_ELEMENT	STATUS
INTEREST_LIST_USER	STATUS_RULE
ITEM_GROUP_TYPE	TITLE_MAP
ITEM_TYPE	UDF
LAYOUT	UDF_LIST
LAYOUT_ELEMENT	
LAYOUT_ELEMENT_ATTRIBUTE	

Layout		
	AREA	PROJECT
	DATA_DICTIONARY	SECURITY_GROUP
	LAYOUT	SECURITY_MODULE
	LAYOUT_ELEMENT	SECURITY_PERMISSION
	LAYOUT_ELEMENT_ATTRIBUTE	TITLE_MAP
	LAYOUT_TYPE	UDF
	MODULE	UDF_LIST
	PRODUCT	

Report		
	AREA	PROJECT
	CALCULATED_FIELD	RELATIONSHIP_GROUP
	CATEGORY	REPORT
	CHART	REPORT_ATTRIBUTE
	CHART_PROPERTY_GROUP	REPORT_GROUP
	CHART_TYPE	REPORT_LAYOUT
	DATA_DICTIONARY	REPORT_SECURITY_GROUP
	EV_HIERARCHY	RESOLUTION
	EV_HIERARCHY_LEVEL	SECURITY_GROUP
	FILTER	SECURITY_MODULE
	FILTER_CRITERIA	SECURITY_PERMISSION
	FILTER_GROUP	SECURITY_USER
	ITEM_TYPE	SEVERITY_LEVEL
	LAYOUT	SORT_ORDER
	LAYOUT_ELEMENT	SORT_ORDER_FIELD
	LAYOUT_ELEMENT_ATTRIBUTE	START_PAGE
	LAYOUT_TYPE	STATUS
	MODULE	SUBREPORT
	OUTPUT_TYPE	TITLE_MAP

	PRIORITY	UDF
	PRIVACY_GROUP	UDF_LIST
	PRODUCT	

User	AREA	START_PAGE
	PROJECT	TITLE_MAP
	SECURITY_USER	

Text Messages	EV_TEXT_LOOKUP	TITLE_MAP

List Value Migration

The default export and import of list fields and their values moves all values in all lists from the source to the target instances. There are some circumstances where you do not want to migrate any or all of the list values within fields. For example, if you are creating field list values through the use of the OBJECT business rule, this might be test data and not data that you want to replicate in the target system. To accommodate this requirement, two features are available.

First you may use the DO_NOT_MIGRATE=true parameter within an OBJECT rule. This flags the values created for special treatment with the XML Export / Import process. Second, the **Manage List Values** utility accessible through the administration utility, the data dictionary and the design center allows you to set any list value with a flag so that it will not be migrated.

Because values being migrated may have dependencies, such as being used in interest lists and escalations, the flagged items are first migrated to the target system, and are then deleted, with the deletion also removing the dependencies.

Import of Item Data

There are three modes of importing item data from an XML format.

- **Batch Mode**. In this mode, an XML formatted input file is processed sequentially, creating new issues in the ExtraView database. This is driven from an administration screen.
- **API Command**. In this mode, an API command is executed, passing a single issue to ExtraView as part of the HTTP data stream. This issue is processed and with success is added to the database. The API command returns a success message or a failure error message.
- **API Command with Input File**. In this mode, an API command is executed. The command references an input file with one or more issues to be imported to ExtraView. The file is processed and results are passed back to the calling command.

The XML data to be imported must adhere to the Document Type Definition (DTD) detailed below. This is supplied with ExtraView, in the WEB-INF/data/xml_dtd directory of your installation, and can be directly referenced from your XML import file or data stream. The file name supplied is extraview_item.dtd.

```
<?xml version="1.0" encoding="UTF-8" ?>

<!ELEMENT locale (EMPTY) >
<!ATTLIST locale region CDATA "US">
<!ATTLIST locale language CDATA "en">
<!ATTLIST locale variant CDATA " ">

<!ELEMENT item_list (item*) >
<!ELEMENT item (short_descr , severity_level , priority , status , product_name , date_created , owner , timestamp , assigned_to ,
privacy , last_change_user , alt_id , area , project , category , resolution , product_line , date_last_status_change , date_closed ,
release_found , release_fixed , contact , originator , item_id, item_udf*, item_release*, item_module*, item_attachment* )>

<!ELEMENT item_udf (title_specifier?, name_specifier?, CDATA*)>

<!ELEMENT item_release (title_specifier?, name_specifier?, short_descr , severity_level , priority , status , product_name ,
date_created , owner , timestamp , assigned_to , privacy , last_change_user , alt_id , area , project , category , resolution ,
product_line , date_last_status_change , date_closed , release_found , release_fixed , contact , originator , item_id, item_udf*)>
```

```
<!ELEMENT item_module (title_specifier?, name_specifier?, assigned_to ,status ,timestamp , rc_version ,last_change_user ,item_module_id )>
```

```
<!ELEMENT title_specifier (CDATA)>
```

```
<!ELEMENT name_specifier (#PCDATA)>
```

```
<!ELEMENT short_descr (CDATA) >
```

```
<!ELEMENT severity_level (CDATA) >
```

```
<!ELEMENT priority (CDATA) >
```

```
<!ELEMENT status (CDATA) >
```

```
<!ELEMENT product_name (CDATA) >
```

```
<!ELEMENT date_created (CDATA) >
```

```
<!ELEMENT owner (CDATA) >
```

```
<!ELEMENT timestamp (CDATA) >
```

```
<!ELEMENT assigned_to (CDATA) >
```

```
<!ELEMENT privacy (CDATA) >
```

```
<!ELEMENT last_change_user (CDATA) >
```

```
<!ELEMENT alt_id (CDATA) >
```

```
<!ELEMENT area (CDATA) >
```

```
<!ELEMENT project (CDATA) >
```

```
<!ELEMENT category (CDATA) >
```

```
<!ELEMENT resolution (CDATA) >
```

```
<!ELEMENT product_line (CDATA) >
```

```
<!ELEMENT date_last_status_change (CDATA) >
```

```
<!ELEMENT date_closed (CDATA) >
```

```
<!ELEMENT release_found (CDATA) >
```

```
<!ELEMENT release_fixed (CDATA) >
```

```
<!ELEMENT contact (CDATA) >
```

```
<!ELEMENT originator (CDATA) >
```

```
<!ELEMENT item_id (CDATA) >
```

```
<!ELEMENT rc_version (CDATA) >
```

```
<!ELEMENT item_module_id (CDATA) >
```

```
<!ELEMENT item_attachment (CDATA)>
```

```
<!ATTLIST item_attachment file_desc CDATA #IMPLIED >
```

```
<!ATTLIST item_attachment file_name CDATA #IMPLIED >
```

```
<!ATTLIST item_attachment path CDATA #IMPLIED >
```

```
<!ATTLIST item_attachment content_type CDATA #IMPLIED >
```

<!ATTLIST item_attachment file_size CDATA #IMPLIED >

<!ATTLIST item_attachment date_created CDATA #IMPLIED >

<!ATTLIST item_attachment created_by_user CDATA #IMPLIED >

Note that all data related to items, including repeating records, UDF's, and attachments are described with the DTD and can all be imported by ExtraView. A portion of a sample XML file is shown below, with two issues to be imported. This references the above DTD. Note that the attachment within the example import file is shortened, for brevity. The following points should be noted about the import of items:

- The DOCTYPE refers to the DTD and the path should correspond to the path where it is stored on your server.
- UDF's can be referred to by NAME or by TITLE. We recommend that you use the NAME wherever possible, as the TITLE field may not be unique across your installation. Furthermore, the TITLE may vary with the locale specified in the DTD. You may create different DTD's for each locale, and use the localized TITLE within the XML files being imported.
- We recommend that all character data is embedded within CDATA tags.
- Certain constructs are required in the XML, following the web interface rules. For example, a file-name-attribute should always appear in the item-attachment-attributes list.
- UserCustom methods prAddPreInsert and prAddPostInsert are executed on each item insert.
- Allowed values between fields are ignored when you import a record via this interface. If you mistakenly create an invalid parent – child relationship, this will be indicated when you try to update the record via the web interface.
- The ITEM_ID is allocated to the new issue at the time the ExtraView code validates the issue being read from the XML input. If the record is rejected for any reason, the ITEM_ID is discarded, and an issue with this ID will never be created. The next record to be imported will be allocated the next number in the sequence.
- If a field is read-only, on the add screen layout (as defined by the user role, business area and project of the issue), then this will be respected with the item import function. Further, if this field has a default value defined in the data dictionary, this value will be inserted.
- Attachments must be encoded using the Base64 algorithm.
- Values in other CDATA sections may be encoded in Base64 if desired or required. Whether or not the string is encoded is signaled by a sentinel ("%25S") in the front of the CDATA string. Encoding is required if any of the following arise:
    a. The string value starts with the sentinel value ("%25S"), or
    b. The string value contains the CDATA end marker ("]]>"), or
    c. The string value contains any non-CDATA-permissible characters. CDATA-permissible characters are defined using the following production with Unicode character values as derived from the web page at http://www.w3.org/TR/REC-xml - NT-Char

    ```
    Char ::= #x9 | #xA | #xD | [#x20-#xD7FF] | [#xE000-#xFFFD] | [#x10000-#x10FFFF]
    ```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE extraview_item SYSTEM "file:///C:/path_to_my_dtd/extraview_item.dtd"> <ITEM_LIST>
<ITEM>
<AREA><![CDATA[Bugs]]></AREA>
<PROJECT><![CDATA[&Bugs Data&]]></PROJECT>
<CATEGORY><![CDATA[Software]]> </CATEGORY>
<PRODUCT_NAME><![CDATA[ExtraView]]></PRODUCT_NAME>
<ITEM_UDF>
<TITLE_SPECIFIER><![CDATA[OS]]></TITLE_SPECIFIER>
</ITEM_UDF>
<ITEM_RELEASE>
<RELEASE_FOUND><![CDATA[1.2.3.4]]></RELEASE_FOUND> </ITEM_RELEASE>
<ITEM_RELEASE>
<ITEM_UDF>
<TITLE_SPECIFIER>
<![CDATA[This is text in a field]]>
</TITLE_SPECIFIER>
<![CDATA[BILL.SMITH]]>
</ITEM_UDF>
</ITEM_RELEASE>
</ITEM>
<ITEM>
<CATEGORY>
<![CDATA[Software]]>
</CATEGORY>
<PRODUCT_NAME>
```

```
  <![CDATA[ExtraView]]>
  </PRODUCT_NAME>
  <ITEM_UDF>
  <NAME_SPECIFIER>OP_SYS</NAME_SPECIFIER>
  <![CDATA[Solaris]]>
  </ITEM_UDF>
  <ITEM_RELEASE>
  <RELEASE_FOUND>
  <![CDATA[2.3.4.5]]>
  </RELEASE_FOUND>
  </ITEM_RELEASE>
  <ITEM_RELEASE>
  <ITEM_UDF>
  <NAME_SPECIFIER>RELEASE_COMMITTED_RELEASE
  </NAME_SPECIFIER>
  <![CDATA[5.2.2.1]]>
  </ITEM_UDF>
  <RELEASE_FOUND>
  <![CDATA[2.5.6]]>
  </RELEASE_FOUND>
  </ITEM_RELEASE>
  <ITEM_ATTACHMENT FILE_DESC='Current log file' FILE_NAME='EVJ.log' PATH='C:/t/' CONTENT_TYPE='text/plain' >
  <ATA[VEhJUyBJUyBBIEZJTEU=]]>
  </ITEM_ATTACHMENT>
  </ITEM>
  </ITEM_LIST>
```

Export of Item Data

To achieve the export of data from ExtraView, the user should use the Command Line Interface (CLI) commands, evhist, evget and evsearch. These commands are covered in depth in the Command Line Interface Guide.

Note: For this version of ExtraView, the above import and export functions use different DTD's. Thus, records exported from ExtraView cannot be imported with the functions documented here directly. A future version will offer compatibility of import and export XML data, by changing the CLI commands.

Metadata Export

The export creates a flat file containing the metadata in an XML format that can be read by the Metadata Import administration utility. This file can be moved between different platforms and different instances of ExtraView. The order of objects in the file is defined by the requirements of ExtraView, not by the user. The data export is defined satisfying the requirements of being able to perform an import of the data, to build new objects. In short, all dependent data must follow the data upon which it is dependent.

There are a wide range of options, allowing you to perform exports of all or subsections of the metadata within an ExtraView installation.

Each option that defines a subset of the database is termed a family. This option is available from the **Import/Export** tab of administration.

Note: If you are using ExtraView with the Apache Derby database, the metadata export feature is significantly limited. You may only perform a metadata export of the complete database, and cannot use the features described below to constrain the export to small parts of the database.

Exporting ExtraView Families

The following are the metadata families that may be exported. Note that any filters you set will also apply to the export file you create:

- Export all metadata - a complete export is created of all the metadata

- Export all metadata tables except reports - a complete export less the reports within the source system is created. Your target production system may have reports which differ from your source system, and you may not want to touch the reports on the production system

- Export layouts and supporting information - this will provide the layouts within the filters you apply, along with all supporting information such as the data dictionary fields

- Export reports and associated data - this option only exports the reports and their supporting information, such as the fields in the source system

- Export business rules - only the business rules in the souce system will be exported

- Export user profile information- only the user information in the source system will be exported

- Export text messages - this will export all the message tables for one or more locales. If you are working within a system with the behavior setting MULTI_LOCALE set to YES, a select list showing all the allowed locales will appear on the administration screen. The default is that all locales will be exported, but you can use the multi-select list to export just one or any selection of the allowed locales. The typical use of this feature allows the administrator to migrate the messages for just a single locale when upgrading a system and several language translations are happening in parallel. Only one or more languages may be ready to ready to migrate so you may control which locales are migrated. Note that the base locale of the system is always exported in addition to the selected locales. Once exported, you can use the generated file to import a new language set into a target system.

   Note that when exporting text messages different handling is applied to User Defined Fields which have values flagged with the **Do Not Migrate** checkbox. This flag is ignored for the export so that all values for all fields are exported. If the User Defined Field and/or the value in the export file does not exist in the target environment, they are then ignored when the import takes place. This allows for the migration of localized values if and only if the value for the base language exists in the target environment.

- Export a solution - this option allows the preparation of an export file which will be used to migrate an ExtraView solution to one or more remote systems. The option minimizes information outside of your solution from the source database. What will be exported is confined to the following:

- Only the layouts within the source system that are within the Business Area(s) you select as filters. Layouts in the global area will not be exported
- Only the fields and their permissions that are used by layouts within the source system will be exported
- Allowed values for any fields within the layouts will be exported
- Relationships used by the layouts will be exported
- Only reports within any folders you select will be exported

If you click the option **Minimize the user information exported**, then only the essential user information to export the metadata is written to the export file. This may solve a number of issues. For example, user information is often dynamic, and users may frequently change passwords, or administrators may activate and deactivate accounts on a production system or the references to the user information must always be taken from an LDAP or Active Directory server. While you may want the metadata from a development system to be migrated to a production system, you often may want to leave the production data related to users in place. That being said, users who are referred to within metadata such as filter values, interest lists, escalation rules, etc. are exported, so the information can be reproduced in the target system when the import occurs. Users only referred to within metadata such as the date created and date last modified fields are not exported when the user information is minimized.

You may choose some overriding filters for the export. As opposed to exporting the data across all Business Areas and Projects, you may select a single Business Area and Project to export. You may also choose to only export data updated since a specific date and time.

Once you have set up an export profile, you may decide to save the combination for future use. This is accomplished through the **Load / Manage Export Profiles** button on the menu bar. This will display a popup as follows:
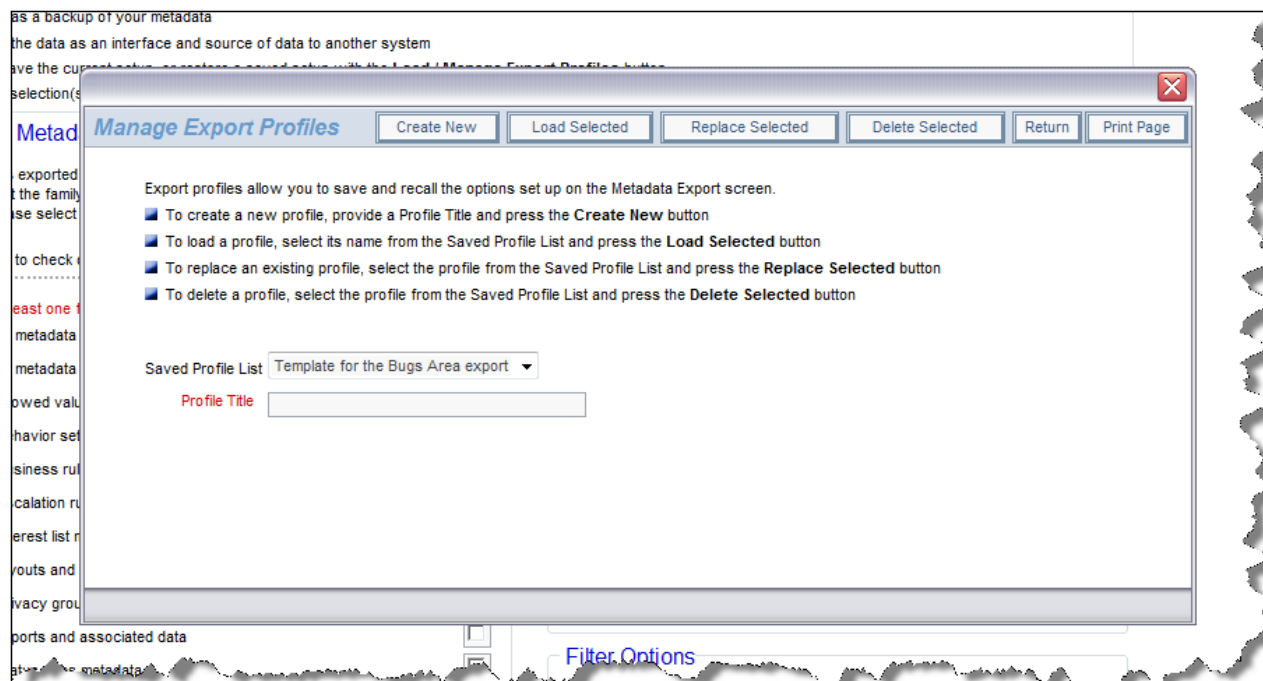
You may create new profiles and save them, load an existing template, replace an existing template or delete an existing template from the popup window. Once you have set up the profile to export, click the **Continue with Export** button. You are taken to a screen similar to this:

Continuing with the metadata export process

Confirm your choices on this screen, before clicking the **Perform Export** button.

You will be prompted to enter a file name for the export. This file will be saved on your local file system of your client computer.

An export can take some time, based upon the amount of metadata in your system. Files generated vary in size from a few to more than 50MB in size, dependent on the amount of metadata. As the export starts, one of the first steps is to compare the database against a known reference set of data that describes what the database should look like. In a perfect world, this step would not be needed, but databases can accumulate defects caused by any one of a number of factors. A previous update script might have failed, but the error not being noticed by the administrator; there might have been an incorrect command executed by a database administrator. There are other potential sources of failure as well. The metadata export process examines the database to ascertain if there is any obvious problem that will interfere with the process. The most typical problem encountered is that there is a missing database constraint. When a problem such as this is found, you will see an alert message warning of the event. You will be able to elect to continue, or to stop the process. It is recommended that you do not continue if you are not sure of the ramifications of the error.

An export file can be imported into the same or a different instance of ExtraView. The file can also be used to integrate with other applications.

Migrating Text Messages

It is worth discussing how text is migrated from one instance to another and how it is handled in a multi-lingual system. There are several options that can be used, but are dependent upon how you export the metadata.

There are two principal options available when you want to migrate messages from one instance to another.

- Use the **Export system messages** family option to prepare the XML file. This results in a format which can be imported into the target installation without regard to the context of metadata messages. For example, if you have a single term in your base locale (usually English) that occurs more than once within different metadata items such as list or field titles and the translator used different terminology to localize the term for each occurrence, then the import process ignores this, and the target instance will have the same localized term wherever it occurs. The advantage is that this is a relatively fast migration path
- Use the **Export all metadata** family option to prepare the XML file. This results in a format which retains the context of each and every term. For example, if you have used different terms in a language that all correspond to the same term within the base locale (usually English), then the target system will retain the separate values. When importing the metadata file, you may choose the **Localization Update/Merge** option to migrate only the text messages from the export file. The disadvantage is that this migration may take longer to execute than the first option. You may still perform a full **Update/Merge** or **Merge** when importing the metadata into the target instance.

Items that are Not Exported

There are a number of behavior settings that are not exported with this utility. This is deliberate, as it is most likely that these settings are specific to the target environment and may be different from those in the source environment. The list of settings that are not migrated is:

- APP_HOME

- CSS_HOME
- DEBUG_MODE
- DEFAULT_LANGUAGE
- DEFAULT_REGION
- DEFAULT_VARIANT
- EMAIL_DIRECTORY
- HELP_HOME
- HTTP_CHARSET
- IMG_HOME
- LOG_DIR
- NOSPILL_SESSION_COUNT
- SCHEMA_VERSION
- SITE_URL
- URL_PATH
- USER_ACCEPTS_LICENSE

Metadata Import

- Add new comment

Note: Again, we strongly recommend that you backup your data before you use the import facility, using your database's standard facilities. Importing metadata, layouts, reports and user data is irreversible, and it is possible that a failure can occur during the process.

The import menu is situated within the Import/Export tab of the administration menu, and allows the administrator to specify the file to load.

The import function reads a flat image file (which is in XML format) and updates the tables as required by the administrator's selection of objects to load. The utility takes into account the relational dependencies and all internal sequence renumbering as needed, ensuring consistent and correct instance updates.

Validity checking for data adherence to new rules is performed as early as possible, thereby minimizing the impact of some validity failure that may cause termination of the import.

Most imports will utilize the Update / Merge operation. This will update records that exist, and merge into the database the new records that were part of the file being imported.

Although it is possible to create your own file to be used as an import file to ExtraView, extreme care should be taken before attempting to do this, and it is essential that you have a complete and thorough knowledge of the ExtraView schema. Often, it may appear straightforward to compose an XML file, or to edit an existing file that was created by ExtraView's own Export feature, but there are many relationships described in the file that are not obvious and that must be maintained for an import to be successful. We suggest that before attempting to create your own import file by modifying an existing file or creating it from scratch, that you consult with ExtraView support. In any case, **always back up your target database before performing any import.**

Starting the importing process

Follow the instructions on the screen. As displayed, the principal import methods are:

Update / Merge	This combines operations of update and merge, Records that already exist in the target schema are updated, and a new record is inserted if record being imported does not exist.
Merge	This leaves existing records in the target schema alone, and adds new ones from the file that is being imported. ExtraView will use new sequence numbers for the operation within the database.
Localization Update/Merge	This mode only imports the localized messages from the import file, and does not touch the remainder of the metadata. This is typically used to move localized messages from one installation to another, providing just the updated localized messages to the target installation.

If you have altered metadata such as the titles of objects in lists, within the destination database, and these same modifications to the data were not made in the source database from which the export was taken, ExtraView gives you an opportunity to remap the fields, as opposed to creating new fields from the import file, and inserting these into the destination database. To perform this check, make sure the **Map unmatched titles** prompt is checked.

When you perform the check, the import is split into two parts. First the uploaded file is preprocessed, and each list value in the import is checked against the target destination database. If there are any values in the import file that do not exist in the destination database, then you will be shown the values from the import file that do not have a matching value, and you will be able to select from the values in the target database that do not have a value specified in the import file. After selecting the values to map, if there are discrepancies, you then proceed to the second part of the import. The file is processed again, the destination database is updated, and the mappings you selected are applied.

When importing data into a business area, if the import process is able to create a project with an ID numbered 0, and the project ID in the import file is 0, then a project ID of 0 is used and mapped into the target database. If a project ID of 0 already exists in the target business area, the project ID of 0 in the import file is mapped to a new non-zero project ID. In any case, the matching project title within the business area and the mapped project overrides these decisions.

This is important, as a project ID of 0 is used for inheritance of layouts, permissions and other objects for the non-zero project ID's within the business area, and ExtraView is trying to ensure that a project ID of 0 always exists for the inheritance to be allowed to work.

There is one circumstance in which the import process may have difficulty in resolving all the data from the export file. If you are not importing user data, and the information in the import file contains references to users that do not exist in the target database, then import errors may result. This is especially noticed if the missing user was the person who created or last updated items such as layouts. These will not be imported, as ExtraView must maintain referential integrity of the users who are connected to its internal objects.

In general, allowed value lists that have an invalid current value in parent or child will cause an import error and will not be imported. This may result in incomplete allowed value lists.

Results of preprocessing   Print Page

Update/Merge: Combines update and merge, updates records if they exist in the target schema, and inserts new records if they do not.

Number of rows read   36,893

Number of rows inserted   0

Number of rows updated   0

Number of tables read   49

No mapping errors were found during preprocessing. You should proceed with the import. Once again, do not interrupt the process.

Proceed to import uploaded XML file with mappings   Return   Print Page

Results from preprocessing the import file

While data is being preprocessed or imported, a status bar is shown as follows:

Preprocessing XML input record 12,258 of 37,337

Do not interrupt this process

Progress bar during the importing process

Note: Importing the data can take some time, from ten minutes in smaller installations, to thirty minutes or more in larger installations. It is dependent upon the size of the file being processed, factors such as the number of fields, layouts and permissions, and the speed of the server loading the data. Due to the inherent limitations of browsers, there is little feedback during the initial part of the process, when the browser is uploading the import file from the client machine to the server. It is recommended that you prohibit users from using ExtraView during this time, especially if you are importing layouts. You can do this by locking users out of the system while you are importing the data. You can achieve this with the **Disable and Enable User Access** on the **Users** tab of the administration menu.

Note: You can improve the time to execute the import operation, by zipping the import file. Exported files compress significantly (perhaps to less than 10% of their original size). ExtraView will look for the XML file contained within a zip file as long as the name of the zip file and the name of the import file are the same.

Note: Once the import operation is started, you must not interrupt it. Once the actual import has begun, ExtraView will not cancel the server operation, even if you close your browser.

Note: When your file is imported, there is additional information within the file, termed a manifest. This manifest is used as a check against the data that you import, and provides a report if there are any unanticipated differences between the imported information and the database into which it is being imported.

After the import is complete, you will see a summary screen that shows the number of records imported into each database table. Any errors encountered will be displayed here. Note that the results of the import vary according to the exact tables imported from the file you uploaded. There may be anything from a few to more than one hundred tables displayed in the results.

Summary screen of the imported data

Handling User Data

Users frequently change their personal settings when using ExtraView. For example, users may change their Home Page reports, their display format for the timestamp, their password, or one of many other attributes. If you export the metadata from a development system, wait some time and then import this metadata to a production system, you may overwrite recent changes made by users in their personal data. A behavior setting named OMITTED_IMPORT_USER_COLUMNS allows the administrator to set up specific columns that will be ignored upon the import of the metadata. The value of this field is a comma-separated list of column names. Note that you set this in the target system, not the system from which you are exporting the data. The possible values you can add to this list are as follows:

Column Name	Meaning	Notes
ADDITIONAL_EMAIL	The user's secondary email address	Recommend this field's omission
ADDITIONAL_EMAIL_ON	Switch to indicate whether the user's secondary email address is turned on or off	Recommend this field's omission
ADDRESS_LINE1	Address line 1	Recommend this field's omission
ADDRESS_LINE2	Address line 2	Recommend this field's omission
CELL_PHONE	Cell phone number	Recommend this field's omission

CHART_FONT	Chart font	Recommend this field's omission
CITY	City	Recommend this field's omission
COMPANY_NAME	Company name	Recommend this field's omission
COUNTRY	Country	Recommend this field's omission
CREATED_BY_USER	The user ID of the person who created this user's account	
DATE_CREATED	The date the user's account was created	
DATE_FORMAT	Format to display dates	Recommend this field's omission
DRILLDOWN_REPORT	Where does the user drill down to for reports?	Recommend this field's omission
EMAIL	The user's email address	Recommend this field's omission
EMAIL_CHARSET	The character set of the email	Recommend this field's omission
EMAIL_FORMAT	Format for received email	Recommend this field's omission
EMAIL_ON	Switch to indicate whether a user's email is turned on or off	Recommend this fields omission
ENABLED_USER	Indicates whether the user is enabled or disabled	Recommend this field's omission
FAX	Fax number	Recommend this field's omission
FILE_ATTACH_CHARSET	The user's default character set for their attachments	Recommend this field's omission
FIRST_NAME	User's first name	Recommend this field's omission
HOME_TELEPHONE	Home telephone number	Recommend this field's omission
HTTP_CHARSET	Browser character set	Recommend this field's omission
JOB_TITLE	User's job title	Recommend this field's

		omission
LANGUAGE	Display language	Recommend this field's omission
LAST_ACCESS_DATE	Timestamp the user last accessed ExtraView	Recommend this field's omission
LAST_DATE_UPDATED	The date of the last update to the user's record	Recommend this field's omission
LAST_NAME	User's last name	Recommend this field's omission
LAST_UPDATED_BY_USER	The user ID of the person who last updated the record	Recommend this field's omission
LDAP_UPSERT_TIME	The last date / time the user's account was synchronized with a LDAP server	Recommend this field's omission
LOGIN_ID	The alternative user ID	Recommend this field's omission
LOGIN_MSG_DT	Not used	Recommend this field's omission
MS_OFFICE_CHARSET	Character set output to MS Office applications	Recommend this field's omission
NEW_PANELS	Selects whether user created a new workspace panel or uses the same panel for each new Quicklist report when drilling down from queries and reports	Recommend this fields omission
NOTIFY_ON_OWN_UPDATES	User notification when making an update	Recommend this field's omission
PAGER	Pager number	Recommend this field's omission
PANEL_AUTOSCROLL	User auto-scrolls to workspace panels when they click on them	Recommend this field's omission
PASSWORD_EXPIRY_DATE	Password expiry date	Recommend this field's omission
PASSWORD_INTERVAL	The interval between password expiries	Recommend this field's omission
PASSWORD2	The current, encrypted password of the user. All password changes use this field. There is a field named SECURITY_PASSWORD which contains the legacy password, if it has not been updated since version 7.0	Recommend this field's omission
POSTAL_CODE	Postal code or Zip code	Recommend this field's omission
RECORDS_PER_PAGE	The default number of records on each report output page	Recommend

			this field's omission
REGION	Locale region		Recommend this field's omission
REPORT_1_ID	Home page report 1		Recommend this field's omission
REPORT_2_ID	Home page report 2		Recommend this field's omission
REPORT_3_ID	Home page report 3		Recommend this field's omission
SCROLL_PANELS	Allows the user to scroll the workspace canvas or any workspace panel or element within that panel when the mouse pointer is over the respective area		Recommend this field's omission
SECURITY_PASSWORD	User's password. Note that since version 7.0, the field PASSWORD2 is used for the purpose of holding the encrypted password		Recommend this field's omission
SECURITY_USER_ID	The User's ID		
SOUNDEX_LAST_NAME	The soundex code for the user's last name		Recommend this field's omission
START_PAGE_ID	The ID of the start page the user is using		Recommend this field's omission
STATE	State		Recommend this field's omission
STYLESHEET	Size of text in browser		Recommend this field's omission
TIMEZONE	User's time zone		Recommend this field's omission
TWENTY_FOUR_HOUR_TIME	Display of times in 12 or 24 hour format		Recommend this field's omission
USER_FIELD_1 to USER_FIELD_10	The user defined fields		Recommend these fields omission
USER_ROLE	User's current role		Recommend this field's omission
VARIANT	Locale variant		Recommend this field's omission
WORK_TELEPHONE	Work telephone number		Recommend this field's omission
WORKSPACE_1_ID	The ID of the default workspace of the user		Recommend this field's

omission

Allowed Value Conflicts

Importing allowed values presents some challenges, when the data in the target database has changed significantly from the data in the source database. An entry in allowed values is unique for a given Title, Parent and Child value.

- For any given combination of a Parent and Child, only 0 or 1 entry can be set to be *Enabled*
- For any given Parent and Child there can be any number of entries set to *Disabled*
- For each allowed value, there can be entries in the Allowed Values database table with allowed values for that type

The points to observe during an import of allowed values are:

- Making sure that only one entry for a given Parent and Child is set to enabled. This means importing an enabled row requires a check that there is no other entry enabled. If there is another entry enabled for the given Parent and Child combination, then the import entry is set to disabled and imported
- Matching imported allowed value entries to existing allowed value entries. The Title, Parent and Child are used to find a matching row. If none is found, the imported entry is inserted. If a matching row is found, it is updated and the value lists are merged
- The allowed value lists for allowed value types are only added to; they are never reduced
- Given the import of an enabled allowed value type is not allowed if another allowed value type is enabled in the database, if the import wants to change which entry is enabled, it must first disable the currently enabled entry and then import the enabled entry from the import file. This implies that the export process should export its disabled entries before exporting its enabled entry. When imported they will then be in the correct order. The values list is merged in either case

		Imported Value of Allowed Value Type *Enabled*	
		Disabled	Enabled
Current Value of Allowed Value Type *Enabled*	Disabled	OK	OK only if enabled count = 0
	Enabled	OK – AVT will become disabled	OK

Importing List fields with Aliases

If an existing list field is not an alias of another list field, the import logic maintains this, and keeps the list field non-aliased in the target database. The import cannot make a non-aliased list field an aliased field during the import because it cannot guarantee that the lists are intended to be the same. For this same reason, the import will not change an aliased list field to be the aliase of a different list field. In normal operation, ExtraView checks these conditions and only allows valid operations, therefore the import must also make the same checks. In the context of this section, list fields include all fields with a display type of list, popup, tab, and radio button.

The import rules for the *Alias of* field values are as follows:

- If the list field entry being imported is not an aliased field, it is imported and the list remains non-aliased after import. The imported list values are merged for these fields
- If the list field entry being imported is an aliased field, it is imported only if the imported aliased field definition is the same in both the file being imported and in the target database. The imported list values are merged
- A non-aliased list cannot be aliased by the import process
- The values within an aliased list cannot be changed by the import process

		Imported Alias Field Name	
		None	*List A*
Current Alias Field Name	None	OK	No
	List A	OK	OK
	List B	OK	No

Error Logging During Import

A text log of errors is created on the server, as the XML updater executes. Each error entry will contain as much useful information as possible, including the table, row number, the data being used at the time, and the exception message, if any. This error log is echoed to the summary screen in the browser.

The error file is stored in the system temp directory under the name: <time_in_ms>_<user_id>_XML_ERRORS.txt

for example:

1026520070671_JEFF.SYKES_XML_ERRORS.txt

All error messages begin with

::XML_UPDATE_ERROR::

plus a new line, and end with ::END_XML_UPDATE_ERROR:: followed by two new line characters.

Here is an example of an error:

::XML_UPDATE_ERROR::
An error occurred updating row 125 of ITEM_MODULE with this data: {blah = blah, this = that, you = them}

Exception: java.sql.SQLException: ORA-00904: invalid column name
::END_XML_UPDATE_ERROR::

::XML_UPDATE_ERROR::
An error occurred updating row 126 of ITEM_MODULE with this data: {blah = blah, this = those, you = me} Exception:
java.sql.SQLException: ORA-00904: invalid column name
::END_XML_UPDATE_ERROR::

In order to be able to store multi-byte characters within the error file, note that the error file is not created as standard text. If you open the file in a standard editor, you will notice that what looking at most of the metadata in the file, every second character is null.

Handling rejected records during the import of metadata

As ExtraView processes the XML import file and fires the updating routines, if errors prevent an update are generated, the problematic XML nodes' data is captured, and used to generate a new XML file. This file is of the same structure as the original file, but only includes those items that could not be properly updated. If there are no errors generated during the upload process, this file is deleted at the end of the update process, as it is empty.

This file is saved under the system temp directory under the name:

<time_in_ms>_<user_id>_XML_REJECTS.xml

for example

1026520070671_JEFF.SYKES_XML_REJECTS.xml

Using Export and Import to Update Instances

The Metadata Export and Metadata Import procedures are designed to make it straightforward to update your production instance of ExtraView. The update may be required for a number of reasons –

- You are developing new functionality in the form of business areas, projects, layouts, logic, reports, etc. and do not want to do this in your production, live instance of ExtraView
- You have a new release of ExtraView software and want to upgrade your production system, after testing the new release with your application

It is important to have a repeatable, recoverable process to perform these tasks, and the following recommendations should give you a robust method to fulfill these tasks. The recommendations below can be simplified somewhat, according to your circumstances and the level of risk you are willing to carry. If you want to simplify these steps, you may move your development instance directly to the production instance, without going through the staging instance.

The instances

- **Production environment** – this is the server instance of ExtraView that is used for the live processing of your issues
- **Staging environment** – this is an intermediate server which will have a copy of production data and a copy of the development ExtraView system. No changes should be made to ExtraView metadata on this instance. It is purely used to check and test the functionality of ExtraView before applying the changes to the production environment
- **Development environment** – this is the instance of ExtraView where you will carry out all changes to the configuration before applying the changes to the staging environment

Step 1 – Create a development instance

- Use your database backup procedures to take a dump of the production database and create the database schema in a new development database
- If you are upgrading ExtraView and there is a database upgrade script to run, do this now against the development database
- Install the ExtraView application code, configuring it to connect to the new development database and start the application server
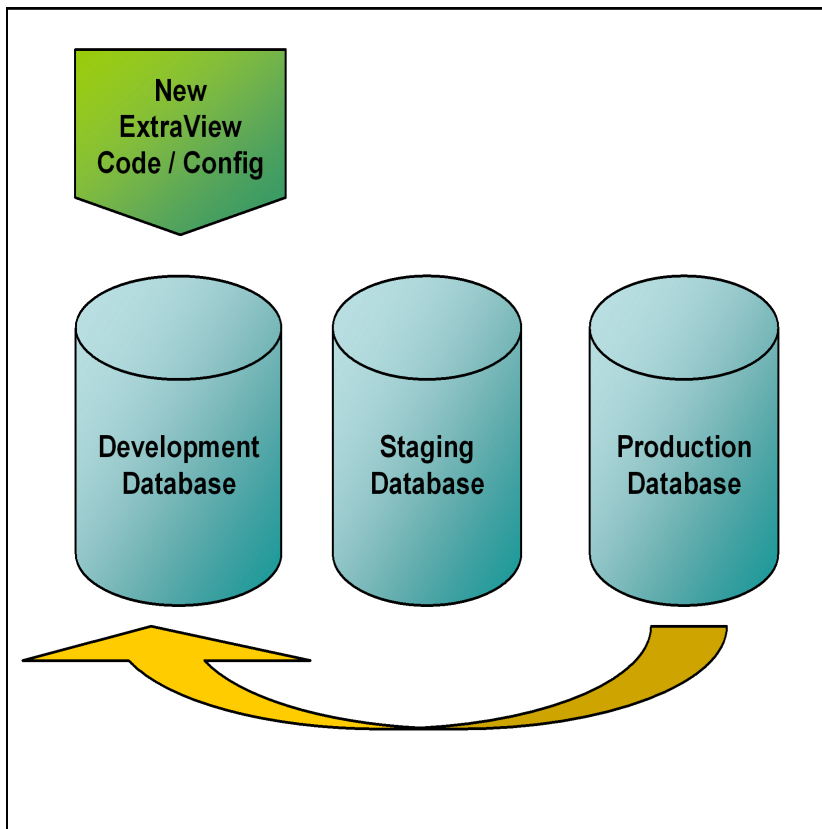
Step 2 – Perform the configuration work

Within the development environment, perform all you configuration and testing, to develop the completed system you intend to deploy

Step 3 – Move the development environment to the staging environment

The purpose of this step is to obtain a full working system on which you can perform all your quality assurance and final testing.

- Take another backup copy of the production database
- Create a new staging database, and import the production data into this new database
- If you are upgrading ExtraView and there is a database upgrade script to run, do this now against the staging database. **It is critically important that the version number of the ExtraView database in the staging environment be the identical version number you are about to import from the development environment**
- Use the Metadata Export feature to export the ExtraView metadata
- Use the Metadata Import feature to import the ExtraView metadata into the staging database
- If you are upgrading the ExtraView code, install this now and restart the application server
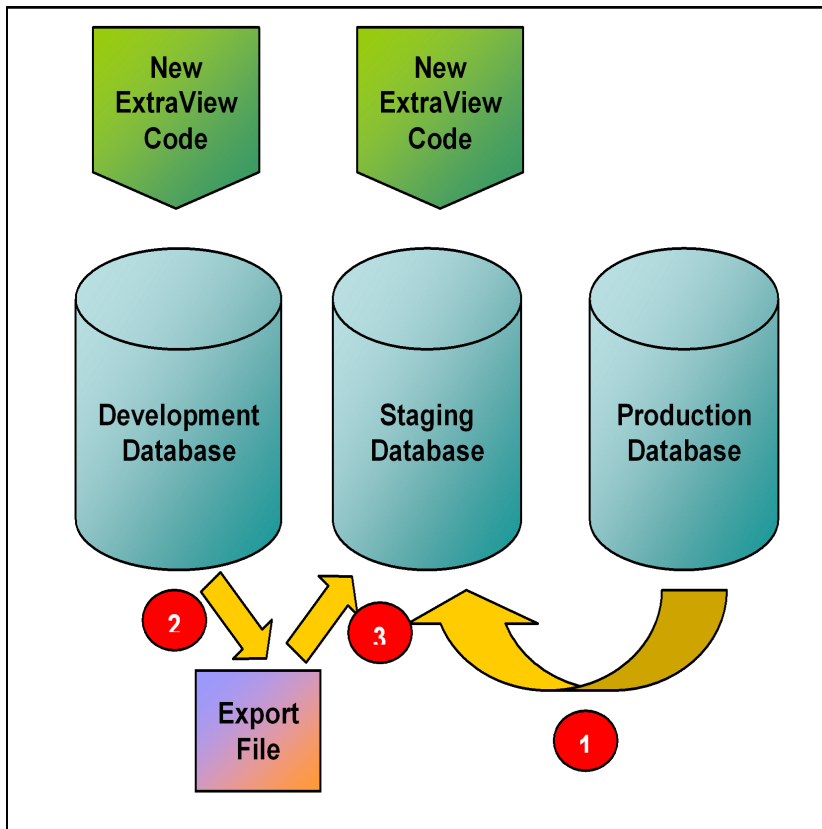
Moving the development instance to the staging instance

Step 4 – QA the entire ExtraView environment in the staging database

This is an essential part of the process to ensure that you gain the expected results, and that you can accurately predict the results you will obtain in the final production environment.

Note: You should not make any configuration changes to ExtraView here. If you find problems, you should repeat the process from step 1.

Step 5 – Upgrade the Production database

- Turn off the production database so users cannot access this during the upgrade process
- Take a backup of the production database
- If you are upgrading ExtraView and there is a database upgrade script to run, do this now against the production database. **It is critically important that the version number of the ExtraView database in the staging environment be the identical version number you are about to import from the development environment**
- Use the Metadata Import feature in the production database to import the export file that was created in Step 3
- If you are upgrading the ExtraView code, install this now and restart the application server
- With administrative access, check that the production environment is working correctly
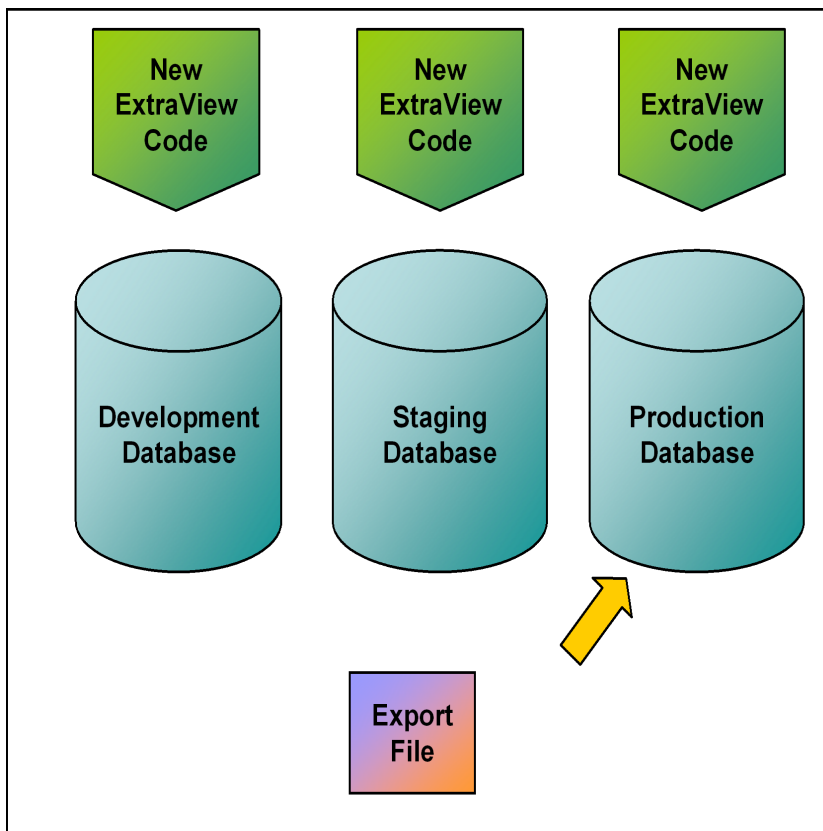
Updating the production instance

The process to upgrade your production ExtraView instance is now complete.

Item Data XML Import

Batch Mode

This function is driven from the **Import/Export** tab of the **Administration** menus, in the function named **Item Data XML Import**. When you access this function, you will see a screen similar to the following:

Batch interface to load an XML item file

Only a "merge" function is imported at this point in time. Future versions will support additional methods of importing item data. Use the Browse button to navigate to the XML file containing the item data to be imported. Then press the Upload XML file. Be

prepared to wait until the XML data file is uploaded to the server. This may take some time, depending upon its size.

Note: To save time on a large import, you may zip the input file and upload the zipped file. ExtraView will automatically unzip the uploaded file.
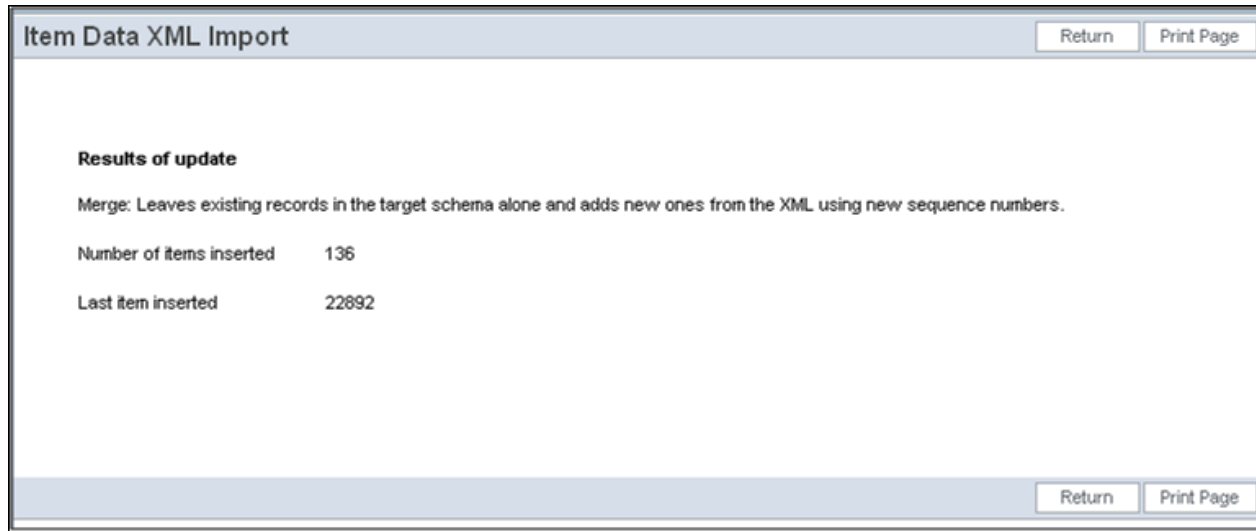
Results screen from a batch XML import

While the file is being processed, you will see a status bar informing you of progress. Once complete, you will see a screen similar to the one above. If there are any errors in the import file, you will see these on the results screen.

API Commands

These commands are also explained in the ExtraView CLI & API Guide.

An API action named xml_insert creates a new record or records in the ExtraView database from input formatted as XML. The input can be made as part of the HTTP data stream, or can be input from a file in XML format.

Syntax:

http://*www.myserver.com/evj/ExtraView*/ev_api.action?
user_id=*username*
&password=*password*
&statevar=insert_xml
&xml_file_name=*filename* | &xml_string=*xml_data*
&template_file=*file.html*

- Provide either the xml_file_name or xml_string, but not both. You provide xml_file_name if the input is from a file that exists at the time of the execution of the command. You provide xml_string, if the data for the insert is provided as part of the HTTP request. This string contains the XML data to be parsed.
- template_file is the name of the template to be used for return value string generation. Generally, this template file is stored on the server in the WEB-INF/user_templates directory. On normal completion of the operation, this template undergoes parameter substitution with the following variable names:

Tag	Explanation
__ID__	The item number of the last item inserted
__NUMBER_ITEMS_INSERTED__	The number of inserted items
__ITEM_TITLE__	The title of the ITEM_ID dictionary entry

See the section on Templates in the ExtraView CLI & API Guide, for a full explanation of how to create user templates

If no template file is requested, the command returns a completion message to the calling program via HTTP.

On error completion, the return string contains an error message substituted into the error.html user template in the format:

error-message "at line=xxx and column number=yyy"

where xxx and yyy are the values returned by the XML parser.

As an example, the following message may be returned:

```
"The end-tag for element type "ITEM" must end with a '>' delimiter at line=8 and
column number=9"
```

- Only one record should be inserted with the XML_STRING in one call to the API. When the input is in a file, there is no restriction to the number of records in a single operation.

Item Data Import Errors

Error Message	Explanation
Ambiguous Area title: *title*	The business area title in the XML is not unique within the database. Use its name instead.
Ambiguous Module title: *title*	The module with the title is not unique. Use the module name in this case to avoid conflicts
Ambiguous Project title: *title*	The project title in the XML is not unique within the database. Use its name instead.
Ambiguous Repeating record title: *title*	The repeating record with the title is not unique. Use the name in this case to avoid conflicts
Ambiguous UDF title: *title*	The UDF with the title is not unique. Use the UDF name in this case to avoid conflicts
Bad zip file or invalid XML tag at beginning of file	ExtraView cannot read the zip file with the XML data, or the XML tag at the beginning of the file is invalid
Cannot dereference list entry for ddname [*value*]	There is no list value corresponding to the *value* for the data dictionary name
End Attachment tag outside Attachment	The end attachment tags must be at the end of the attachment data
End item tag outside item	The end item tags must be at the end of the item data
End module tag outside module	The end module tags must be at the end of the module data
End *name* tag outside *name*	The end name tags must be at the end of the name data
End Repeating record tag outside Repeating record	The end repeating row tags must be at the end of the repeating row data
End UDF tag outside UDF	The end UDF tags must be at the end of the UDF data
Illegal NAME construct ignored	Provide valid name constructs
Invalid characters in the title value: [*value*]	The title value contains illegal characters
No ddentry for dereferenced name: *dd_name*	There is no data dictionary entry corresponding to *dd_name*
No Area with title: *title*	The business area with the *title* does not exist in the database
No Module with title: *title*	The module with the title does not exist
No product name for module: *name*	You must always provide the product_name field in the XML when adding modules
No product name for release: *name*	You must always provide the product_name field in the XML when adding repeating records which are dependent upon this field
No Project with title: *title*	The project with the *title* does not exist in the database
No Repeating record with title: *title*	The repeating record with the title does not exist

No such XML file: *file_name*	Indicates the input XML file does not exist
No UDF with title: *title*	The UDF with the title does not exist in the data dictionary
Nothing to convert from XML to DB; quitting	Input XML file appears to be empty
Unrecognized end element tag: *name*	The end tag for the name does not match the beginning tag for the name
Unrecognized start element tag: *tag_name*	The element tag name is not valid
WARNING: this source locale: [*locale*] is not available in target system	The DTD indicated that a locale is to be used that does not exist within the ExtraView installation
Validation exception: *exception*	General validation exception, as given in *exception*

File Import - Issue Data

The file import utility allows the administrator to upload a tab-delimited or comma-delimited file from their client computer, for the fields and values within this file to be mapped to ExtraView values, to validate the data and to finally import the data. Data can be uploaded as new issues, or can be used to update existing issues. Access to this feature is via the security permission key named CF_FILE_IMPORT. The import works as a separate process for each business area and project, therefore your import files will only target a single business area and project. For a record to be imported and inserted as new, it must exist on the appropriate *add* layout or one of its embedded layouts for the business area and project. For a record to be imported and update an existing record, it must exist on the appropriate *edit* layout or one of its embedded layouts for the business area and project.

Note: The Add / Update task must be configured and running for the file import utility to work. This is because the performance is optimized when the utility is running in the background. Further, this allows multiple copies of the utility to run at one time, further increasing the performance of the import.

Typically, the files to be imported will contain data from other systems that is to be inserted into ExtraView. Microsoft Excel is a good tool that can be used to prepare data in the appropriate format for the import. The process followed is:

- Create a template for the import and upload the file
- ExtraView automatically provides mappings for fields that are both in the uploaded file and within the ExtraView database
- ExtraView automatically provides mappings for field values that are both in the uploaded file and within the ExtraView database
- You can see the results of the mapping and alter the mapping of any field or field value
- You may then validate the uploaded data, ensuring the file is error free before performing the upload
- On the validation screen, there is an alternative to download the file that was uploaded. This is in addition to being able to observe the errors on the screen. If there were errors in the uploaded data found by the validation process, these errors will be shown within the file you can download. This simplifies the process of finding, fixing, re-uploading and re-validating the data before you perform the import
- Finally, you can perform the import and insert the records.

Note: ExtraView will ignore whether fields are or are not mandatory when importing data. It will also ignore whether fields have correct allowed value relationships when importing data. It will not ignore read-only fields and later in this section, you will find how to import fields that are read-only on your *edit* screen layout. However, when you edit any of these records in the web-interface or via the CLI, and attempt to update, the appropriate validations will be performed and you must correct any errors of these types at this time. Fields with default values defined in the data dictionary will have the default value created, unless there is a value provided in the import file.

Import Templates

When you use the File Import Utility, you create a template that may be reused as many times as needed. This aids the sometimes repetitive process of validating and correcting your data. You can set up the field and value mappings once, and upload your import file as often as is required, validating the data as you progress. This also aids the routine uploading and importing of data from other sources, as the same mappings can be used each time, without the need to set these up.

There are three import modes from which to select, when you create the import template:

1. **Insert Issues** - New issues are created from each record in the import file.
2. **Update Issues** - Existing issues are updated from each record in the import file. There must be an **ID** column within the import file, and this must contain the **ID** of the issue to be updated

3. **Update and Insert Issues** - New issues may be created and existing records may be updated. To facilitate this, there must be an **ID** column within the import file, and this must contain the **ID** of issues to be updated. If the field is left blank, then the record will be inserted. Note that the corresponding *add* layout for the business area and project is used for validation purposes, when inserting issues, and the *edit* layout is used when updating issues

Large Volume Imports

Special consideration may be made when you have tens of thousands or more records that must be imported in a short space of time. You may configure multiple import files and these can be imported in parallel. Please see the section on <u>large volume imports</u> for more information.

Import Strategies

Legacy Data Issues

Importing data from a legacy system is often problematic, for a number of reasons. These include –

- Data in legacy systems is rarely "clean". For example, you may have been required to type in entries that become list values in ExtraView. Typing data often leads to mistakes, making it difficult to map different entries. ExtraView allows you to map more than one input value to a list value
- You may not want to carry over all users from a legacy system to ExtraView. For example, you may not want to purchase an ExtraView license for an ex-employee who was the creator of an issue in ExtraView. You can either create a user account in ExtraView and deactivate the account, or you can map the name to a different person in ExtraView
- To take advantage of all the exciting features in ExtraView, you may be changing your workflow and data items that you capture in a significant way. Not all legacy data will have value in these circumstances, but you may want to import it for historic purposes. ExtraView allows you to do this
- If you want to import data that is read-only when in the mode of updating an issue, temporarily alter the field to be read-write, import your data, then reset the field to be read-only.
- One field of particular note when importing data from a legacy system is the ID field. ExtraView licensees may want to retain the legacy number. ExtraView must use its own sequence of numbers. To facilitate this requirement, a text field named ALT_ID is defined in the ExtraView data dictionary. Map your legacy field to this ExtraView field and retain it on the layouts while you make the transition. Eventually, you may remove this field from the ExtraView layout.

Fields with Special Treatment Upon Import

When a record is inserted into ExtraView or updated by ExtraView there are several fields which are maintained automatically, and cannot be inserted or updated via the web-based interface, the API or the CLI. However, when importing legacy data, you may need to import values from these fields. ExtraView supports this. However, the fields must be on the add screen layout, and must have write access to accommodate the import utility. This table specifies the treatment of these fields:

Data Dictionary Field Name	Treatment
DATE_CREATED	If this exists in the import file and is mapped, the value provided will be inserted, else the current date and time will be used
DATE_CLOSED	If this exists in the import file and is mapped, and the STATUS of the issue equals the STATUS defined in the behavior setting named STATUS_CLOSED_NAME, the value provided will be inserted, else no value will be inserted
DATE_LAST_STATUS_CHANGE	If this exists in the import file and is mapped, the value provided will be inserted, else the current date and time will be used
ID	The ID field is always created by ExtraView, without exception. It is provided a sequential value when the issue is first created. When you are using the file import utility to update data, this is normally used as the key field upon which you map the record being imported to the existing issue within ExtraView. If your installation is configured to use ALT_ID as the primary means of identifying an issue, then you can expect ExtraView to create this from records within the import file, and you will use this as the key field upon which to map issues being updated
LAST_CHANGE_USER	If this exists in the import file and is mapped, the value provided will be inserted, else the current user ID will be used

ORIGINATOR	If this exists in the import file and is mapped, the value provided will be inserted, else the current user ID will be used
TIMESTAMP	This field cannot be mapped when importing data. If you want to preserve the timestamp, or an issue's last date updated, it is suggested you create a user defined field and import the data into this field.

Business Areas and Projects

The import utility will only import data into a single business area and project. It is not possible to map and import data into multiple business areas and projects, or into the main item structure and the repeating record structure at the same time with the import facility. Other methods exist within ExtraView to handle complex structures; the utility described here is intended for the import of simple structures (this method handles about 80% of requirements), and it is quick and simple to use. For complex data migration purposes, consider the XML import utility or the CLI evimport function. The strategy to import data into multiple areas and projects is to segment your data by area and project in your import file, and to import each one separately.

Read-only Fields

Most of the time, the data you import will be writeable through the add screen layout that you are using. The layout selected for the import is defined by:

- The business area defined in the import template (if you are using business areas in your installation)
- The project defined in the import template (if you are using business areas and projects in your installation)
- The current role of the user performing the import. Normally this is an administrative role, such as admin.

Import Strategy

If you have a straightforward file to import, simply use the add screen layout you have created, and create an import file to upload that reflects its contents. However, if your administrative role does not have all the fields on the add screen and they are not all writeable, or you have a relatively complex system with multiple Business Areas and Projects, the following strategy is good to follow. This is especially true for companies needing to import data from other sophisticated tracking systems which contain a good deal of metadata about each issue:

- Create a new user role named *import* or similar
- For this new user role create a new add screen layout. The speediest way of doing this is to choose the an *add* or *edit* screen layout for a role with a layout close to what you need, to then alter the user role on the *edit* screen for the layout to the new user role, then save the layout as the *add* screen for this role
- Make sure that the fields are writeable on this new layout
- Alter your current role to the new user role
- Perform the import

If you have a very large volume of data to import, ExtraView supports some advanced features which allow you to import many files in parallel. These features are described [here](#).

The upload file

Sample upload file in Excel

The top arrow in the diagram points to row 1 of the spreadsheet. Here you see that each column contains the field names that are part of the input. The remaining rows of the spreadsheet contain the data that corresponds to the field names. Note:

- Fields in the import file may have different names to the fields in ExtraView. However, if they have the same name (when compared on a case insensitive basis), ExtraView will automatically map the field from the import file to its own data dictionary
- ExtraView will attempt to map, automatically, values that are in fields that are of display type List, Tab, Popup and User. This is on a case insensitive basis. For example, a status in the import file named *open* will be automatically mapped to an ExtraView status value of *Open*. If the value cannot be mapped by ExtraView, you must provide a mapping before the record can be uploaded
- Values within the import file, in fields of display type List, Tab, Popup and User must exist in ExtraView, before the data can be imported. ExtraView will not automatically create new values for you, because of the inherent dangers of propagating bad data from a legacy system to ExtraView. However, ExtraView will allow you to map from a non-existent value to a legitimate value.
- If you are importing date fields, then the format of date is important and must match the format of date for the user who is performing the import. If this did not happen, then ExtraView may encounter ambiguous dates as there will be no certainty of the date format in the import file. To achieve this, the user should enter their personal options screen and select a date format from those shown that matches the format in the import file. If no format in the list matches the format in the import file, then **Custom date mask** should be selected, and the date format should be entered into the field provided for the custom format. See Appendix A of this guide for the specifications of custom date masks.
- If a default value is set in the data dictionary for any field, this value will be inserted into all records, if there is no value in the import file.

Once the spreadsheet is prepared, use the File, Save As menu within Excel, and choose either Text (Tab Delimited) or CSV (Comma Delimited) as the type.

Note: There is a limit of 500 Mbytes and 300 fields per record to the size of files that can be imported. If you have a very large amount of data to import, it is often more efficient to break it down into several import files, and to process these individually.

Fields

Multi-Value Fields

Fields to be imported that are of a multi-value type require special preparation within the import file. All the values to be imported must exist in between the delimiters of the file (tabs or commas), yet ExtraView must be able to distinguish between the different values. This is accomplished by using the system-wide delimiter in the behavior setting named DEFAULT_TEXT_REPORT_DELIMITER. This has a default value of ":". Use this delimiter to distinguish values. For example, you may have a list with days of the week (*Sun, Mon, Tue, Wed, Thu, Fri, Sat*). If you need to indicate that the valid values of data within the record are *Tue, Fri* and *Sat*, you would use an entry of *Tue:Fri:Sat* on your spreadsheet:
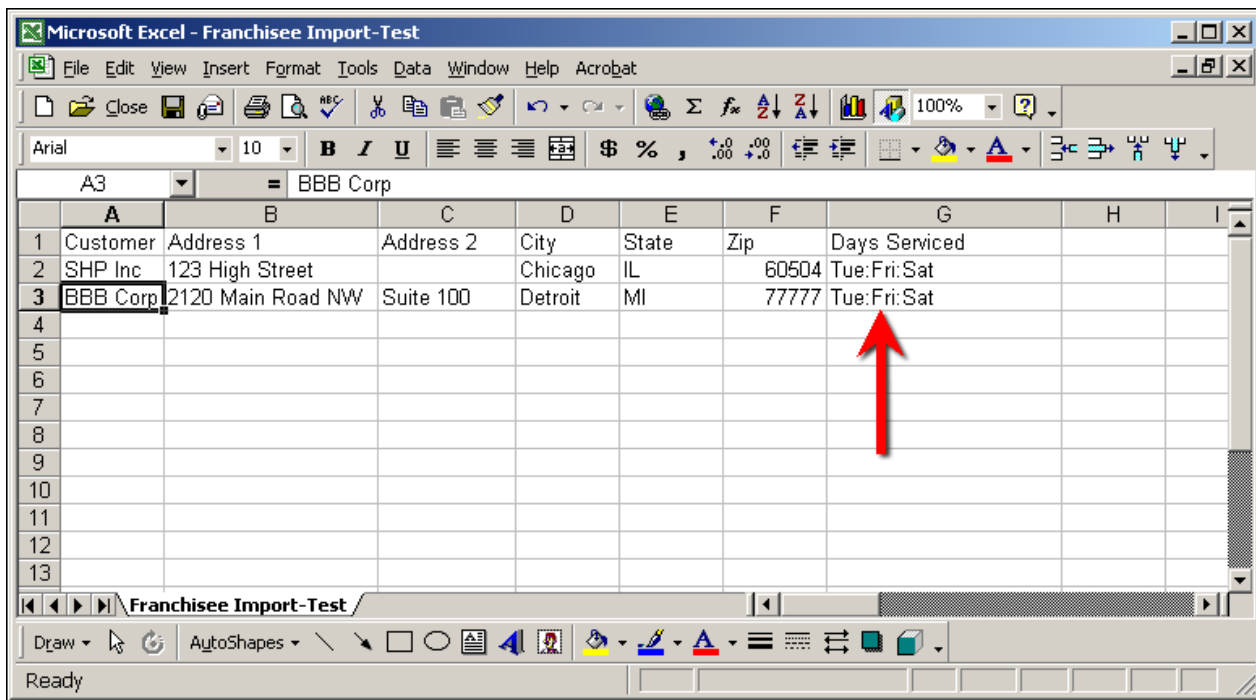
Importing multi-valued fields

Text Fields

Fields that are to be mapped to Text Area, Log Area and Print Text display types have special treatment, to allow you to import values that contain a return character (ASCII 13). This character is normally used by many applications to indicate the end of data of a row. This implies that the data following the return character is a new record. Unfortunately, when you export a comma or tab delimited file from Excel or other applications, these return characters are treated as an end of record. To counteract this in a way acceptable to Excel and the import facility in ExtraView:

1. Ensure new lines separated by a return character appear inside quoted strings (double quotes only). They will be transferred over into the record as is. In detail, carriage returns are discarded by ExtraView because they are non-existent within Unix files, and browsers behave correctly when they only receive a linefeed character
2. Tab characters may be inserted into the record inside quotes (if tab-delimited) or anywhere (if comma-delimited). They will be carried over to the record as is.
3. To get a double quote into a field, you need double it. Thus, "" represents one quote character. However, you can also put a double quote into a field if it is NOT the first non-whitespace character in a word. Thus, abc"def will carry across without modification, but "abcdef" will require modification.

User Name Fields

ExtraView expects to find user name in the import file in the same format as specified by the behavior setting USERNAME_DISPLAY. For example, this may be FIRST, LAST or ID. We recommend that for large imports where there may be more than one person with the same first and last name, that you import data using the ID setting. This will ensure that all users can be identified uniquely.

Maximum Field Sizes

Fields that are imported are subject to the maximum size constraints of their field types, and are as follows –

Field / Field Type	Maximum size
List field titles (loaded as metadata)	100 bytes
User ID (loaded as metadata)	30 bytes
User First Name (loaded as metadata)	128 bytes

User Last Name (loaded as metadata)	128 bytes
SHORT_DESCR	255 bytes
ALT_ID	128 bytes
UDF text field display types	256 bytes
UDF text area, log area, print text display types	Approx 10MBytes. Note that if you are using Microsoft Excel to prepare your input file, there is a 31k limit on the size of a cell.

Mapping your data

Within ExtraView administration, enter the File Import Utility. The following screen appears:

File Import Utility

Create a new import template and the following screen appears. Simply provide a name for the template, and select the file that is to be uploaded. Note that if your installation uses Business Areas and Projects, select lists for these will appear on the screen. Once a template has been created for a specific Business Area and Project, these cannot be altered. Note that you select whether the template is to be used to create new records, or to update existing records. You achieve this using the **Select Mode** prompt.

Creating the import template

Now you will see the main import screen, as follows:

Business Area　Bugs
Project　Bugs Data
Title　Sample import of bugs data　[Localize]　← **Template details**　　**Funtion buttons**

Upload a different file　[_____]　[Browse...]

Sample of uploaded file　Category,Originator,Date Created,Status,Resolution,Priority,Severity,Assigned To,Changed by,Title,Product,Description
Hardware,CSR,12/17/2003,Unassigned,,P 1,High,Developer,CSR,Metal mounting bracket is too short,Tracker,The bracket is n...　← **Sample data**
Software,CSR,12/17/2003,New,,P 3,Low,Developer,Developer,Turn left at the next Exit,Tracker Enterprise...
Documentation,CSR,12/18/2003,Fixed,Fixed,P 2,Medium,Test,CSR,Build it and they will come,Tracker Lite,You can leave it to N...

Tab-separated fields　☐

Field Mappings　For each import file field, choose an ExtraView field to map your data into. ExtraView will map the values within a field, if the same value exists in both the import file and ExtraView. To provide alternative mappings, click on the Map button. Choose *None* if you do not want to map a field in the import file. Click on the Refresh mappings button to see the results of your mapping.

Import file field　ExtraView field

[1] Category　　　Category(CATEGORY)　[Map]
[2] Originator　　Originator(ORIGINATOR)　[Map]
[3] Date Created　* None *　[Map]
[4] Status　　　　* None *　[Map]
[5] Resolution　　* None *　[Map]　　　← **Field mappings**
[6] Priority　　　Priority(PRIORITY)　[Map]
[7] Severity　　　Severity(SEVERITY_LEVEL)　[Map]
[8] Assigned To　Assigned To(ASSIGNED_TO)　[Map]
[9] Changed by　* None *　[Map]
[10] Title　　　　Title(SHORT_DESCR)　[Map]
[11] Product　　　Product(PRODUCT_NAME)　[Map]
[12] Description　Description(DESCRIPTION)　[Map]

Mapping Results　This displays the results of mapping your data and how it will appear in ExtraView. The top field name in the table is the field in your import file, the bottom field is the ExtraView field. Click on any header, if you want to map list values from your input file, to different values in ExtraView. Press the Refresh button after changing any mapping.

Field values will be inserted into ExtraView　　Field values will not be inserted into ExtraView

Row Number	Category > Category(CATEGORY)	Originator > Originator(ORIGINATOR)	Date Created	Status	Resolution	Priority > Priority(PRIORITY)	Severity > Severity(SEVERITY_LEVEL)	Assigned To > Assigned To(ASSIGNED_TO)	Changed by	Title > Title(SHORT_DESC
1	Hardware	CSR	12/17/2003	Unassigned		P 1	High	Developer	CSR	Metal mounting bracket is too shor
2	Software	CSR	12/17/2003	New		P 3	Low	Developer	Developer	Turn left at the nex Exit
3	Documentation	CSR	12/18/2003	Fixed	Fixed	P 2	Medium	Test		Build it and they w come
4	Hardware	CSR	12/19/2003	Closed	Implemented	P 4	High	Developer	Developer	Have wings will fly that's what they s
5	Documentation	CSR	12/20/2003	Duplicate		P 3	Medium	Test	CSR	A single swalow does not a spring make
6	Software	CSR	12/20/2003	Rejected	Not Found	P 2	Medium	Developer	CSR	Save it to a file wit a good name
7	Hardware	CSR	12/20/2003	Unassigned		P 3	High	Developer	Test	Metal mounting bracket is too shor
8	Software	CSR	12/21/2003	New		P 2	Medium	Test	CSR	Turn left at the nex Exit
9	Documentation	CSR	12/21/2003	Closed	Fixed	P 1	Low	Developer	Developer	Build it and they w come
10	Software	CSR	12/21/2003	Closed	Implemented	P 3	Low	Test	CSR	Save it to a file wit a good name
11	Software	CSR	12/22/2003	Fixed	Fixed	P 2	High	Test	Developer	Turn left at the nex Exit
12	Hardware	CSR	12/22/2003	Closed	Implemented	P 4	Medium	Test	Test	Have wings will fly that's what they s
13	Documentation	CSR	12/27/2003	Duplicate		P 3	Medium	Developer	Test	A single swalow does not a spring make
14	Software	CSR	12/27/2003	Closed	Fixed	P 3	High	Test	Developer	Build it and they w come
15	Hardware	CSR	12/27/2003	Fixed	Fixed	P 2	High	Test	Test	Have wings will fly that's what they s
16	Documentation	CSR	12/27/2003	Rejected	Not Found	P 3	Medium	Developer	Developer	Metal mounting bracket is too shor
17	Software	CSR	12/27/2003	Fixed	Fixed	P 4	Medium	Test	CSR	Turn left at the nex Exit
18	Documentation	CSR	12/28/2003	Closed	Implemented	P 3	High	Developer	Developer	Save it to a file wit a good name
19	Hardware	CSR	12/28/2003	New		P 2	Medium	Developer	CSR	Turn left at the nex Exit
20	Hardware	CSR	12/28/2003	Fixed	Fixed	P 3	Medium	Test	Developer	Save it to a file wit a good name

← **Results of mapping** (pointing at Severity column, row 3)
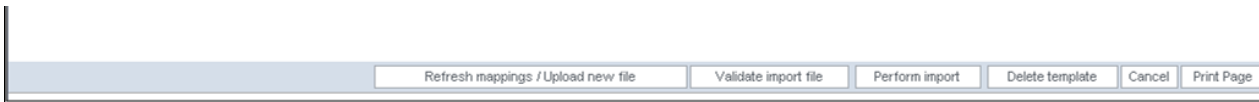
Import screen

Function buttons – these perform the following functions:

- *Refresh mappings / Upload new file* – if you alter the mapping of any field in the field mappings section, or use the Map button to alter a value mapping, or upload a new file, pressing this button will refresh the screen, redisplaying the results section
- *Validate import file* – this function checks each row of the import file and displays any errors in the input file. These errors should be corrected in the source file and the file uploaded again before performing the import. This can be done without leaving the import screen. You may validate the import file, correct errors within the file and re-upload the file as often as you need until the import file is error free
- *Perform import* – once the mappings are complete and the data validated, this button will perform a final validation and import the data into ExtraView. ExtraView will create new issues for each row of the uploaded, including new ID's
- *Delete template* – this deletes the current template and all mappings
- *Cancel* – this returns you to the initial file import screen with the list of templates.

Template details – shows the title of the template and allows you to upload another file into the template. Note that if the installation utilizes Business Areas and Projects, these will be shown for the current import template.

Sample data – this section shows a sample of data from the file that has been uploaded. You cannot edit the data within this text box, but must return to your source data to make any amendments.

Field mappings – Here you will see a fixed field name for each field header in the import file and a select list offering all the available fields within ExtraView to which you can map that column. To be a valid field for the import, the field must exist on the add screen layout or one of its embedded layouts for the appropriate business area and project, if these are in use and for the current role of the user performing the import. The field must have its security permissions set so the field can be updated. Further, you cannot map to the ID field. The ID field is maintained by ExtraView, and no value can ever be assigned to this from an external source. See later in this section on how to deal with read-only fields on your add screen layout. If you are trying to preserve the key identifying field from a previous system, the usual practice is to map the record identifier to the ALT_ID field.

ExtraView will attempt to map the field in the import file to a field on the *edit* screen layout, by comparing the field name in the import file to the data dictionary titles of the fields on the *edit* screen layout. This comparison is case insensitive. When a match is found, ExtraView sets the ExtraView field as the selected field for the mapping. You can override this selection, and you can provide your own selection for any field. This includes mapping a field to * None *, thereby skipping it during the import.

In a similar way that fields are mapped, ExtraView will attempt to map the values within any field that has a display type list, tab, or popup.. User fields cannot be mapped at this point. Performing a validation will show which field values are not mapped. To map the values within a field, press the Map button to the right of the field. The screen that appears will be similar to the following:
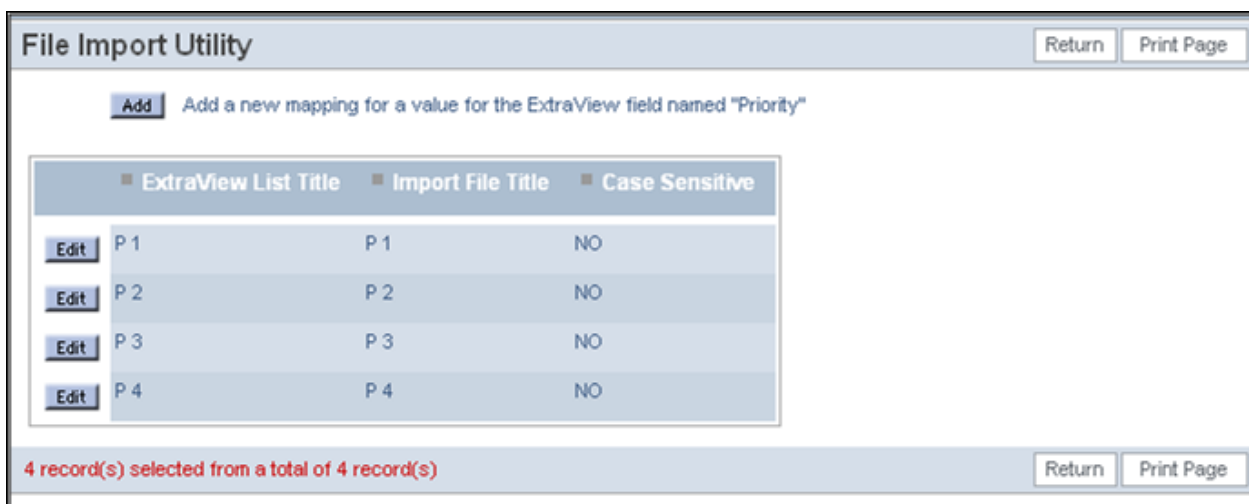
Mapping field values

You may press **Add** to create a new mapping, or you can edit an existing mapping. Note that you have a choice as to whether you want the mapping to be case sensitive or case insensitive. One point of note is that you can map fields with null values to a specific

value in ExtraView, and you can map values to different values than set as default. This gives you flexibility in altering the data that you are importing.

Note: Field mappings are retained within your template. This means that if you set up a number of mappings and want to re-use these with a different import file, you simply need to re-use the same template.

Results of mapping – This section of the screen shows which values will be inserted on which records, when you perform the import. The color-coding shows the fields that will be skipped when importing the data and shows the specific values that will be inserted into specific fields in the ExtraView database.

The Business Area and Project

If you provide the Business Area or the Project field in your input file, then you must provide both of these.

If you do not include these, then the File Upload Utility will use those specified in the template for the upload.

Updating Existing Issues

The file import utility has the capability to upload a file and to then update existing issues within ExtraView. To achieve this, a match must be made of a key field within the import file, with an existing key field within ExtraView. The only two key fields that can be used are either the ID field, or the ALT_ID field. Further, as ExtraView does not impose a unique value limitation on the ALT_ID field, the user must ensure that the values are unique. In practice this is usually achieved with the ALT_ID field by using user custom code.

Importing Repeating Row Data

If your ExtraView installation is configured to use repeating rows within the defined process, special consideration must be given in preparing the input file, to describe the repeating row fields, and to provide the data for the repeating rows. This diagram shows the structure if there is a single repeating row within each issue:
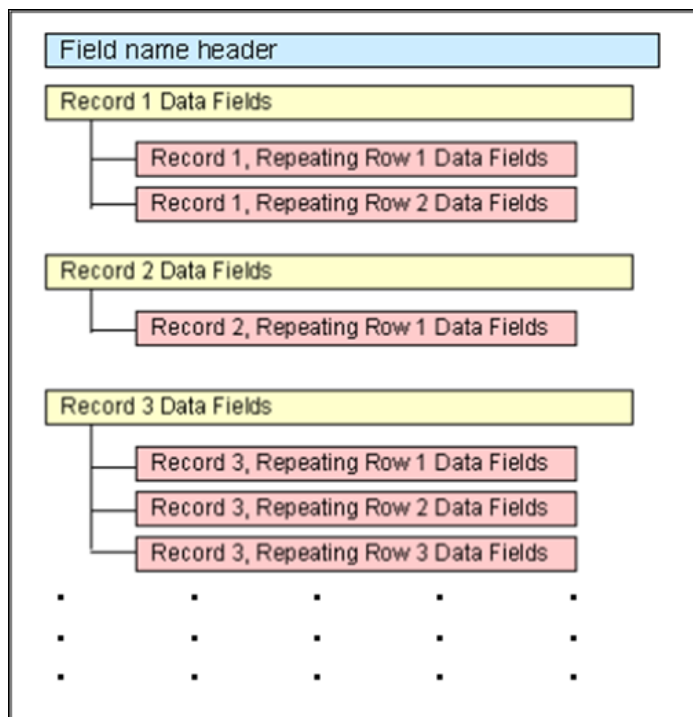
Structure of the import file

Within the import file that you create, this is reflected as shown in this screenshot of Excel, being used to prepare a comma separated value file:

Import file with repeating row data

Note the presence of the first column with the title of **Issue Type.** A value of 0 indicates the row contains the issue field data. A value of 1 indicates that the row contains data for a repeating row. The 1 value is used for the built-in RELEASE repeating row. You will provide a different (arbitrary) number for repeating row layout types that you have created.

The red arrows point to the field titles, first for the issue itself, and then for the repeating row field titles. The blue arrows show how the repeating row data values are then inserted into the spreadsheet as rows following the issue. There may be any number of repeating rows inserted for any issue.

Note these differences when preparing a file for import that contains repeating rows compared to a file that only has issue data:

- **Header Row**
  - The header row with the field names must begin in the first column with the value **Item Type** or **ITEM_TYPE_ID**. **Item Type** is the title and **ITEM_TYPE_ID** is the name. These terms are synonymous for the purpose of preparing the input file and can be used interchangeably
  - You place each of the field names (or titles) to be imported on the first row of the spreadsheet. First, you will have all the field names / titles for the main issue data in the order that the data will appear within the data rows
  - Next you will have all the field names / titles for each of the fields within the repeating row, in the same order they will appear within the data rows of the spreadsheet. Placing these once on the header row negates the need to repeat these for each series of repeating rows within the data rows of the spreadsheet
- **Data Rows**
  - Each row in the data must begin with the type of data on the row. The valid entries are either:
    - a **0** to represent **Issue Item** rows or
    - an ITEM_GROUP_TYPE_ID or its **File Marker,** to represent **Repeating Row Items**. See the next section on this page for an explanation of the **File Marker**
  - You can see from the above example that these all appear with the header row of the column being **Item Type**. An ITEM_GROUP_TYPE_ID value of **1** represents the built-in **RELEASE** repeating row type
  - The first data row of each issue will be an **Issue Item** row with a value of **0**, followed by its set of repeating rows. The remaining columns of the first data row will be the field values for the main issue fields. Each of the field values will thus appear within the spreadsheet column whose header row is the field name or field title to the field value
  - You may now create any number of rows with the **Repeating Row Item Group** type of an ITEM_GROUP_TYPE or one of its F**ile Markers**. For each row that is of a repeating row type, a new repeating record will be created within the

main issue. You begin the value data for each repeating row in column B of the spreadsheet, and place the values in the same order as the field names / titles were entered into the Header Row
   - For each issue you are inserting, with its set of repeating rows, repeat the steps in this section starting in the next blank row of the spreasheet.

Repeating row data stored within ExtraView issues may or may not have a unique identifier for each row of data. This is controlled with the behavior setting named ENFORCE_UNIQUE_RELEASES. When this is set to YES, you must:

- Map one field in the repeating row data of ITEM_GROUP_TYPE_ID **1** to be imported to the ExtraView field named RELEASE_FOUND
- Have a unique value of the field being mapped to RELEASE_FOUND for each row in the import file, within each issue
- Have write permission to the RELEASE_FOUND field

Importing Files Which Contain Data for Multiple Repeating Row Types

The preceding information relates to the importing of data which contains a single repeating row **Item Type**. ExtraView can import files which contain multiple repeating rows within a single file, using an extension of the above methodology.

You adjust the spreadsheet format for each repeating row type you wish to import by simply adding the field titles for each repeating row type in the header row, and then creating a data row for each repeating row type. The **Issue Type** column provides the indicator as to the repeating row to which the data belongs.

Each repeating row in an ExtraView database has an internal identifier which is not readily seen or displayed to the administrator creating the import file, therefore an intermediate *File Marker* is used to provide the connection between the field data within the import file and the data being stored within the ExtraView database. Follow this procedure to prepare your import file:

- **Header Row**
   - Follow the instructions in the preceding section, but simply continue the entries in the header row with the names / titles of the fields for each repeating row type within the data to be imported. Again, these should appear in the order that the value data will appear within the spreadsheet
- **Data Rows**
   - You arbitrarily select a number for each repeating row type that is to be uploaded. **0** is always the number to be used for the **Issue Item** rows of data. 1 is always used for the inbuilt RELEASE releating row layout type
   - The remaining rows will have their arbitrary number in column A of the spreadsheet. Obviously, each repeating row of a specific type should use the same arbitrary number within the input data values
   - You simply use the arbitrary number to identify which repeating row fields are to be associated with the data on the rows within the spreadsheet; these numbers can be mapped to the internal ITEM_GROUP_TYPE numbers representing repeating row types
   - When you see the mapping results within the File Import screen, you will see a connector field for each repeating row type. This has the title **File Marker.** You see the fields for each repeating row type within the mapping results grid on the screen, and you place the number you used within the spreadsheet you uploaded, within the **File Marker** field to identify which data rows in the spreadsheet belong to which fields in the header row.
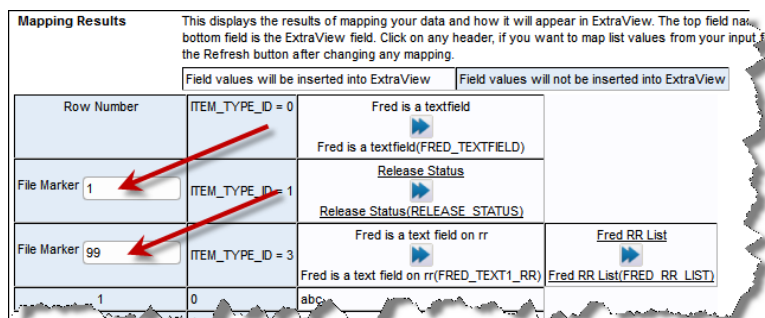
After uploading the spreadsheet with the multiple repeating row types, performing the mapping, and validating the data, you proceed with the file import.

Importing Image & Document Fields

You may import both documents and images into their respective field types via the File Import utility. This page describes the way in which you prepare a structure on the file system of the server where you will place the files to be loaded into the ExtraView database. This structure must be accessible by the ExtraView application server. The image and document files will be loaded from

there into ExtraView - either into the database or onto a part of the file system maintained by ExtraView, dependent on the value of the behavior setting ATTACHMENT_REPOSITORY_OPT.

The fields to be imported into your database are specified with either the first row of the imported file or through the mapping interface described in the previous section. If an image or document display type field is defined as an imported field, then the contents of that column in each row is a filename that points to a file accessible to the ExtraView application server. The file is used to populate the image or document field contents in the target issue.

Image and document fields have the following metadata fields associated with them:

- File name
- Alternate text
- Character set
- Created by user
- Last updated by user
- MIME type

These metadata fields may be set from the row data that updates the image or document field.

To set the metadata field, the associated dictionary fields, named **EVFI_xxx** are used to define a column where the metadata values exist. There are 12 dictionary entries that represent the image/document metadata: 6 for item fields, and 6 for repeating row fields - those ending in **_RR**:

- File name: EVFI_FILE_NAME[_RR]
- Alternate text: EVFI_ALT_TEXT[_RR]
- Character set: EVFI_CHARSET[_RR]
- Created by user: EVFI_CREATED_BY_USER[_RR]
- Last updated by user: EVFI_LAST_UPDATED_BY_USER[_RR]
- MIME type: EVFI_MIME_TYPE[_RR]

These field names may be chosen in the File Import Utility to indicate where the metadata values in the row(s) reside. Note that only one set of metadata per row may be specified, and hence importing more than one image/document per row will not permit metadata values unique to the different fields.

Example

Header record (on a single row of the import file)

Description, FRED_MULTI, FRED_IMAGE, EVFI_ALT_TEXT, EVFI_CHARSET, EVFI_CREATED_BY_USER, EVFI_LAST_UPDATED_BY_USER, EVFI_FILE_NAME, EVFI_MIME_TYPE

Data record (on a single row of the import file)

freddie description,frd2, C:\Users\Public\Pictures\Sample Pictures\desert.jpg, "Alternate Description", "Shift-JIS", gary.rather, gary.rather, theDesert.jpg, image/jpeg

Note: The reader will note that the header record contains data dictionary names instead of titles to illustrate the example; in a true import, the header record should contain dd titles to assure correct mapping.

In this example, the metadata fields will populate the image field "FRED_IMAGE" when the issue is created.

Default Metadata Values

When metadata is not specified in the field mapping for an IMAGE/DOCUMENT field, the following defaults are applied:

- File name: the file name portion of the full path name of the input image/document file
- Alternate text: the file name
- Character set: UTF-8
- Created by user: the user running the import operation
- Last updated by user: the user running the import operation
- MIME type: computed from the file extension of the file name, using the ExtraView internal mapping

Repeating Rows

Images and documents fields may be imported to repeating rows using the same type of mapping described in the Importing Repeating Row Data page. In this case, there must be repeating row values to set the metadata - this is the purpose of the

EVFI_xxx_RR fields. These metadata fields must be used to set the metadata for repeating row image/documents.

Update Option

Images and documents may be imported using either the INSERT option or the UPDATE option of the File Import utility.

Validation Option

The validation operation of the File Import function performs the following tests on imported images and documents:

- The file's existence and readability
- If an Image field, whether the file contains an image (a thumbnail can be created)
- For metadata
    - the MIME type must be recognized as a valid type
    - the CHARSET must be a valid recognized character set name

The same validation tests are done on the import operation. Validation errors will cause the import of that row to fail. Other rows are not affected, unless they also have errors.

Importing Attachments

You may import file attachments via the File Import utility. This page describes the way in which you prepare a structure on the file system of the server where you will place the files to be loaded into the ExtraView database. This structure must be accessible by the ExtraView application server. The image and document files will be loaded from there into ExtraView - either into the database or onto a part of the file system maintained by ExtraView, dependent on the value of the behavior setting ATTACHMENT_REPOSITORY_OPT.

Attachment metadata may be specified using similar metadata entries as for importing image and document fields. The attachments are specified in the import file by using the data dictionary name **ATTACHMENT** as a mapped field, and defining the file name as the value for that column. The remaining metadata entries that can be used are optional, and are defined as follows:

- EVFI_ALT_TEXT - the description of the attachment, with a default of the file name
- EVFI_CHARSET - the character set used for displaying the attachment, with a default of UTF-8
- EVFI_CREATED_BY_USER - the User ID of the user creating the attachment. The default is the User ID of the user running the import
- EVFI_UPDATED_BY_USER - the last updated user of the attachment, with a default of the User ID of the user running the import
- EVFI_FILE_NAME - the file name of the attachment, with the default of the actual file name
- EVFI_MIME_TYPE - the mime type of the attachment. The default is computed by ExtraView from the file's extension

These metadata fields may be set from the row data that describes the attachment to be imported.

To set the metadata field, the associated dictionary fields, named **EVFI_xxx** are used to define a column where the metadata values exist.

These field names may be chosen in the File Import Utility to indicate where the metadata values in the row(s) reside. Note that only one set of metadata per row may be specified, and hence importing more than one image/document per row will not permit metadata values unique to the different fields. Import document and image fields is similar to importing file attachments. You may view the [example](#) on that page as a model on how to use this feature.

Importing Multiple Attachments for a Single Issue

You may import multiple attachments for a single issue, by delimiting the attachment metadata within the appropriate fields. The delimiter used is defined by the behavior setting named DEFAULT_TEXT_REPORT_DELIMITER. Normally, this is a colon character. For example, the EVFI_FILE_NAME field for an issue may contain:

file_name1:file_name2:file_name3

This implies three separate attachments are going to be uploaded. Similarly, use delimited text in fields such as EVFI_ALT_TEXT to provide separate descriptions for each of the attachments.

Update Option

Attachments may be imported using either the INSERT option or the UPDATE option of the File Import utility.

Validation Option

The validation operation of the File Import function performs the following tests on imported images and documents:

- The file's existence and readability
- If an Image field, whether the file contains an image, so that a thumbnail can be created
- For metadata
  - the MIME type must be recognized as a valid type
  - the CHARSET must be a valid recognized character set name

The same validation tests are done on the import operation. Validation errors will cause the import of that row to fail. Other rows are not affected, unless they also have errors.

Large Volume Imports

Overview

Using the file import utility for the bulk import of tens of thousands (or more) issues currently requires a large investment of time due primarily to the sequential processing of the records being imported. ExtraView supports parallelism of the file import process by spawning multiple add/update requests to be processed simultaneously.

Current File Import Processing

There are two components to the overall processing. The File Import Utility is invoked from the Administration interface. This utility is responsible for:

- Interacting with the user to set up and performing the file import
- Uploading the file to be imported and saving it internally
- Creating the mapping of fields and rows to item sub-objects
- Starting the background File Import Worker task
- Reporting on the progress of the File Import process
- Formatting, displaying and downloading errors and results to the user.

The File Import Worker is the second component and is responsible for:

- Reading the internal file
- Performing the necessary field and row mapping to item objects
- Adding or updating records to the database
- This is a background task, is not interuptable, and only communicates its progress to the user interface component.

Each of these entities performs its responsibilities in a sequential, non-parallel manner.

Parallelization

The user can create multiple files to be uploaded within the File Import utility. With configuration of multiple File Import Worker background tasks, the import operations are then performed in parallel, increasing the number of simultaneous operations that can be processed in a given period of time.

Configuring ExtraView to Handle Parallel Imports

Each input file is processed separately as a background operation by a File Import Worker task. If there is more than one File Import Worker task, the input files are processed simultaneously, in parallel, improving performance. In a multi-node installation, you should configure at least one File Import Worker task running on each node. Generally, it makes no difference on which node a File Import Worker task is running on unless an attachment field, image, or document is being uploaded, in which case the File Import Worker task must be running on the same node as the File Import front end, so as to be capable of reading the referenced files. You should consider configuring one File Import Worker task for each file containing your import data that you are going to upload for processing at the same time.

The File Import Worker task(s) sends add/update requests to a queue for the task named Add Update, which also runs in the background. There must be at least one Add Update task running on each node within a multi-node system. This queue is processed

by each running Add Update task taking the top entry from the queue, and performing the add or update to the database. When multiple Add Update tasks are configured, the queue is processed more quickly than if there is a single Add Update task.

Note that there is no direct relationship between the number of File Import Worker tasks and the number of Add Update tasks. There are also limits to the number of each tasks that you should configure as there are diminishing returns, dependent upon factors such as the speed of the hardware.

The optimization process is somewhat arbitrary and there are no hard and fast rules as to the optimal configuration. As a rule of thumb, these factors have proven useful in configuring large imports of data:

- For less than 30,000 records on a single node installation, only configure a single File Import Worker task and process a single import file. You might consider configuring a second Add Update task
- If you are working with larger data sets, and you have a multi-node installation, the parallelization configuration becomes worthwhile. Consider implementing a File Import Worker task for each import file and a similar number of Add Update tasks
- A limitation is that if you are importing images/documents/attachments, you must have a shared file system or limit your workers to a single node, as the file system is used to hold the temporary copy of the file so the tasks must both reside on the same physical server.

The CF_RUN_AS_ADMIN Security Permission Setting

This permission setting enables a checkbox on the file import screen. When this checkbox is set, the import utility will behave as the ADMIN user account and will ignore security permissions when importing the records. This proves to be a useful method of significantly speeding the import. It should be understood by the user performing the import that this is happening and that no field-level permission checking is happening while they import the data.

File Import - User Information

The import utility for user data allows the administrator to upload a tab-delimited or comma-delimited file from their client computer, for the fields and values within this file to be mapped to ExtraView values, to validate the data and to finally import the data. Data being uploaded is imported as new user data. You cannot use this utility to update existing user data. Access to this feature is via the security permission key named CF_USER_FILE_IMPORT. Much of the user data file import utility is similar to the previous topic of importing issue data, so this section focuses on the differences. When you choose to import user data, similarly to importing issue data, you will create a template that can be reused with different uploaded files. This is shown in the next screenshot.
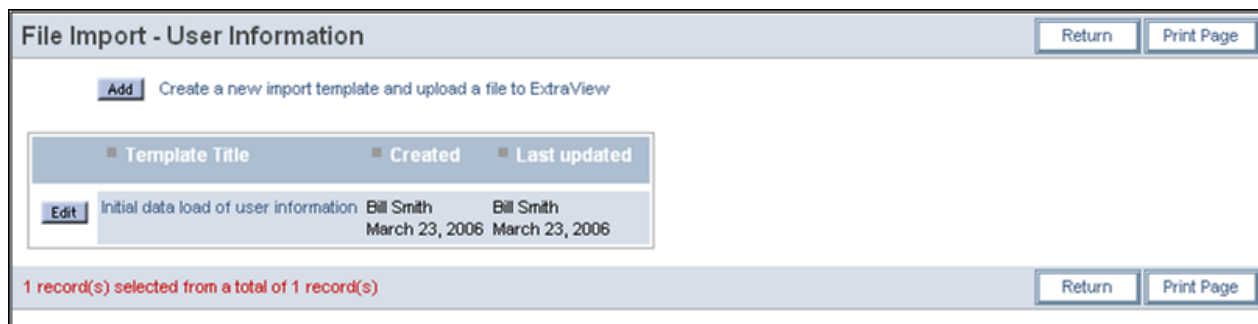
Adding a new user upload template

Once you have created a template, the process is to:

- Upload the data file containing the user data
- Map the fields in the import file to fields in ExtraView
- Make sure you have write permission to all the user fields that you are creating. The security permission fields for the user fields are all prefaced with USER
- Validate the data
- Import the data.

All mapping operations are similar to those for importing issue data. The fields that can be mapped to within ExtraView are:

Field	Purpose
ADDRESS_LINE1	The first line of the user's address. If none is specified, this is left blank
ADDRESS_LINE2	The second line of the user's address. If none is specified, this is left blank

ALPHA_TIMEZONE	The user's time zone. If none is specified, the value in the behavior setting named DEFAULT_TIMEZONE is used
AREA_ID	The default business area for the user. If none is supplied, the user will be placed in the current business area of the user performing the import
CELL_PHONE	The cell phone number of the user. If none is specified, this is left blank
CHART_FONT	The user's font they will use on charts they produce. If none is specified, the value in the behavior setting named DEFAULT_CHART_FONT is used
CITY	The City of the user. If none is specified, this is left blank
COMPANY_NAME	The user's company name. If none is specified, this is left blank
COUNTRY	The Country of the user. If none is specified, this is left blank
DATE_FORMAT	The user's date format If none is specified, the value in the behavior setting named DEFAULT_DATE_FORMAT is used
DRILLDOWN_REPORT	This has the value *QUICKLIST* by default. The two valid values are *QUICKLIST* and *DETAILED*
** EMAIL	The user's email address If none is specified, the value is left blank
EMAIL_FORMAT	The default for this field is the value HTML. The list of valid values is *HTML, TEXT, BRIF* and *VERY_BRIF*, corresponding to HTML, text, brief text and very brief text
ENABLED_USER	The default for this field is the value *Enabled*. If you want to create the user in a disabled state, then use the value of *Disabled*.
FAX	The fax number of the user. If none is specified, this is left blank
** FIRST_NAME	The first name of the user. If none is specified, this is left blank
HOME_TELEPHONE	The home telephone number of the user. If none is specified, this is left blank
HTTP_CHARSET	The user's character set If none is specified, the value in the behavior setting named HTTP_CHARSET is used
JOB_TITLE	The job title of the user. If none is specified, this is left blank
LANGUAGE	The language defaults to the setting in DEFAULT_LANGUAGE. Note that this should be in lower case. If you set another value, then ensure you have created the appropriate locale
* LAST_NAME	The last name of the user. Note this is a required field in the import file.
LOGIN_ID	The alternative User ID. If none is specified, the SECURITY_USER_ID is used
MS_OFFICE_CHARSET	The user's character set to use when outputting information to Microsoft Office products. If none is specified, the value in the behavior setting named MS_OFFICE_CHARSET is used. Note this is only useful if you are running versions of Microsoft Office prior to the Office 2003 versions
NOTIFY_ON_OWN_UPDATES	The default for this value is *Y*. You can also give it the value of *N*.
PAGER	The pager number of the user. If none is specified, this is left blank
* PASSWORD	The password, in plain text, for the user. The password is not stored in plain text, but is encrypted within the database as the user is created.
PASSWORD_EXPIRY_DATE	The date you want the user's account to expire. Note that if you do not provide this value, the password will never expire
PASSWORD_INTERVAL	The number of days between password expirations for the user. If this is not supplied, the default is zero which means the password will not expire
POSTAL_CODE	The postal code of the user. If none is specified, this is left blank
PRIVACY_GROUP	Each user may belong to multiple privacy groups; therefore importing privacy groups requires a different syntax than for most other user fields. Refer to the section above titled **Importing Repeating Row Data**. This shows how to construct the spreadsheet data to import user roles into ExtraView. Use *Privacy Group* as the title to the field.

PROJECT_ID	The default project for the user. If none is supplied, the user will be placed in the current project of the user performing the import. The project must be valid for the AREA_ID specified
RECORDS_PER_PAGE	This defaults to *20* and is the number of records that will be displayed on a home page report
REGION	The language defaults to the setting in DEFAULT_REGION. Note that this should be in upper case. If you set another value, then ensure you have created the appropriate locale with its region
REPORT_1_ID REPORT_2_ID REPORT_3_ID	This is the location to set the three Home Page reports. For each report, you must use the precise mapping of the report's name. This is case sensitive. As an example, if you want to refer to a **Public** report that has a title of **Assigned to you**, and a description of **Issues assigned to you** then you would enter **Public:Assigned to you, Issues assigned to you**
* SECURITY_USER_ID	This is a required field and is the key field for the entire user record. It must be alphanumeric, and be from one to thirty characters in length. As well as alphanumeric characters, the user ID may also contain periods ('.') and underscores ('_').
START_PAGE_ID	By default this has a value of zero. The potential values for this are: 0 Home Page 1 Query Screen 2 Add Issue Screen Note that it is possible for the administrator to add additional values to this list.
STATE	The state of the user. If none is specified, this is left blank
STYLESHEET	The default value for this field is *small*. Other valid values are *medium* and *large*. These correspond to the font size for the user's account
TWENTY_FOUR_HOUR_TIME	This defaults to *N*. You may also provide a value of *Y* if you want the user to use a twenty-four hour clock
USER_FIELD_1 USER_FIELD_2 USER_FIELD_3 USER_FIELD_4 USER_FIELD_5	These are user defined fields that may be imported into ExtraView. Their titles are set in the data dictionary, and you may import values into these fields from your import file.
USER_ROLE	Each user may have multiple roles; therefore importing roles requires a different syntax than for most other user fields. Refer to the section above titled **Importing Repeating Row Data**. This shows how to construct the spreadsheet data to import user roles into ExtraView. Use *Role* as the title to the field. See below for additional information
VARIANT	The user's region variant. If none is specified, the value in the behavior setting named is used. This is not commonly used.
WORK_TELEPHONE	The work telephone number of the user. If none is specified, this is left blank.

** indicates that the field is always required. * indicates that the field may be required, depending on the value of the behavior setting ENFORCE_DETAILED_USER_INFO. This screenshot displays a sample import file. Note how the PRIVACY_GROUP and ROLE fields may be mapped to multiple values, for a single user. The fields with the multiple values are defined in the first row, and the values are placed in rows following each user, using an **Issue Type** of **1** as opposed to the user details which have an **Issue Type** of **0**.

	A	B	C	D	E	F	G	H	I	J
1	Issue Type	SECURITY_USER_ID	FIRST_NAME	LAST_NAME	EMAIL	COMPANY_NAME	JOB_TITLE	PASSWORD	PRIVACY_GROUP	ROLE
2	0	bsmith	William	Smith	bsmith@xyz.com	Superior Software	Doctor	Welcome		
3	1	Internal	Customer Support							
4	1		IT Support Manager							
5	0	jjone	Jennifer	Jones	jjones@xyz.com	BBC Company	Laboratory	Welcome		
6	1	Cust A	Guest							
7	0	ssingh	Sanjay	Singh	ssingh@xyz.com	Superior Software	Nurse	Welcome		
8	1	Internal	Administrator							
9	1	HR	IT Support Manager							
10										

The following screen shot displays the screen where you map the fields, validate the data and perform the import. Again, please consult the previous section for full details. The one difference to importing issue data is the prompt **Allow duplicates**. When this is checked, any duplicate combinations of **First Name** and **Last Name** or duplicate values of **Email Address** will be allowed. If this is not checked, then whenever a duplicate is found, the value will be rejected during the import phase.

Validating and Importing Users

When validating the data and performing the import you will receive feedback in the same way as described in the previous section.

Note: For security purposes, the initial password of the user is set to a random value and is not defaulted to any value. The passwords will need to be set by the administrator individually, or via a custom-designed process.

Note: If inserting a new user will violate the number of licensed users, then an error is generated, and no update to the database is made. However if you do not specify the license type for a user, then they will be set to be concurrent, and this most probably will avoid any license constraints when loading the user. You may also load users in the disabled state by setting the ENABLED_USER field to a value of **D**.

Note: If your input file contains, or would cause a user to be created with a duplicate email address, or with a duplicate combination of first and last name, an error is also generated and the record is rejected.

More about Mapping User Roles

Excel is a great tool but has one inherent disadvantage for importing user role information. Each user may have multiple roles, and we need to represent that a single user may have multiple roles, all on a two-dimensional spreadsheet. To achieve this, we use the first column in the spreadsheet (with the title **Issue Type** as a label to indicate whether the row of data is the user information or the role information that refers to the user. The user information is indicated with a **0** in the cell, whereas we indicate that it is user role information in the role by placing a **1** in the cell. When you signify that the row is to hold user roles, the headings in the first row of the spreadsheet do not apply, and you can place the roles in the columns of that row. However, you must still tell ExtraView that the spreadsheet contains user roles, and this is done by placing the title **ROLE** in the first row of the spreadsheet. The Excel screenshot further up this page shows how this is configured, for both Privacy and User Roles.

File Import - Data Dictionary

The purpose of this feature is to create new fields in the data dictionary from a spreadsheet. When creating a large ExtraView installation, there is often a significant amount of replication with similar fields, but with different **names**. You are able to create a spreadsheet with a set of fields plus their attributes from a comma- or tab-delimited file, and upload this file into ExtraView. Note that the template that you download is a Microsoft Excel spreadsheet. This allows the use of validation within the spreadsheet while preparing the input file. **You must save the file as a comma-delimited or tab-delimited file before you import it back into ExtraView.**

Significant validation is performed on the file you upload, with a similar two-step process as in uploading issue data. You are also able to save the templates you create for future use, allowing you to prepare fields in batches, and upload them as and when you are prepared.

Importing fields from a spreadsheet

Preparing the import file

Use Microsoft Excel or a similar spreadsheet tool to create your input data. An empty spreadsheet for use as an input file can be downloaded from within the utility. **You must save the file as a comma-delimited or tab-delimited file before you import it back into ExtraView** See the previous screenshot for details of how to download an empty spreadsheet.
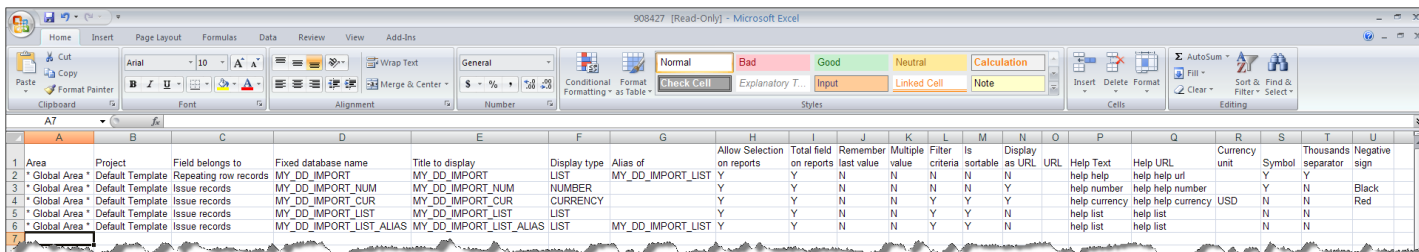
Similar to the web-based input for the data dictionary, the following fields are available for input:

Field Name	Description
Area	The title of the Business Area for the field. If the field is to be global, use * **Global Area** *. This field is required
Project	The title of the Project within the Business Area to which you want to load the fields. If you are creating a global field, this will be * **Master Project** * If you are loading the fields for use across a single Business Area, this will be the default Project for the Business Area. If you want the field to only be visible within a single Project, then you will use the title of that Project. This field is required
Field belongs to	This must be set to **Issue records** if you want the field to be available to the main body of issues, or to **Repeating row records** if you want to use the field on a repeating row. This field is required
Fixed name	This is the fixed name for the field. Names can only consist of the characters A to Z, 0 to 9, and '_'. The first character of a fixed name must be alphabetic and the name can be up to 30 characters in length. Names cannot contain characters from non-English alphabets. A further restriction is that you may not have two underscore characters together in a name. This field is required
Title to display	The display title for the field. This may be up to 100 characters long and may contain any characters such as punctuation. This field is required
Display type	This must map to a valid display type within ExtraView. This field is required. The valid entries are: • Button • Checkbox • Currency • Custom • Date • Day • Decimal • HTML Area • Image • Label • List • Log Area • Number • Pop-up • Print Text • Radio Button - Horizontal • Radio Button - Vertical • Tab • Text Area • Text Field • Time Interval • User
Alias of	If the field is to be created as an alias of another field, and the other field is of type list, tab, or radio button then place the other field name in this column
Allow Selection	Create as Yes or No

on reports	
Total field on reports	Create as Yes or No
Remember last value	Create as Yes or No
Multiple value	Create as Yes or No
Filter criteria	Create as Yes or No
Is sortable	Create as Yes or No
Display as URL	Create as Yes or No
URL	If you are using the field as a display as URL, then provide the URL here
Help Text	The help text for the tooltip on the field
Help URL	The URL to use if you are providing your own help on the field
Currency unit	The currency unit for the field
Currency symbol	The currency symbol to be used for the field
Thousands separator	Create as Yes or No
Negative sign	Create as Yes or No
Internal precision	A number from -24 to 10 indicating how many digits of precision should be maintained for internal computations with this field
Report precision	A number from -24 to 10 indicating how many digits of precision should be displayed on reports or read-only displays of this field
Rounding mode	Select from one of the following list: • Round Up • Round Down • Round Floor • Round Half Up • Round Half Down • Round Half Even
Percentage type	Create as Yes or No

Uploading the File and Providing any Mappings

This utility is very similar in usage to importing issue data. Please see the previous section in this guide for full instructions.

File Import - Data Dictionary | Refresh mappings / Upload new file | Validate import file | Perform import | Delete template | Cancel | Print Page

From this screen you can perform all the tasks to import your file. You can edit the template details, and upload a different import file to use with the template. This is where you map fields and their values from the imported file to ExtraView. Finally, you can validate your imported file and perform the actual import.

Title: Fields for new business area   [Localize]

Upload a different file: [_____] [Browse...]

File character set [?]: UTF-8 Unicode 8-bit Transfer

Sample of uploaded file:
```
Area, Project, Field belongs to, Fixed name, Title to display, Display type, Alias of, Allow Selection on reports, Total field on
* Global Area *,Default Template,Repeating row records,MY_DD_IMPORT,MY_DD_IMPORT,LIST,MY_DD_IMPORT_LIST,Y,Y
* Global Area *,Default Template,Issue records,MY_DD_IMPORT_NUM,MY_DD_IMPORT_NUM,NUMBER,,Y,Y,N,N,N,N,Y,,h
* Global Area *,Default Template,Issue records,MY_DD_IMPORT_CUR,MY_DD_IMPORT_CUR,CURRENCY,,Y,Y,N,N,Y,Y,Y.
```

Tab-separated fields [?] [ ]

Override existing field [?] [ ]

Field Mappings

For each import file field, choose an ExtraView field to map your data into. ExtraView will map the values within a field, if the same value exists in both the import file and ExtraView. To provide alternative mappings, click on the Map button. Choose *None* if you do not want to map a field in the import file. Click on the Refresh mappings button to see the results of your mapping.

Import file field	ExtraView field	
[1] Area	Area(AREA_ID)	Map
[2] Project	Project(PROJECT_ID)	Map
[3] Field belongs to	Field belongs to(ITEM_TYPE_ID)	Map
[4] Fixed name	Fixed name(NAME)	Map
[5] Title to display	Title to display(TITLE)	Map
[6] Display type	Display type(DISPLAY_TYPE)	Map
[7] Alias of	Alias of(ALIAS_OF)	Map
[8] Allow Selection on reports	Allow selection on reports(SELECT_FOR_REPORTS)	Map
[9] Total field on reports	Total field on reports(COL_REPORT_TOTAL)	Map
[10] Remember last value	Remember last value(REMEMBER_LAST_VAL)	Map
[11] Multiple value	Multiple value(MULTIPLE_VALUE)	Map
[12] Filter criteria	Filter criteria(FILTER_CRITERIA)	Map
[13] Is sortable	Is sortable(IS_SORTABLE)	Map
[14] Display as URL	Display as URL(DISPLAY_AS_URL)	Map
[15] URL	URL(URL)	Map
[16] Help Text	Help Text(HELP_TEXT)	Map
[17] Help URL	Help URL(HELP_URL)	Map
[18] Currency unit	Currency unit(CURRENCY_UNIT)	Map
[19] Currency symbol	Currency symbol(SHOW_CUR_SYMBOL)	Map
[20] Thousands separator	Thousands separator(SHOW_THOU_SEP)	Map
[21] Negative sign	Negative sign(NEGATIVE_SIGN)	Map

Mapping Results

This displays the results of mapping your data and how it will appear in ExtraView. The top field name in the table is the field in your import file, the bottom field is the ExtraView field. Click on any header, if you want to map list values from your input file, to different values in ExtraView. Press the Refresh button after changing any mapping.

Field values will be inserted into ExtraView   Field values will not be inserted into ExtraView

Row Number	Area Area(AREA_ID)	Project Project(PROJECT_ID)	Field belongs to Field belongs to(ITEM_TYPE_ID)	Fixed name Fixed name(NAME)	Title to display Title to display(TITLE)	Display type Display type(DISPLAY_TYPE)	Alias of Alias of(ALIAS
1	* Global Area *	Default Template	Repeating row records	MY_DD_IMPORT	MY_DD_IMPORT	LIST	MY_DD_IMPO
2	* Global Area *	Default Template	Issue records	MY_DD_IMPORT_NUM	MY_DD_IMPORT_NUM	NUMBER	
3	* Global Area *	Default Template	Issue records	MY_DD_IMPORT_CUR	MY_DD_IMPORT_CUR	CURRENCY	
4	* Global Area *	Default Template	Issue records	MY_DD_IMPORT_LIST	MY_DD_IMPORT_LIST	LIST	
5	* Global Area *	Default Template	Issue records	MY_DD_IMPORT_LIST_ALIAS	MY_DD_IMPORT_LIST_ALIAS	LIST	MY_DD_IMPO
6	* Global Area *	Default Template	Issue records	MY_DD_IMPORT_LIST_ALIAS_2	MY_DD_IMPORT_LIST_ALIAS_2	LIST	MY_DD_IMPO

Importing fields into the data dictionary

The key difference in using this utility and using the import issue data feature is that there is a prompt on the screen, **Override existing field**. If you check this, then you may update the properties of the field given in the **Fixed name** column.

Advanced Menu

The Advanced Menu contains administrative topics that are not regularly accessed. The administrative features within this menu are:

- ExtraView Version Information – This displays information regarding the server installation

- ExtraView Licenses – the End User License Agreement (EULA) for ExtraView
- Allowed Locales – If this is enabled in your system, this utility allows you to add a new language locale to the installation
- Translate System Messages and Prompts – If this is enabled in your system, this utility allows you to translate all the system messages and metadata in your system to an allowed locale
- Upload Server-side templates for use with ExtraView API – If you are creating server-side templates for use with external web pages, this utility allows you to upload the templates, without having access to the file system of the server
- Statistics – Key statistics of ExtraView usage
- System Log Types – Maintains the list of items that are logged within the System Log
- System Security Keys – This displays and allows you to create security keys within the installation, to which permissions are attached by the Grant Security Privileges routine
- View Allowed Functions – This displays all the functions within your installation and shows the difference between the allowed functions in the ExtraView Standard version and the ExtraView Global Compliance version
- Start Page Management – This controls the list of pages that can be addressed directly when a user first signs on to ExtraView
- Pre-cache Layouts at System Startup – This controls the list of layouts that can be loaded when the application server is started, providing improved performance when users first sign onto the system
- Edit the labels of the Administration tabs – This allows the editing of the labels on the tabs of the administration page. The only purpose for this utility is to localize the labels on the tabs

Version Information

This function, found under the **Advanced Menu** tab within Administration, provides the system user with all the vital operational, maintenance, and troubleshooting information available about their specific ExtraView installation. Companies that are self-hosting ExtraView on their own servers may use this feature as a convenient way to find information regarding their environment such as the ExtraView build number, third party software versions, operating system details, Java environment details, etc.

A screen similar to the following appears:

ExtraView Version Information		Return	Print Page

Build label	UNKNOWN
Build date	Mon 02/27/2006 16:07:53.73
Build OS	Windows_NT
Build DB Host	127.0.0.1
Build DB SID	nerdvana
Build DB User	best_data
Build DB Url	jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(HOST=127.0.0.1)(PROTOCOL=tcp)(PORT=1521))(CONNECT_DATA=(SID=nerdvana)))
Build Instance	/evj5.0
Build Domain	.extraview.net
Build Connection Pool Size	20
Build Connection Pool Max	200
Build Connection Use Count	500
Build Connection Timeout	10
Build Connection Pool Timeout	20
License expiration date/time	NEVER
Database product name	Oracle
Database product version	Oracle9i Enterprise Edition Release 9.2.0.1.0 - Production With the Partitioning, OLAP and Oracle Data Mining options JServer Release 9.2.0.1.0 - Production
JDBC version	9.2.0.3.0
java.runtime.name	Java(TM) 2 Runtime Environment, Standard Edition
sun.boot.library.path	C:/JBuilderX/jdk1.4/jre/bin
java.vm.version	1.4.2_01-b06
java.vm.vendor	Sun Microsystems Inc.
java.vendor.url	http://java.sun.com/
path.separator	;
java.vm.name	Java HotSpot(TM) Client VM
file.encoding.pkg	sun.io
user.country	US
sun.os.patch.level	Service Pack 2
java.vm.specification.name	Java Virtual Machine Specification
user.dir	C:/Documents and Settings/Carl/My Documents/ExtraView/projects/evj5.0
java.runtime.version	1.4.2_01-b06
java.awt.graphicsenv	sun.awt.Win32GraphicsEnvironment
java.endorsed.dirs	C:/JBuilderX/jdk1.4/jre/lib/endorsed
os.arch	x86
java.io.tmpdir	C:/DOCUME~1/Carl/LOCALS~1/Temp/
line.separator	
java.vm.specification.vendor	Sun Microsystems Inc.
user.variant	
java.naming.factory.url.pkgs	org.apache.naming
os.name	Windows XP
sun.java2d.fontpath	
log4j.configuration	file:c:/ApacheGroup/jakarta-tomcat-4.1.30/conf/log4j.properties
java.library.path	C:/JBuilderX/jdk1.4/bin; .;C:/WINDOWS/system32;C:/WINDOWS;C:/WINDOWS/system32;C:/WINDOWS;C:/WINDOWS/system32/WBEM;C:/oracle/ora92/bin;C:/Program Files/Oracle/jre/1.3.1/bin;C:/Program Files/Oracle/jre/1.8/bin;C:/Program Files/ATI Technologies/ATI Control Panel;C:/Program Files/Intel/Wireless/Bin/;C:/Program Files/Microsoft SQL Server/80/Tools/Binn/;C:/Program Files/UltraEdit;C:/Program Files/SecureCRT;C:/Program Files/Perforce;C:/Program Files/Intel/Wireless/Bin/;C:/Program Files/Intel/Wireless

Version Information screen

This information is useful for support purposes. ExtraView support personnel may ask you to review information on this screen when debugging problems.

ExtraView Licenses

The ExtraView end user license agreement (EULA) is available for reading within the **Advanced Menu** section of Administration.

Note that the license displayed here is the standard online version, which is agreed to when you first start ExtraView. This license may be superseded by a license agreement negotiated between ExtraView Corporation and your company. If such as license exists, then the license displayed here is not valid.

Note that, like virtually all modern enterprise software applications, ExtraView takes advantage of several open source components, which themselves are subject to licenses provided by the authors of these components. ExtraView only uses these open source components in their original, binary form, and does not alter the code. The licenses for these open source components are stored within the ExtraView distribution. These can be found within your deployed tree within the directory at /locales/en_US/license.

Allowed Locales

This function allows you to define the locales that users may select with the Language prompt on the user administration screen. Before this option appears within administration, you must set the behavior setting named MULTI_LOCALE to YES, and the security permission key named CF_ALLOWED_LOCALES must give you read and write permissions.

When you first access this function, you will see a list of the locales that are set up, and you can choose to modify the settings of these with the **Edit** button, or you can add a new locale.

List of Allowed Locales

When you add a new language package, you instantly allow any user to switch to that language, but localized messages may not exist for that language. You can use the **Translate System Messages and Prompts**, or the **Localize** buttons that appear to provide the messages in any new language you define.

Adding a new locale does more that giving the capability to translate the messages into that language. All date formats and other locale specific features will use different defaults. For example, the default locale is en_US (English US), but you may add a new locale of en_GB for British English. When you do this, the default date formats will be those to which a British user is familiar, rather than the traditional ones familiar to a user in the USA.

When no localized messages are available, or if the program is missing some localized messages in any specific language, ExtraView will display the language entry for the behavior setting named in DEFAULT_LOCALE. This is normally en_US. Note that a user in a country such as Britain can override a few localized messages, to eliminate any US English spelling.

Note: Once the setting named MULTI_LOCALE has been set to YES in an ExtraView database, and any changes to any text message has been made, the setting should not be reset to NO. Doing so may result in seeing text in a variety of locales being displayed to the user.

Localization

If the option to **Translate System Messages & Prompts** is enabled, the end user is able to select, as a personal preference, which language in which he wants to operate. There are three fundamental areas where localized messages can be entered and maintained:

- The user interface of ExtraView, including all system messages and prompts. Typically, these messages are the same for all installations of ExtraView. For example, this includes informative and error messages that come from the ExtraView software.
- The metadata values provided by the administrator, which are local to the installation. For example, if the administrator creates titles for each status value used across his organization in English, and he wants to provide the localized equivalents in French, German and Japanese, he can insert the appropriate translations.
- The metadata values provided by users, which are local to the installation. For example, if a user creates a report title used across his organization in English, and he wants to provide the localized equivalents in French, German and Japanese, he or another user may insert the appropriate translations.

These areas of localization are independent, and can be used in a mutually exclusive fashion. Localized versions of images can also be installed on the server. Please see the section on Initial Setup & Configuration for details.

Note: If a particular term or message is not translated into a specific language, ExtraView will choose to display the item with the default locale (normally US English). This can be used with advantage, for example, to provide a UK English localization, with the minimum of effort, only translating messages and metadata that have different spellings between the two languages.

Note: No matter what setting is used for a user's language preference, data can be entered and updated in any language, assuming the user's character set and browser provide the necessary support. For this reason, ExtraView recommends that the default character set used in the installation of an Oracle database be set at UTF-8, and that for a Microsoft SQL Server database, the character should be UCS-2.

Note: Before performing any localization, you should check your browser is set to use the UTF-8 character set. Failure to do this may result in corrupting previously saved messages that contain accented or double-byte characters. If you are not sure whether your browser is correctly configured, make sure you take a backup of your database before you start. Testing is extremely important as part of localizing ExtraView metadata and prompts into any language. Please thoroughly check your work before deploying into a production environment.

Entering Accented Characters

As stated above, your browser should be set to use the UTF-8 character set. However, there are still some issues to be aware of when entering characters. The characters that present the most problems are the quote (') and the double quote ("). The reason for this is that these characters are often used within a computer program to specify a string value being sent to the user interface. For example, the statement –

```
document.write('Today is Wednesday')
```

may be written by a JavaScript function to a screen. When this is translated to French, this would look like –

```
document.write('Aujourd'hui est Mercredi')
```

The additional quote in the middle of the text will cause a syntax error when the browser attempts to display the character. For this reason, you may need to substitute the character value of the quote as so –

```
document.write('Aujourd&#39;hui est Mercredi')
```

or as

```
document.write('Aujourd&#x27;hui est Mercredi')
```

The first method uses the decimal value of the quote; the second method uses the hexadecimal value.

This same principle can be used for all characters. Please see the Appendix on Character Set Values for a list of characters, their decimal and hexadecimal values. Some characters, such as the double quote also have a form like –

```
&quot;
```

End-User Selection of Language

Each user can select the language in which he wants to operate, by updating his account options. From the title bar at the top of each screen, the user clicks on the Account: link, enters their password and then sees a screen similar to the following, after clicking on the **Personal Options** tab.

The pull-down list labeled *Language* displays the available languages for this installation. Once the language is chosen, the account is updated and this language will be used for the display of all system messages, prompts and metadata.

User Accounts Screen

Localizing Screen Messages

This function is accessed from the Administration tab named **Advanced**, and is named **Translate System Messages and Prompts**. When you select this link, you will see a screen similar to the following:

Selecting a language to translate

Choose the language you want to localize and the category of messages. Choosing the * **System Text Messages** * option provides access to all the system level messages. Other categories you may choose are * **End User System Text Messages** * and * **Admin**

System Text Messages *. Alternatively choose a different category to localize the metadata values of a specific object. Once you choose these options, the screen refreshes, similar to the following:

Screen where messages are localized

Note the option with the radio button that allows you to display all messages, or to only display the messages that have not yet been translated. This gives you the ability to home in on the remaining messages to be localized, as opposed to viewing all messages. The messages appear in alphabetic order of the 'from' language. Edit any of the messages or prompts and then press the update button at the foot of the screen.

- Messages may contain one or more numbers within braces such as **Add entry to the {0} for {1}** or **Currently, there are {0,number,integer} user(s) connected.** Do not remove the numbers and information within braces from the localized message. The numbers within braces may be placed anywhere within the localized message
- Occasionally, messages in English may contain a numeric representation of a character. An example might be **Alternative User ID's must be unique.** Again, do not alter the representation, which is **'** in the example
- If there is HTML formatting tags within the message, such as: **<i>ExtraView</i> will not function correctly without cookies.** You can safely alter the HTML tags, but it is recommended that you leave them unaltered
- Messages that are longer than 100 characters in length are not displayed in full, but the full message to be translated can be displayed. Place your mouse over the shortened message and a tooltip will appear that displays the entire message
- The Java file names at the right hand side of the screen represent which file(s) the messages appear within, and the line number at which the message occurs within the file. This gives an indication of where about on the user interface that any mesage appears.

Localizing Business Rules

Business Rules are developed in the base language of the system, indicated by the behavior settings named DEFAULT_LANGUAGE and DEFAULT_REGION. DEFAULT_LANGUAGE normally has the value of **en** for English. The DEFAULT_REGION normally has the value of **US** which represents the value for the United States of America. Note that these values are case sensitive. It is unusual to alter these values, but it is usual to define other locales and assign users to these locales. The **Translate System Messages and Prompts** utility allows for the translation of messages into any locale, which is the combination of the language and region. The allowable locales for any system are defined within the administration utility named **Allowed Locales**.

In order to localize business rules, special consideration needs to be given and rules for the alternative languages need to be developed.

The following business rule actions may need to be localized, as they typically contain text which is displayed to the end user:

- ERROR
- STOP
- REAUTHORIZE

Consider a business rule in US English, such as this:

```
{STOP: Your entry is in error.  Please correct and submit again}
```

If you also want to support French and Spanish langauge alternatives that have been added as allowed locales, you may do this:

```
if (USER.{language}) = 'en') {
    {STOP: Your entry is in error.  Please correct and submit again}
}
if (USER.{language} = 'fr') {
    {STOP: Votre entrée est erronée. Veuillez corriger et soumettre de nouveau }
}
if (USER.{language}) = 'es') {
    {STOP: Su entrada está en error. Corrija y envíe de nuevo }
}
```

If you want to add variants of languages, you will also need to use the user's region within the logic. For example, if you want to support variants for different versions of English, you might do this:

```
if (USER.{language}) = 'en' && USER.{region} = 'US') {
    {STOP: Your entry for the field named Color is in error.  Please correct.}
}
if (USER.{language}) = 'en' && USER.{region} = 'GB') {
    {STOP: Your entry for the field named Colour is in error.  Please correct.}
}
```

Note the different spellings for the word Color.

Localizing Custom Code

In order to be able to localize messages that appear to users and that are coded within Java in ExtraView's custom code modules, the messages must be correctly coded to conform to the output requirements. Assuming the messages are correctly coded, the following procedure will integrate the messages into the appropriate message tables within the database. Thereafter, the messages are integrated into the translation screens and can be translated into any allowed locale.

This process should only be carried out by a developer who has good knowledge of ExtraView's internal structure and user custom coding.

- Check out the **UCMessageTest.java** from your source code repository, and make sure it is writeable
- Select and run the **build fapUC** build target within the **build.xml** script. This generates a new version of **UCMessageTest.java**
- Run the script **WEB-INF/data/UpgradeMessage.sh -u** or **WEB-INF/data/UpgradeMessage.bat -u** to insert/update the messages within the database.

There is no problem executing this process as many times as required. On each run, only new or changed messages will be inserted into the ExtraView database.

Localizing Metadata

Once localization is switched on within an ExtraView installation, all titles that can be localized will have a new button, named Localize, by the title within the administration section. For example, the titles of items with lists can be localized, and the titles within the data dictionary can be localized.

Note: You may localize all the metadata values on the same screen as that which localizes system messages. The two methods complement each other. You may either localize the metadata values when you access the underlying item, or in a single place on the system metadata screen.

It is partiuculary convenient to localize all the titles of user defined fields using the **Translate System Messages and Prompts** screen.

Once again, if you do not provide a localized equivalent for a value, ExtraView will revert to displaying the default locale's language. Usually this is US English. When you edit a status value, for example, the screen will look something similar to:

Localization button for metadata

When you first press the Localize button, you see a screen similar to the following:

Localizing metadata

Notice that as no localization has yet been completed for this value, all available languages show the term "Open" as the value to display. Editing the Japanese value will show a screen like:

Creating a Japanese Status entry

When you have completed editing all the localized value for the status fields, the screen for the values will look similar to:

Completed localized text for a Status entry

Localizing User Defined Field Titles

From the **Translate System Messages and Prompts** administration screen, select the *target language*, and the metadata category of **UDF_LIST**. You may then select the specific User Defined Field for which you want to localize the titles. You will only see the fields to which you have write permission. This produces a screen similar to the following from which you can translate the titles to the UDF you selected:

Localizing Popup Calendars

The popup calendars seen on *add* and *edit* screens have a series of files that should be localized for each language. A wide variety of languages have predefined calendars, but should you need to localize a langauge not supplied, follow this procedure:

- Navigate to the web\javascript\zpcal\lang directory within your ExtraView installation
- You will see that the naming convention for the language files is **calendar-xx.js** where **xx** is the 2 character country code, such as **es** for the Spanish language
- Copy the **calendar-en.js** file to a file named **calendar-xx.js**, replacing **xx** with the appropriate 2 character country code
- Edit the new file and replace each character string for day names, month names and the tooltips with the appropriate local language equivalents
- Save the file.

Globalization File Structure

If ExtraView Corporation is hosting your installation, you do not have direct access to the file system of the server to configure, alter or use this feature without contacting ExtraView support.

If you are considering altering the user interface, or adding additional image sets, or customizing the help pages, it is important to understand where ExtraView places files that can be altered or localized. The types of files that can be altered are as follows:

- Image files – note that the file with your company's logo is treated differently from the remainder of the images
- Help files

The following tree structure shows the location of directories on a Microsoft Windows server.
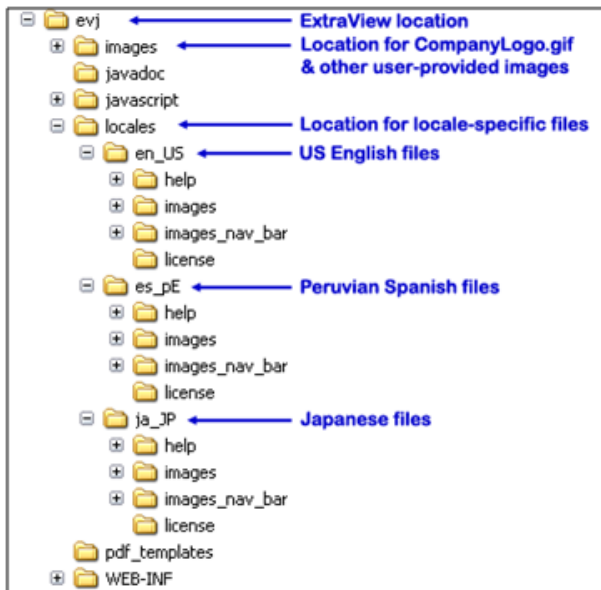
Localization directory structure

- The image with your company logo is placed within the directory named /images . Its name is CompanyLogo.gif. There may only be a single company logo file for an instance.
- The /images directory also contains a directory named /tab_corners . This contains the images used to construct tabs that appear within ExtraView.
- The /locales directory is where all the locale specific image and help files are stored. ExtraView is always shipped with at least one set of files within this directory. This is named /en_US and contains US English image and help files. /en_US is the default locale for the installation. Other locale specific directories can be created and will contain the appropriate images and help.

Metadata Migration of Localized Data

The administrative utilities **Metadata Export** and **Metadata Import** are used to move localized data from one ExtraView instance to another. The migration of localized data may be independent of the movement of other metadata or may be part of the overall movement of data.

For further details consult the pages on these utilities here.

Manage Server-Side Templates

Server-side templates are used to format information returned from the API to a client browser or script. For example, when you enter a new issue from the API to ExtraView, you can format the information returned by the command (the ID of the issue or an error if the command failed) with a template, perhaps to style the output in the same form as your company's website.

This is fully explained in the **Application Programming Interface Guide**. Templates are typically stored in the directory named WEB-INF/user_templates. To aid in placing templates on the server, when you may not have access to the directory structure of your server, this administrative function allows you to upload server-side templates, and allows you to delete them. They must be edited on your client computer.

Access to this feature is controlled by the security key named CF_UPLOAD_USER_TEMPLATES.

From the Administration menu, click on the **System Controls** tab. Then click on the link named Upload Server-Side Templates for use with ExtraView API. You will see the following screen:
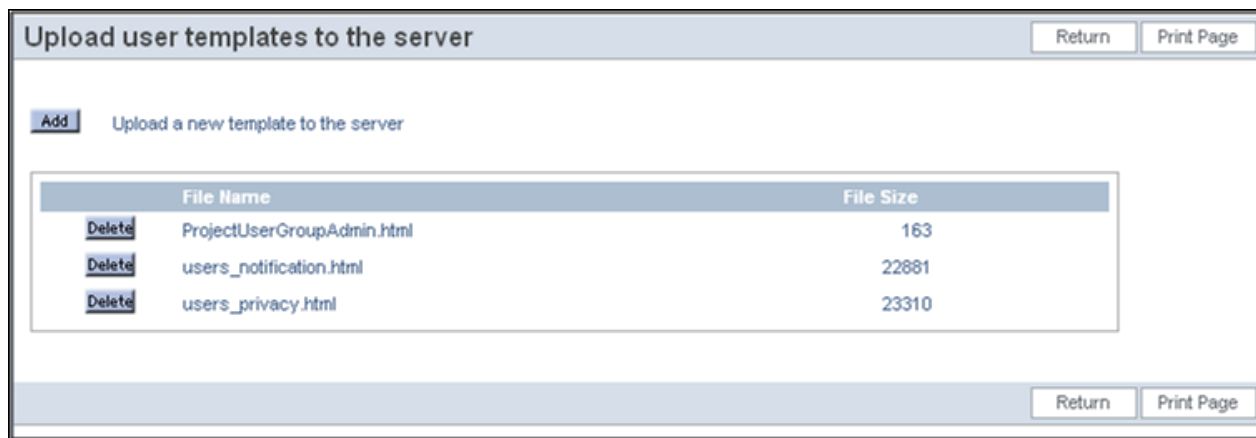
Managing Server-side templates

The **Add** button allows you to add a new template. Once templates have been uploaded to the server, you can delete them with the **Delete** button.

Note: You cannot use this feature if you are running ExtraView from within a deployed War file.

Statistics

ExtraView maintains statistics on key information. Statistics are accessed via the Administration menu, under the System Controls tab.

Click **Statistics** to view the statistics.

The page is divided into 5 main sections: User Statistics, Issue Statistics, File Attachments, Background Time Counts, and Background Frequencies

User Statistics	
Total number of users (active and inactive)	15
Number of active users	14
Number of active customer users	2
Number of internal (non customer) users	12
Number of active users in user role *Administrator*	4
Number of active users in user role *Customer*	7
Number of active users in user role *Customer Support*	5
Number of active users in user role *Customer Support Manager*	2
Number of active users in user role *Development Engineer*	7
Number of active users in user role *Engineering Manager*	3
Number of active users in user role *IT Support*	5
Number of active users in user role *IT Support Manager*	3
Number of active users in user role *Quality Assurance*	6

User Statistics section

- **Total number of users (active and inactive)**. This is the total number of users stored in ExtraView's database. It includes both active and deactivated user accounts.
- **Number of active users**. This is the number of active user accounts, and excludes deactivated accounts.
- **Number of active customer users**. This is the number of active users, in the user role specified by the behavior setting named LIMITED_USER_ROLE.
- **Number of internal (non-customer) users**. This is the number of users who are allocated to users groups other than the LIMITED_USER_ROLE. If a user is allocated to both LIMITED_USER_ROLE and any other group, they will appear in this total.
- **Number of active users in user role *group_name***. This is the number of active users in the user role named *group_name*.

Issue Statistics	
Total number of issues in database	347
Number of issues created to date this month	0
Number of issues created last month (January)	0
Number of issues created in last 30 days	0
Number of updates applied to date this month	0
Number of updates applied last month (January)	0
Number of updates applied in last 30 days	0

Issue Statistics section

File attachments	
Number of attachments in system	29
Average size of attachment (kB)	126
Maximum size of an attachment (kB)	1602

File Attachments section

Background Time Counts	
Database data manipulation language	0
Database connection allocation	0

Background Time Counts

Background Frequencies	
Database data manipulation language	15
Database connection allocation	1

Background Frequencies

The Background Time Counts and Frequencies match those shown in EVSTATS for individual services, but they are accumulated for any task that runs in the background, such as SESSION_MONITOR, TASK_CONTROL, FULL_TEXT_SYNCHRONIZE, and BATCHMAIL when run as a background task.

The values are accumulated for the period of time from the most recent instance start of your application server (boot time) to the current time. The database manipulation language (DML) statistics are those used to access or modify the database, in particular, any SELECT, UPDATE or DELETE SQL statements.

The database connection allocation frequency is the number of times a connection is allocated from the ExtraView Connection Pool.

System Security Keys

It is unlikely that you will need to access this screen, therefore it is turned off by default. Should you require access to this feature, set the security permission key named SE_SECURITY_MODULE to read and write.

Each field, button, each screen and many features within ExtraView have corresponding System Security Keys. From the **Grant Security Privileges** section under the **Fields & Layouts** tab, you are able to control the visibility of the elements controlled by these keys, thereby restricting the access to these fields, buttons, and features.

Note: Changes in the system security keys area have system-wide implications. ExtraView Corporation strongly recommends that, other than editing the display titles (which appear in the grant security privileges screens) you not attempt to make changes in this area without first contacting ExtraView Support. It is unlikely that you will need to create new keys.

System Security Keys Summary screen

Edit an Existing Security Key

1. Click on the **System Security Keys** option on the ExtraView Administration menu, under the **Advanced** tab.
2. Click on **Edit** and change the description or title as needed.

System Log Types

It is unlikely that the administrator will need to add or alter entry types to the system log. Each entry is used to monitor the usage of ExtraView metadata alterations by individual users, giving an audit of who changed which piece of metadata. The system log itself can be viewed from the **System Controls** administration menu.

Allowed Functions

This screen is purely for information purposes and there is no user interaction allowed.

ExtraView is licensed software that comes in several versions. The specific features for any version are controlled with an internal, encrypted, security device. This screen shows which features are available with each of the different license levels.

Access the screen via **Admin**, **Advanced**, **View Allowed Functions**. You will see a screen similar to this:

Managing Start Pages

The behavior setting named USER_DEFINED_START_PAGE must be set to YES before this feature is used. This feature allows the administrator to add additional start pages to be available. A start page is the location on which a user lands, after they successfully sign on. The default pages available are the Home Page, the Add Issue screen, the Search / Report screen and the Administration screen. It is unlikely that you will need more than these, but they can be added through this screen if necessary. Before adding a new entry, you will need to know the calling convention to be used. These are internal to ExtraView and are defined in terms of the Action and the Option. We suggest that if you need a new start page, you should consult with ExtraView support to define these values.

The list of start pages

Changing the Administration Menu Titles

This option is rarely used, but allows the administrator to re-title or localize the administration menus. You can turn this option off with the security permission key named CF_ADMIN_MENUS if you do not wish to see this on your screen.

Altering the titles of the options within Administration

To alter or localize a title, access this function, then press the Edit button. There you may alter the title, or provide a localized title. Update after you alter the title and you will immediately see your new title within the administration screen.

Pre-Caching Layouts Upon System Restart

- [Add new comment](#)

When a layout is first accessed, either upon a restart of the application server, or upon a change to the layout by the administrator, it is read by ExtraView, and compiled into a form that can be rendered speedily. This action may take a half-second for simple layouts to many seconds for complex layouts.

This feature is controlled by the security permission key named SE_TEMPLATE_PRECACHE.

If you do not want an end-user to face this delay when they are the first person to access a layout, you can use the utility at **Fields & Layouts** --> **Controls pre-caching templates at system startup** to compile and cache layouts when the application server is restarted.

Pre-caching Layouts

ExtraView has an overall limit of 100 layouts held in the cache at one time, and if this number is exceeded, then the oldest layout is removed from the cache. At the same time, there is a cost in time to pre-cache layouts, therefore the recommendation is to only pre-cache layouts if a user is likely to complain about the time taken to reach the layout if they are first person to use it.

To pre-cache layouts, use the filters at the top of the screen to select the layout(s) you want to cache. This will be a combination of user role, business area, project, layout type and the locale with which the layout is to be displayed. Once you select the filters, use the **Add Selected Keys** button to add the layouts to the pre-cache list. You can remove layouts individually using the **Del** button, or you can set up keys and remove a list of layouts at one time.

The caching is controlled by a system task named TEMPLATE_LOADER. This task must be started in order for the caching to take place. Go to the **Manage Tasks and Threads** administrative utility and **Add** the task. The task should be set up to run on each node of a clustered system. It is recommended that you set the poll interval to zero. This causes the task to run once upon system startup and not to be run again.

Caching layouts is somewhat tricky, as a full understanding of exactly what needs to be cached to speed up the initial load time is essential. Some recommendations:

- Do not assume that trying to pre-cache all layouts will lead to better performance
- Do not cache the layouts in the Master Projects for any area. Only cache the layouts in the projects that actually store the user's data
- First cache the *Add* and *Edit* layouts for the most frequently used business areas in their data projects
- If you have embedded layouts in the *Add* and *Edit* layouts then cache these
- Cache the layouts only for the most frequently used roles
- Cache the layouts only for the most frequently used locales

- On a sizeable system, there can be many, many layouts when you look at the combination of the layouts, the business areas, the projects, the user roles and the locales. Look to cache only the most frequently used combinations, because of the limitation of being able to pre-cache 100 layouts.

Encryption Key Management

ExtraView is a secure application and has been developed to adhere to the strictest standards. Additional security is important and should be implemented by ensuring the environment within which ExtraView resides is secure, using firewalls and encryption of the network traffic with SSL, etc.

It is possible to protect sensitive data within the database using a further level of encryption. This protects against your database from unauthorized access, and unauthorized users using SQL commands to extract data. This encryption is provided for fields with a display type of text, and is applied on a field-by-field basis. This can be used, for example, to provide added protection for fields such as social security numbers and bank account numbers.

Authorized users will still see data that is encrypted within the database in the normal way. It is assumed that users will be educated on the sensitivity and privacy of data, and that the screens and reports they view may contain proprietary data that should not be stored with unauthorized personnel.

This feature is only available in the ExtraView Enterprise version. To provide the security, secret keys (or passwords) are stored in what is termed a keystore within ExtraView. You may use a single key for all the fields you protect, or use different keys for different fields.

These keys must be stored safely by the administrator, and **there is no way to ever retrieve a lost key**. This is critical to understand and neither the ExtraView application nor ExtraView Corporation personnel can ever retrieve a lost key. It cannot be stressed too much how important it is to have a process to record the keys used, and to keep them safe.

Further, here are items that should be considered before implementing encrypted fields:

- Only text fields that do not have data stored as yet may be encrypted
- Once encrypted, you cannot revert a field to being non-encrypted
- You cannot use an encrypted field as a filter in a query or report
- Encrypted fields cannot be sorted within reports

The above limitations may be relaxed within future versions of the product.

There are 3 possibilities of how the secret keys may be used to unlock the data, dependent upon your requirements:

1. The secret key(s) must be entered manually by the administrator on each and every occasion after starting the application. This is accomplished by the administrator going to the **Encryption Key Management** utility and entering the passwords for each of the secret keys
2. The passwords may be entered into the `Configuration.properties` file which is read by ExtraView upon startup of the application. This is more convenient than entering the passwords manually, but not quite as secure. Of course, the filesystem with the `Configuration.properties` (and other) files should be protected
3. You can encrypt the entries within the `Configuration.properties` file using the technique described in the ENCRYPT_PROPERTIES section of the [Installation & Configuration Guide](#).

Setting Up Encryption

1. Enter the administrative utility, **Encryption Key Management**, on the **Advanced** administration tab
2. Carefully read the directions, remembering lost keys can never be retrieved
3. Click **Add a new encrpytion key to the keystore**
4. Provide a fixed name for the key. This follows the usual naming conventions of all ExtraView names
5. Enter the secret key password
6. Re-enter the secret key password
7. Now, use the **Data Dictionary** to add the field whose contents you wish to encrypt. It must have a display type of text, and you must select the option to **Encrypt this Field.** Note that encrypted fields may not be used as filters, nor may they be made sortable fields
8. You will be prompted for the secret key to use for the encryption. Enter the password you entered into the keystore
9. From this point, the field is configured like all other fields and may be placed on any applicable layout or report.

Unused keys may be deleted from the database, and the keys within the keystore may be updated by the administrator after providing the current key.

Recommendations

- Only encrypt fields that contain sensitive data. There is an overhead to encrypt and decrypt their contents and you do not want to impact performance negatively
- Only provide multiple secret keys if different administrators each maintain the field contents that are encrypted within each field. You do not want to create an excessive administrative burden
- Again - make sure you keep the passwords safe. ExtraView Corporation cannot retrieve lost passwords. This is the essence of security, and there is no "back door" to work around this.

Appendices

Date and Time Formats

Overview

Each ExtraView user has the capability to set their date and time format. The date and time formats determine how dates and times are displayed in output text or html and how they are parsed when entered by the user. The user can select their date and time format on the Personal Options screen. Either one of the in-built formats can be selected, or the user can set the **Date Format** to the value of **Custom Date Mask** and then provide their own mask in the field labeled **Date Mask**.

Localizable Date and Time Formats

Dates and times are represented differently around the world. An ExtraView user may choose one of the date/time formats that are sensitive to the specific locale they are using. The following date and time formats will be represented differently dependent on the user's current locale setting. The following table shows the date format names used in ExtraView, with an example of how this appears in English/US locale representation:

Format Name	Description	Example
SHORT	Short without time	12/30/02
MEDIUMDATE	Medium without time	Dec 30, 2002
LONGDATE	Long without time	December 30, 2002
FULLDATE	Full without time	Monday December 30, 2002
SHORTDATETIME	Short with time	12/30/02 1:15 PM
MEDIUMDATETIME	Medium with time	Dec 30, 2002 1:15 PM
LONGDATETIME	Long with time	December 30, 2002 1:15 PM
FULLDATETIME	Full with time	Monday December 30, 2002 1:15 PM

Standard Date/Time Parsing Formats

Certain date and time formats are built in to ExtraView as patterns to be used for parsing dates. If a date is entered in one of the standard formats, it will be parsed without any errors. Use of a custom date mask extends the possible date/time entry formats to include the custom mask pattern.

The following are the standardized date/time parsing formats:

Format	Example
MM/dd/yy	04/21/03
dd/MM/yy	21/04/03
MM/dd/yyyy	04/21/2003
dd/MM/yyyy	21/04/2003
dd-MMM-yy	21-April-03

dd-MMM-yyyy	21-April-2003
yyyy/MM/dd HH:mm:ss z	2003/04/21 10:23:34 PST
yyyy/MM/dd HH:mm:ss	2003/04/21 10:23:34
yyyy/MM/dd	2003/04/21
yyyy-MM-dd HH:mm:ss	2003-04-21 10:23:34
yyyy MM dd HH:mm:ss	2003 04 21 10:23:34
MM-dd-yyyy HH:mm	04-21-2003 10:23
yyyy-MM-dd	2003-04-21
MM/dd/yy HH:mm	04/21/03 10:23
MM/dd/yy HH	04/21/03 10
yy-MM-dd	03-04-21
MMM dd yyyy	April 21 2003
MMM dd, yyyy	April 21, 2003
MMM. dd yyyy	Apr. 21 2003
MMM. dd, yyyy	Apr. 21, 2003

Custom Date Masks

A custom date mask provides the ExtraView user with the maximum of flexibility, but the representation of dates and times using a custom date mask are formatted similarly regardless of the user's current locale setting. Text strings such as month names or days of week within the representation are locale-sensitive.

Entry of date values into forms can use one of the ExtraView standard date formats or whatever format the user has defined using a custom date mask.

Custom date masks consist of a sequence of characters consisting of punctuation or pattern letters that are chosen from the following:

Letter	Date or Time Component	Usage	Examples
G	Era designator	G	AD
y	Year	yyyy, yy	1996; 96
Y	Week Year (context sensitive)	YYYY, YY	2015; 15
M	Month in year (standalone	MMMM, MMM, MM	July; Jul; 07
L	Month in year	LLLL, LLL, LL	July; Jul; 07
w	Week in year	ww	29
W	Week in month	W	3
D	Day in year	DDD	219
d	Day in month	dd	10
F	Day of week in month	F	3
E	Day in week	EEEE, EE	Tuesday; Tue
a	Am/pm marker	a	PM
H	Hour in day (0 - 23)	HH	24
k	Hour in day (1-24)	kk	24
K	Hour in am/pm (0 - 11)	KK	10
h	Hour in am/pm (1-12)	hh	12

m	Minute in hour	mm	30
s	Second in minute	ss	55
S	Millisecond	SSS	834
z	Time zone (general)	Z	Pacific Standard Time; PST; GMT-08:00
Z	Time zone (RFC 822)	Z	-0800
X	Time zone (ISO 8601)	X	-08:00

Pattern letters are usually repeated, as their number determines the exact presentation:

- **Text**: For formatting, if the number of pattern letters is 4 or more, the full form is used; otherwise a short or abbreviated form is used if available. For parsing, both forms are accepted, independent of the number of pattern letters.
- **Number**: For formatting, the number of pattern letters is the minimum number of digits, and shorter numbers are zero-padded to this amount. For parsing, the number of pattern letters is ignored unless it's needed to separate two adjacent fields.
- **Year**: For formatting, if the number of pattern letters is 2, the year is truncated to 2 digits; otherwise it is interpreted as a number. For parsing, if the number of pattern letters is more than 2, the year is interpreted literally, regardless of the number of digits. So using the pattern "MM/dd/yyyy", "01/11/12" parses to "Jan 11, 12 A.D". For parsing with the abbreviated year pattern ("y" or "yy"), ExtraView must interpret the abbreviated year relative to some century. In ExtraView, the year on a date parsed from user input is always checked to see if it is greater than 1000. If not, the year is adjusted to a more "rational" value by adding either 1900 or 2000 to the specified year. 2000 is added if the result would be less than or equal to the current year, otherwise, 1900 is added. For example, using a pattern of "MM/dd/yy" and a date presented on Jan 1, 1997, the string "01/11/03" would be interpreted as Jan 11, 2003 while the string "05/04/64" would be interpreted as May 4, 1964.
- **Month**: If the number of pattern letters is 3 or more, the month is interpreted as text; otherwise, it is interpreted as a number.
- **General time zone**: Time zones are interpreted as text if they have names. For time zones representing a GMT offset value, the following syntax is used: GMTOffsetTimeZone: GMT Sign Hours : Minutes Sign: one of - Hours: Digit Digit Digit Minutes: Digit Digit Digit: one of 0 1 2 3 4 5 6 7 8 9 Hours must be between 0 and 23, and Minutes must be between 0 and 59. The format is locale independent and digits must be taken from the Basic Latin block of the Unicode standard. For parsing, RFC 822 time zones are also accepted.
- **RFC 822 time zone**: For formatting, the RFC 822 4-digit time zone format is used: RFC822TimeZone: Sign TwoDigitHours Minutes TwoDigitHours: Digit Digit TwoDigitHours must be between 00 and 23. Other definitions are as for general time zones. For parsing, general time zones are also accepted.

Security Permission Keys

Permission Key	Explanation
CF_ADMIN_MENU	Edit the labels of the Administration tabs
CF_ALL_BEHAVIOR_SETTINGS	The complete list of behavior settings (in alphabetic order) can be viewed and modified from this list. Write access is needed to control this
CF_ALLOW_CHANGE_ROLE	Display the Role: change option on the menu bar
CF_ALLOWED_VALUES	Access to the screen for maintenance of allowed value lists. These lists can also be maintained individually from the child list maintenance screen
CF_ALLOWEDLOCALES	Maintain the list of locales allowed to be set by users in this installation. Note that even though a locale is defined, the translated messages may need to be created. Write access is needed to control this
CF_ALLOWEDVALUE_TYPE	Define and specify a parent-child hierarchical relationship between two fields. Write access is needed to control this
CF_API_SETTINGS	Access to the configuration screen to maintain behavior settings that access ExtraView via the API. Write access is needed to control this
CF_AREA	Maintain the list of Business Areas for the inbuilt AREA field. Write access is needed to control this
CF_CATEGORY	Maintain the list of Categories for the inbuilt CATEGORY field. Write access is needed to control this
CF_COMPANY_SETTINGS	Access the administration screen to set your company name, address and contact information. Write access is needed to control this

CF_DATA_DICTIONARY	Access to the data dictionary where you create and maintain inbuilt and user defined fields & other objects, including modifying their title, display type, basic behavior and default values. Write access is needed to control this
CF_DEBUG_SETTINGS	Access the behavior settings for debugging options and the user custom classname. Write access is needed to control this
CF_DISPLAY_INFO	Access the behavior settings for fonts and colors to be used within ExtraView screens and layouts. Write access is needed to control this
CF_DISPLAY_SETTINGS	Control and set behavior settings that affect how screens and layouts are presented. Write access is needed to control this
CF_EMAIL_SETTINGS	Setup and administer email behavior settings. Write access is needed to control this
CF_ENABLE_DISABLE_USER_ACCESS	Access the administrative option to quiesce the ExtraView system for maintenance. Write access is needed to control this
CF_ENVIRONMENT	Access the behavior settings that maintain physical paths and URL information for accessing ExtraView. Write access is needed to control this
CF_ESCALATION_RULES	Escalation Rules
CF_EXPIRE_PASSWORD	This key controls access to allow the administrator to expire password in user accounts. Write access is needed to control this
CF_FILE_IMPORT	Access to the File Import Utility to upload and import a tab-delimited or comma-delimited data file containing records. Write access is needed to control this
CF_HIERARCHY	Reporting Hierarchies
CF_HIERARCHY_LEVEL	Reporting Hierarchy Levels
CF_INSTALLATION_DETAILS	Setup the basic behavior settings of the ExtraView installation. Write access is needed to control this
CF_INTEREST_LIST	Interest List access. Note that giving read and write access to this key will allow non-administrative users access to maintain interest lists. This access is from a prompt on the Notification section of their Personal Options screen. Be aware that the end-user will have full access to interest list maintenance if they are given this permission
CF_ITEMDATA	Access the administrative function to export item data to an XML formatted file. The file can be used for import to a different ExtraView schema. Read access is needed to control this
CF_LAYOUT	Create and alter the layouts for all screens and reports. Write access is needed to control this
CF_LAYOUT_TYPE	Maintain the list of types of screen and report layouts that can be defined. Write access is needed to control this
CF_LICENSE	The End User License Agreement (EULA) for ExtraView. Write access is needed to control this
CF_LOCALIZE	Access to this key allows locale-specific titles to be added to metadata field values. Write access is needed to control this
CF_MANAGE_USER_CONNECTION	Manage and disconnect users connection to ExtraView, releasing licenses back to the common pool. This is only used for concurrent license installations. Write access is needed to control this
CF_METADATA	Provide access to the export function where system metadata is dumped in XML format. Read access is needed to control this
CF_METADATA_UPDATER	Imports XML from a formatted file created with the XML export utility, and updates ExtraView with the information. Please make sure you backup your installation before using! Write access is needed to control this
CF_MODULE	Access to maintain the list of modules (MODULE_ID) within products

(PRODUCT_NAME). Write access is needed to control this

CF_MODULE_TYPE	Create and manage the list of module types. Note that this key may be deprecated in a future release. Write access is needed to control this
CF_OBJECT_IMPORT	Access to import item data from an imported file prepared in XML format. Write access is needed to control this
CF_PERSONAL_OPTIONS	This key controls access to each user role's personal option link that appears in the title bar of the screen. Write access is needed to control this
CF_PRIORITY	Maintain the list of values for the inbuilt PRIORITY field. Write access is needed to control this
CF_PRODUCT	Maintain the list of values for the inbuilt PRODUCT_NAME field. Write access is needed to control this
CF_PRODUCT_LINE	Maintain the list of values for the inbuilt PRODUCT_LINE field. Write access is needed to control this
CF_PROJECT	Access and maintain the list of Projects (PROJECT) within Business Areas (AREA). Write access is needed to control this. There is no use case to have read or write access to this key unless the same access is given to the CF_AREA permission key
CF_RELATIONSHIP_GROUP	This key controls access to the global settings and names of issue relationship groups. Write access is needed to control this
CF_RELATIONSHIP_GROUP_DELETE _BUTTON	This key controls access to delete an issue from within a relationship group. Write access is needed to control this
CF_RELATIONSHIP_GROUP_EDIT	This key controls whether a user role may modify an existing relationship group of issues. Write access is needed to control this
CF_RELATIONSHIP_GROUP_EDIT _PROBLEM_BUTTON	This permission key controls the presence of the "Edit issue" button on the relationship group screen. Write access is needed to control this
CF_RELATIONSHIP_GROUP_PROBLEM _ADD_PROBLEM_BUTTON	This key controls whether a user role may add an issue to an existing relationship group of issues. Write access is needed to control this
CF_RELATIONSHIP_GROUP_VIEW _PROBLEM_BUTTON	This permission key controls the presence of the "View" button on the relationship group screen. Write access is needed to control this
CF_REPORT_SETTINGS	This key grants permissions to alter the behavior settings that affect reports and queries. Write access is needed to control this
CF_REPORT_USER	Create and manage reports created by individual users
CF_RESOLUTION	Maintain the list of values for the inbuilt RESOLUTION field. Write access is needed to control this
CF_RUN_AS_ADMIN	This permission setting allows the user who is performing a file import to ignore field level permissions. The primary purpose of this is to speed up the importation of extremely large volumes of records. It's unlikely that this setting is needed with less than 25,000 records within the import file
CF_SESSION_SETTINGS	This key controls access to the maintenance screen for system-wide behavior settings. Write access is needed to control this
CF_SEVERITY	Maintain the list of values for the inbuilt SEVERITY_LEVEL field. Write access is needed to control this
CF_SSO_SETTINGS	Set up ExtraView to work with SSO and LDAP servers
CF_STATISTICS	This provides access to summary information about users, user roles and usage patterns.. Read access is needed to control this
CF_STATUS	Maintain the list of values for the inbuilt STATUS field. Write access is needed to control this
CF_STATUS_RULES	This controls access to the definition screen for rules that apply to status changes. Sets up the business rules for workflow based on status. Write access is needed to control this

CF_SYSTEM_LOG	Controls access to reporting on all metadata changes in the administration log. Read access is needed to control this
CF_SYSTEM_LOG_TYPE	The list of activities in the administration log. New entries must be supported by program changes at the source code level
CF_SYSTEM_SIGNON	Controls access to view the log entries that track the sign on and sign off activities of all users. Read access is needed to control this
CF_TEMPLATE	Controls access to create and edit email templates to use in communication with internal and external users when updating issues
CF_TRANSLATOR	Controls access to translate all system messages and prompts into different languages. Write access is needed to control this
CF_UDF_LIST	Controls access to maintain the list of values for User Defined Fields stored within ExtraView. This is a single key that controls access to all lists. Each member of the list also has an individual security key. Write access is needed to control this
CF_UPLOAD_COMPANY_LOGO	Controls access to the feature that allows an administrator to upload a Company Logo
CF_UPLOAD_USER_TEMPLATES	Write access to this key allows the uploading of Server-Side Templates for use with the ExtraView API
CF_USER_FILE_IMPORT	Controls access to the File Import - User Information feature. Write access allows the administrator to create new users with this facility
CF_USER_SETTINGS	This key controls access to the behavior settings and options associated with user names and users. Write access is needed to control this
CF_VERSION_INFO	This key controls access to version information for ExtraView and its supporting software. Read access is needed to control this
CF_VIEW_APP_SERVER_LOG	Read access to this permission key will place a button on the System Log administration screen. Pressing this button will display the application server log in a new window.
CF_WORKFLOW_DEFAULTS	This key controls access to the behavior settings and options associated with workflow. Write access is needed to control this
MENU_ADD_PROBLEM	This setting controls access to the Add issue layout from the main navigation bar Add button. Write access is needed to control this
MENU_ADMINISTRATION	This setting controls access to the Administration system from the main navigation bar Administration button. Write access is needed to control this
MENU_EDIT	This setting controls access to the Update button on any Edit screen. Write access is needed to control this
MENU_DRILLDOWN_ON_NAV_BAR	This permission key controls the presence of the drilldown box on the navigation bar. Read access is needed to be able to access the drilldown box.
MENU_HOME	This key controls access to the Home button on the Navigation bar. Read access is needed to control this
MENU_RESOLUTION	This key controls access to the Query (Search & Reports) screen. Read access is needed to control this
MOBILE_NAV_BAR	This controls the visibility of the navigation bar within the mobile client. You require read permission to see the nvigation bar
PR_ADD_PROBLEM.ALT_ID	Alternate ID
PR_ADD_PROBLEM.AREA	Business Area
PR_ADD_PROBLEM.ASSIGNED_TO	Assigned To
PR_ADD_PROBLEM. ATTACH_CONTENT_TYPE	This controls the presence of the content type on an attachment record. This can be seen with read access, but cannot be written to

PR_ADD_PROBLEM. ATTACH_CREATED_BY_USER	This controls the presence of the user's name who created the attachment. This can be seen with read access, but cannot be written to
PR_ADD_PROBLEM. ATTACH_DATE_CREATED	This controls the presence of the date an attachment was created. This can be seen with read access, but cannot be written to
PR_ADD_PROBLEM. ATTACH_FILE_DESC	This controls the presence of the attachment description, when you are entering the file name to be uploaded to an attachment. Write access is needed to control this
PR_ADD_PROBLEM. ATTACH_FILE_NAME	This controls the presence of the attachment file name on an attachment record. This can be seen with read access, but cannot be written to
PR_ADD_PROBLEM. ATTACH_FILE_SIZE	This controls the presence of the file size on an attachment record. This can be seen with read access, but cannot be written to
PR_ADD_PROBLEM. ATTACH_HIST_TIMESTAMP	Controls the presence of a timestamp that shows when an attachment was accessed, This is viewable with the read permission key on the History screen of an attachment
PR_ADD_PROBLEM. ATTACH_LAST_UPDT_COMPANY	Controls the presence of the name of a user's company that accessed an attachment, and viewable with the read permission key on the History screen of an attachment
PR_ADD_PROBLEM. ATTACH_PATH	This controls the presence of the original client attachment path to an attachment file, on an attachment record. This can be seen with read access, but cannot be written to
PR_ADD_PROBLEM. ATTACH_THUMBNAIL	Controls the presence of a thumbnail of an attachment of an image type, when the record that contains the attachment is viewed on the edit screen or on a report containing the ATTACHMENT record
PR_ADD_PROBLEM. ATTACHMENT	This is the controlling key for attachments. Without read and write permission to this key, attachments will not appear on the add issue screen layout. Once this key gives permission to the user role, the remaining attachment keys control the read / write permissions to the individual fields
PR_ADD_PROBLEM. ATTACHMENT_ADD	This controls the Add button on attachments. Write access is needed to control this
PR_ADD_PROBLEM.CATEGORY	Category
PR_ADD_PROBLEM.CC_EMAIL	Controls the presence of the CC Email input box on the add screen. The user role should have read and write permission to this field to allow the users to input email addresses on this line
PR_ADD_PROBLEM. CC_EMAIL_BUTTON	Controls the presence of a button by the CC Email input box on the add screen. The user role should have read permission to this field to allow the user to pop up a list of users that can be added to the CC Email box
PR_ADD_PROBLEM.COMMENTS	Comments
PR_ADD_PROBLEM.CONFIRM_ALLOW_EDIT	This permission key controls the visibility of the Edit button on the Add Confirmation screen
PR_ADD_PROBLEM.CONTACT	Contact
PR_ADD_PROBLEM.DATE_CREATED	Date Created
PR_ADD_PROBLEM.DESCRIPTION	Description
PR_ADD_PROBLEM.EMAIL_CUSTOMER	If the user role has Read permission to this field, the checkbox to allow email to be sent to external users will be visible
PR_ADD_PROBLEM.EMAIL_SWITCH	This key enables or disables the Generate Email checkbox for the user role. If there is no Read access to this key, the Generate Email checkbox is not visible, and its setting is obtained from the value of the behavior setting named GENERATE_EMAIL_BOX.
PR_ADD_PROBLEM.ID	Controls access to the inbuilt field named ID
PR_ADD_PROBLEM.INTEREST_LIST	If the user role has write access to this permission key, they will see a

checkbox in the notification area of the add screen and will be able to add themselves to an interest list for the issue they are creating. The user will also be able to add other users to an interest list for the issue via an input box

PR_ADD_PROBLEM.ITEM_ID	Controls access to the inbuilt field named ITEM_ID
PR_ADD_PROBLEM.MAILING_LIST	Controls access to the distribution email list for the issue. When the user has Read access to this permission key, an icon is also available to refresh the list
PR_ADD_PROBLEM.MODULE_ID	Module ID - this is the main key that should be used to control the Module field on a form
PR_ADD_PROBLEM.ORIGINATOR	Originator of the problem
PR_ADD_PROBLEM.OWNER	The owner of the issue
PR_ADD_PROBLEM.PRIORITY	The priority of the issue
PR_ADD_PROBLEM.PRIVACY	The Privacy of the issue
PR_ADD_PROBLEM. PROBLEM_RELEASE_DELETE	This key controls the presence of a check box on a repeating row record, that allows you to delete the current row. Write access is needed to control the checkbox
PR_ADD_PROBLEM.PRODUCT_LINE	Product Line
PR_ADD_PROBLEM.PRODUCT_NAME	Product name
PR_ADD_PROBLEM.PROJECT	Project
PR_ADD_PROBLEM.RELEASE	This is the controlling key for repeating row records. If you have read and write permission to this key, the entire repeating row structure will be available to the user role, and access to the individual fields on the repeating row structure is controlled by the security permission keys for each field
PR_ADD_PROBLEM. RELEASE_ASSIGNED_TO	Release Assigned To
PR_ADD_PROBLEM. RELEASE_DATE_CREATED	Release Date Created
PR_ADD_PROBLEM.RELEASE_FIXED	Release Fixed
PR_ADD_PROBLEM.RELEASE_FOUND	Release Found
PR_ADD_PROBLEM.RELEASE_OWNER	Release Owner
PR_ADD_PROBLEM. RELEASE_PRIORITY	Release Priority
PR_ADD_PROBLEM. RELEASE_PRODUCT	Release Product
PR_ADD_PROBLEM. RELEASE_RESOLUTION	Release Resolution
PR_ADD_PROBLEM. RELEASE_SEVERITY	Release Severity
PR_ADD_PROBLEM.RELEASE_STATUS	Release Status
PR_ADD_PROBLEM. RELEASE_TIMESTAMP	Release Timestamp
PR_ADD_PROBLEM.RESOLUTION	Resolution
PR_ADD_PROBLEM.SEVERITY_LEVEL	Severity Level
PR_ADD_PROBLEM.SHORT_DESCR	Short description or Title of an issue
PR_ADD_PROBLEM. SHOW_PRIVATE_IN_PRIVACY_LIST	When you give a user role write access to this key, they will be able to enter Private issues within the privacy field
PR_ADD_PROBLEM. SHOW_PUBLIC_IN_PRIVACY_LIST	When you give a user role write access to this key, they will be able to enter Public issues within the privacy field
PR_ADD_PROBLEM.STATUS	Status
PR_ADD_PROBLEM.TIMESTAMP	Last date modified

PR_RESOLUTION. ALLOW_EDIT_CLOSED	Access to this key controls whether a user role can edit "closed" issues, as defined by the behavior setting named STATUS_CLOSED_NAME.
PR_RESOLUTION. ALLOW_EDIT_NEXT_PREVIOUS	Access to this key controls whether a user role can move from one issue in a list created from a report, to the next or the previous issue in the report list, reviewing and updating each issue in turn. When this is enabled for a role, there will be two additional buttons on the edit screen of each issue allowing the user to move back and forwards in the list of issues on the current page of the report in which the user first started to edit issues.
PR_RESOLUTION.ALT_ID	Alternate ID
PR_RESOLUTION.AREA	Business Area
PR_RESOLUTION.ASSIGNED_TO	Assigned To
PR_RESOLUTION. ATTACH_CONTENT_TYPE	This controls the presence of the content type on an attachment record. This can be seen with read access, but cannot be written to
PR_RESOLUTION. ATTACH_CREATED_BY_USER	This controls the presence of the user's name who created the attachment. This can be seen with read access, but cannot be written to
PR_RESOLUTION.ATTACH_DATE _CREATED	This controls the presence of the date an attachment was created. This can be seen with read access, but cannot be written to
PR_RESOLUTION. ATTACH_FILE_DESC	This controls the presence of the attachment description, when you are entering the file name to be uploaded to an attachment. Write access is needed to control this
PR_RESOLUTION.ATTACH_FILE_NAME	This controls the presence of the attachment file name on an attachment record. This can be seen with read access, but cannot be written to
PR_RESOLUTION.ATTACH_FILE_SIZE	This controls the presence of the file size on an attachment record. This can be seen with read access, but cannot be written to
PR_RESOLUTION.ATTACH_PATH	This controls the presence of the original client attachment path to an attachment file, on an attachment record. This can be seen with read access, but cannot be written to
PR_RESOLUTION.ATTACHMENT	This is the controlling key for attachments. Without read and write permission to this key, attachments will not appear on the *edit* issue screen layout. Once this key gives permission to the user role, the remaining attachment keys control the read / write permissions to the individual fields
PR_RESOLUTION.ATTACHMENT_ADD	This controls the Add button on attachments. Write access is needed to control this
PR_RESOLUTION. ATTACHMENT_DELETE	This is to enable and disable the delete attachment button
PR_RESOLUTION.ATTACHMENT_EDIT	Controls whether a user role may edit the attachment description of a previously uploaded attachment. Write permission is needed for this
PR_RESOLUTION.ATTACHMENT_VIEW	Controls whether a user role may view a previously uploaded attachment. Read or write permission is needed for this
PR_RESOLUTION.CATEGORY	Category
PR_RESOLUTION.CC_EMAIL	Controls the presence of the CC Email input box on the *edit* screen. The user role should have read and write permission to this field to allow the users to input email addresses on this line
PR_RESOLUTION.CC_EMAIL_BUTTON	Controls the presence of a button by the CC Email input box on the *edit* screen. The user role should have read permission to this field to allow the user to pop up a list of users that can be added to the CC Email box
PR_RESOLUTION.CLONE	Controls the presence of a Clone button on the *edit* screen toolbar. The user should have write permission to be able to clone issues
PR_RESOLUTION.COMMENTS	Comments
PR_RESOLUTION.CONTACT	Contact
PR_RESOLUTION.DATE_CREATED	Date Created

PR_RESOLUTION.DELETE_BUTTON	Controls the presence of the Delete button on the *edit* screen. If a user role has write access to this key they will be able to delete the issue
PR_RESOLUTION.DESCRIPTION	Description
PR_RESOLUTION.EDIT_BUTTON	Controls whether a user role has access to an Edit button within reports and email. Write access is needed for this control
PR_RESOLUTION. EDIT_LOGAREA_FIELDS	Controls whether a user role is allowed to edit historic logarea fields such as COMMENTS. Normally only administrators will be given this control, through write access to the key
PR_RESOLUTION.EMAIL_BUTTON	Determines whether a user role has the Email button on the toolbar of the Edit screen
PR_RESOLUTION.EMAIL_CUSTOMER	If the user role has Read permission to this field, the checkbox to allow email to be sent to external users will be visible
PR_RESOLUTION. EMAIL_FILTER_SCREEN	Read permission to this key controls access to the search filter screen for ad hoc email
PR_RESOLUTION.EMAIL_SWITCH	This key enables or disables the Generate Email checkbox for the user role. If there is no Read access to this key, the Generate Email checkbox is not visible, and its setting is obtained from the value of the behavior setting named GENERATE_EMAIL_BOX.
PR_RESOLUTION.HISTORY_BUTTON	Read access to this key gives access to the history (audit trail) of an issue. History buttons can be placed on the *edit* screen and reports.
PR_RESOLUTION.ID	Controls access to the inbuilt field named ID
PR_RESOLUTION.INTEREST_LIST	If the user role has write access to this permission key, they will see a checkbox in the notification area of the edit screen and will be able to add themselves to an interest list for the issue they are updating. The user will also be able to add other users to an interest list for the issue via an input box
PR_RESOLUTION.ITEM_ID	Controls access to the inbuilt field named ITEM_ID
PR_RESOLUTION.KEYWORD	Determines whether a user role may perform searches by keyword, through the KEYWORD field on a filter layout. Read access is needed
PR_RESOLUTION. LAST_CHANGE_USER	Last user to update an issue
PR_RESOLUTION.MAILING_LIST	Controls access to the distribution email list for the issue. When the user has Read access to this permission key, an icon is also available to refresh the list
PR_RESOLUTION.MASS_DELETE_ISSUES	This controls access to the feature that allows users to perform mass deletions of issues from the database. Read and write access is required. Normally this facility is only given to key personnel
PR_RESOLUTION. MASS_UPDATE_ISSUES	Controls access to the Mass Update button on report output. Write permission is needed to perform mass updates. Normally this facility is only given to key personnel
PR_RESOLUTION.MODULE_ID	Module ID - this is the main key that should be used to control the Module field on a form
PR_RESOLUTION.MONTHS _IN_STATUS	Number of months a problem has remained in its current status
PR_RESOLUTION.MONTHS_OPEN	Number of months a problem has been open
PR_RESOLUTION.OWNER	The Owner of an issue
PR_RESOLUTION.PRIORITY	The priority of the issue
PR_RESOLUTION.PRIVACY	The Privacy of the issue
PR_RESOLUTION. PROBLEM_RELEASE_DELETE	This key controls the presence of a check box on a repeating row record, that allows you to delete the current row. Write access is needed to control the checkbox
PR_RESOLUTION.PRODUCT_LINE	Product Line

PR_RESOLUTION.PRODUCT_NAME	Product name
PR_RESOLUTION_PROBLEM. PROBLEM_RELEASE_DELETE	This key controls the presence of a check box on a repeating row record, that allows you to delete the current row. Write access is needed to control the checkbox
PR_RESOLUTION.PROJECT	Project
PR_RESOLUTION. RELATIONSHIP_GROUP	Relationship Groups on Edit Screen
PR_RESOLUTION. RELATIONSHIP_GROUP_ID	Controls visibility of the relationship group ID on the *edit* screen
PR_RESOLUTION. RELATIONSHIP_GROUP_OWNER	Read access to this key provides the user role with visibility of the owner of relationship groups
PR_RESOLUTION. RELATIONSHIP_GROUP_TITLE	Read access to this key provides the user role with visibility of the title of relationship groups
PR_RESOLUTION. RELATIONSHIP_GROUP_TYPE	Read access to this key provides the user role with visibility of the type of relationship groups
PR_RESOLUTION. RELATIONSHIP_GRP_ADMIN	This permission key controls access to the Manage Relationship Groups button that is placed on the menubar of edit screens. When permission is granted to this key, users can access the management screen for relationship groups directly from the edit screen.
PR_RESOLUTION. RELATIONSHIP_GRP_PARENT_ID	Read access to this key provides visibility to the parent issue of the relationship group
PR_RESOLUTION.RELEASE	This is the controlling key for repeating row records. If you have read and write permission to this key, the entire repeating row structure will be available to the user role, and access to the individual fields on the repeating row structure is controlled by the security permission keys for each field
PR_RESOLUTION. RELEASE_ASSIGNED_TO	Release Assigned To
PR_RESOLUTION. RELEASE_DATE_CREATED	Release Date Created
PR_RESOLUTION.RELEASE_FIXED	Release Fixed
PR_RESOLUTION.RELEASE_FOUND	Release Found
PR_RESOLUTION.RELEASE_OWNER	Release Owner
PR_RESOLUTION.RELEASE_PRIORITY	Release Priority
PR_RESOLUTION.RELEASE_PRODUCT	Release Product
PR_RESOLUTION. RELEASE_RESOLUTION	Release Resolution
PR_RESOLUTION.RELEASE_SEVERITY	Release Severity
PR_RESOLUTION.RELEASE_STATUS	Release Status
PR_RESOLUTION. RELEASE_TIMESTAMP	Release Timestamp
PR_RESOLUTION.RESOLUTION	Resolution
PR_RESOLUTION.RG_EMAIL_BUTTON	Provides access to the email button on the Relationship Group Filter Screen.
PR_RESOLUTION.RG_MERGE_BUTTON	Provides access to the merge button on the Relationship Group Filter Screen
PR_RESOLUTION.RG_SPLIT_BUTTON	Provides access to the split button on the Relationship Group Filter Screen
PR_RESOLUTION. SEARCH_ATTACHMENTS	This key controls the searching of attachments on the query filter screens. When this key has Write access, a checkbox appears underneath the KEYWORD field, giving the ability to search through attachments for the user role
PR_RESOLUTION.SEVERITY_LEVEL	Severity Level
PR_RESOLUTION.SHORT_DESCR	Short description or Title of an issue

PR_RESOLUTION.SHOW_PRIVATE_IN_PRIVACY_LIST	When you give a user role write access to this key, they will be able to enter Private issues within the privacy field
PR_RESOLUTION.SHOW_PUBLIC_IN_PRIVACY_LIST	When you give a user role write access to this key, they will be able to enter Public issues within the privacy field
PR_RESOLUTION.START_DATE	This key offers the user role the ability to enter the Created Date as the lower value in a query filter when searching for an issue. Typically used in conjunction with PR_RESOLUTION.STOP_DATE
PR_RESOLUTION.START_UPDATE	This key offers the user role the ability to enter the Last Modified Date as the lower value in a query filter when searching for an issue. Typically used in conjunction with PR_RESOLUTION.STOP_UPDATE. Note that the data dictionary setting "**Filter criteria**" must be set to **Yes** for this field to work.
PR_RESOLUTION.STATUS	Status
PR_RESOLUTION.STATUS_HIST	Status History
PR_RESOLUTION.STOP_DATE	This key offers the user role the ability to enter the Created Date as the upper value in a query filter when searching for an issue. Typically used in conjunction with PR_RESOLUTION.START_DATE
PR_RESOLUTION.STOP_UPDATE	This key offers the user role the ability to enter the Last Modified Date as the upper value in a query filter when searching for an issue. Typically used in conjunction with PR_RESOLUTION.START_UPDATE. Note that the data dictionary setting "**Filter criteria**" must be set to **Yes** for this field to work.
PR_RESOLUTION.TIMESTAMP	Last date modified
PR_RESOLUTION.VIEW_BUTTON	Controls the visibility of the View button on reports. Read access is required for this
SE_LOGIN_MESSAGE	Provides access to alter the sign on message that appears on all users' Home Pages. Write access is required for this function
SE_PRIVACY_GROUP	Write access allows the creation and management of the list of privacy groups
SE_SECURITY_GROUP	Create and manage the list of user roles with this key. Write access is needed
SE_SECURITY_MODULE	Create and maintain the list of security objects. Normally these are created automatically by ExtraView. Write access is required
SE_SECURITY_PERMISSION	This key controls access to update the security permission keys within the installation. Write access is required. Without access to this key, no security permissions can be altered
SE_SECURITY_USER	Provides access to create and maintain user accounts and user account details. Write access is needed
SE_USER_ATTRIBUTES	If the user's role has permission to this key, then they are able to use the user defined attributes that have been configured within the data dictionary, as fields for a custom operation on the user account administration screens. The use of this feature is always accompanied by user custom code
SR_ALLOW_ADVANCED_QUERIES	When this setting is YES, the user role can create, edit, save, delete and execute reports composed with the Advanced Query option. When this is set to NO, reports composed with Advanced Query may only be run, and only if the user role has permission to the appropriate SR_PUBLIC_COLUMN_REPORT and SR_PERSONAL_COLUMN_REPORT permission keys
SR_ALLOW_EXPANDED_QUERIES	When the user role has read access to this key, the users can switch between condensed and expanded queries. When this key is set to no access, the users will only be able to create queries with the condensed mode and will not be able to switch into the expanded mode
SR_DASHBOARD_ON_HOME_PAGE	Read access provides the user role with access to an executive dashboard report on the home page. Note that this report must be configured in user custom code

SR_KB_ON_HOME_PAGE	Read access provides the user role with access to search the inbuilt Knowledgebase business area from the home page
SR_SET_HOME_PAGE_REPORTS	Read access to this key allows the user to select and set their Home Page reports
SR_PERSONAL_AGING	Read access determines access to save personal aging reports and to place personally created aging report on the user's home page
SR_PERSONAL_CHART	Read access determines access to save personal charts and to place personally created charts on the user's home page
SR_PERSONAL_COLUMN_REPORT	Determines the ability to create and run personal reports. Both Read and Write access can be used to determine the functionality required
SR_PERSONAL_GEOSPATIAL	Read access for this key gives the ability to view and run personal Geospatial reports. Write access gives the ability to create new personal Geospatial reports
SR_PERSONAL_SUMMARY_REPORT	Determines the ability to create and run personal summary reports. Both Read and Write access can be used to determine the functionality required
SR_PUBLIC_AGING	Read access determines access to save public aging reports and to place public aging reports on the user's home page
SR_PUBLIC_CHART	Read access determines access to save public charts and to place public charts on the user's home page
SR_PUBLIC_GEOSPATIAL	Read access for this key gives the ability to view and run public Geospatial reports. Write access gives the ability to create new personal Geospatial reports
SR_PUBLIC_SUMMARY_REPORT	Determines the ability to create and run public summary reports. Both Read and Write access can be used to determine the functionality required
SR_PUBLIC_WORKSPACE	This permission key controls end user access to view and select from the list of public workspaces to which the user has permission. Typically, this means the user must have access to the business area, project and user role in which the workspace is saved
SR_PERSONAL_LINKED_REPORT	This key controls access to the linked report button on the Search screen
SR_PUBLIC_LINKED_REPORT	Read access determines access to the linked report feature
SR_PUBLIC_COLUMN_REPORT	This key controls access to public reports. Read access gives the ability to run public reports, Write access gives the ability to create public reports
SR_RELATIONSHIP_GROUP	This key controls access to the Group Issues button on column reports and Quicklist reports. Write access is needed to control this
SR_SAVE_PUBLIC_FILTERS	This provides the ability to allow the user role to save and delete shared report filters on the Query screen and report editors
SR_USERROLE_AGING	Read access for this key gives the ability to see and run aging reports created for the current user role. Write access gives the ability to create these aging reports
SR_USERROLE_CHART	Read access for this key gives the ability to see and run charts created for the current user role. Write access gives the ability to create these charts
SR_USERROLE_COLUMN_REPORT	Read access for this key gives the ability to see and run reports created for the current user role. Write access gives the ability to create these reports
SR_USERROLE_GEOSPATIAL	Read access for this key gives the ability to view and run public Geospatial reports for the current user role. Write access gives the ability to create new Geospatial reports for the current user role
SR_USERROLE_SUMMARY_REPORT	Determines the ability to create and run summary reports for the current user role. Both Read and Write access can be used to determine the functionality required

Language and Locale Codes

The following is a list of language and locales supported by Java, and therefore by ExtraView.

Language	Country	Locale ID
Arabic	Saudi Arabia	ar_SA
Chinese (Simplified)	China	zh_CN
Chinese (Traditional)	Taiwan	zh_TW
Dutch	Netherlands	nl_NL
English	Australia	en_AU
English	Canada	en_CA
English	United Kingdom	en_GB
English	United States	en_US
French	Canada	fr_CA
French	France	fr_FR
German	Germany	de_DE
Hebrew	Israel	iw_IL
Hindi	India	hi_IN
Italian	Italy	it_IT
Japanese	Japan	ja_JP
Korean	South Korea	ko_KR
Portuguese	Brazil	pt_BR
Spanish	Spain	es_ES
Swedish	Sweden	sv_SE
Thai (Western digits)	Thailand	th_TH
Thai (Thai digits)	Thailand	th_TH_TH

The following list is provided, and believed to work, but is not part of the testing done by Sun with their Java releases.

Language	Country	Locale ID
Albanian	Albania	sq_AL
Arabic	Algeria	ar_DZ
Arabic	Bahrain	ar_BH
Arabic	Egypt	ar_EG
Arabic	Iraq	ar_IQ
Arabic	Jordan	ar_JO
Arabic	Kuwait	ar_KW
Arabic	Lebanon	ar_LB
Arabic	Libya	ar_LY
Arabic	Morocco	ar_MA
Arabic	Oman	ar_OM

Arabic	Qatar	ar_QA
Arabic	Sudan	
Arabic	ar_SD	
Arabic	Syria	ar_SY
Arabic	Tunisia	ar_TN
Arabic	United Arab Emirates	ar_AE
Arabic	Yemen	ar_YE
Bulgarian	Bulgaria	bg_BG
Belorussian	Belorussia	be_BY
Catalan	Spain	ca_ES
Chinese	Hong Kong	zh_HK
Croatian	Croatia	hr_HR
Czech	Czech Republic	cs_CZ
Danish	Denmark	da_DK
Dutch	Belgium	nl_BE
English	India	en_IN
English	Ireland	en_IE
English	New Zealand	en_NZ
English	South Africa	en_ZA
Estonian	Estonia	et_EE
Finnish	Finland	fi_FI
French	Belgium	fr_BE
French	Luxembourg	fr_LU
French	Switzerland	fr_CH
German	Austria	de_AT
German	Luxembourg	de_LU
German	Switzerland	de_CH
Greek	Greece	el_GR
Hungarian	Hungary	hu_HU
Icelandic	Iceland	is_IS
Italian	Switzerland	it_CH
Latvian	Latvia	lv_LV
Lithuanian	Lithuania	lt_LT
Macedonian	Macedonia	mk_MK
Norwegian (Bokmål)	Norway	no_NO
Norwegian (Nynorsk)	Norway	no_NO_NY
Polish	Poland	pl_PL
Portuguese	Portugal	pt_PT
Romanian	Romania	ro_RO

Russian	Russia	ru_RU
Serbian (Cyrillic)	Yugoslavia	sr_YU
Serbo-Croatian	Yugoslavia	sh_YU
Slovak	Slovakia	sk_SK
Slovenian	Slovenia	sl_SI
Spanish	Argentina	es_AR
Spanish	Bolivia	es_BO
Spanish	Chile	es_CL
Spanish	Colombia	es_CO
Spanish	Costa Rica	es_CR
Spanish	Dominican Republic	es_DO
Spanish	Ecuador	es_EC
Spanish	El Salvador	es_SV
Spanish	Guatemala	es_GT
Spanish	Honduras	es_HN
Spanish	Mexico	es_MX
Spanish	Nicaragua	es_NI
Spanish	Panama	es_PA
Spanish	Paraguay	es_PY
Spanish	Peru	es_PE
Spanish	Puerto Rico	es_PR
Spanish	Uruguay	es_UY
Spanish	Venezuela	es_VE
Turkish	Turkey	tr_TR
Ukrainian	Ukraine	uk_UA

Time Zones

Offset to GMT	Time Zone	Offset to GMT	Time Zone
GMT -12:00	Etc/GMT +12	GMT +1:00	Africa/Ceuta
GMT -11:00	Etc/GMT+11		Africa/Douala
	MIT		Africa/Kinshasa
	Pacific/Apia		Africa/Lagos
	Pacific/Midway		Africa/Libreville
	Pacific/Niue		Africa/Luanda
	Pacific/Pago_Pago		Africa/Malabo
	Pacific/Samoa		Africa/Ndjamena
	US/Samoa		Africa/Niamey
GMT -10:00	America/Adak		Africa/Porto-Novo

	America/Atka		Africa/Tunis
	Etc/GMT+10		Africa/Windhoek
	HST		Arctic/Longyearbyen
	Pacific/Fakaofo		Atlantic/Jan_Mayen
	Pacific/Honolulu		CET
	Pacific/Johnston		ECT
	Pacific/Rarotonga		Etc/GMT-1
	Pacific/Tahiti		Europe/Amsterdam
	SystemV/HST10		Europe/Andorra
	US/Aleutian		Europe/Belgrade
	US/Hawaii		Europe/Berlin
GMT -9:30	Pacific/Marquesas		Europe/Bratislava
	AST		Europe/Brussels
	America/Anchorage		Europe/Budapest
	America/Juneau		Europe/Copenhagen
	America/Nome		Europe/Gibraltar
	America/Yakutat		Europe/Ljubljana
GMT -9:00	Etc/GMT+9		Europe/Luxembourg
	Pacific/Gambier		Europe/Madrid
	SystemV/YST9		Europe/Malta
	SystemV/YST9YDT		Europe/Monaco
	US/Alaska		Europe/Oslo
GMT -8:00	America/Dawson		Europe/Paris
	America/Ensenada		Europe/Prague
	America/Los_Angeles		Europe/Rome
	America/Tijuana		Europe/San_Marino
	America/Vancouver		Europe/Sarajevo
	America/Whitehorse		Europe/Skopje
	Canada/Pacific		Europe/Stockholm
	Canada/Yukon		Europe/Tirane
	Etc/GMT+8		Europe/Vaduz
	Mexico/BajaNorte		Europe/Vatican
	PST		Europe/Vienna
	PST8PDT		Europe/Warsaw
	Pacific/Pitcairn		Europe/Zagreb
	SystemV/PST8		Europe/Zurich
	SystemV/PST8PDT		MET
	US/Pacific		Poland
	US/Pacific-New	GMT +2:00	ART

GMT -7:00	America/Boise	Africa/Blantyre
	America/Cambridge_Bay	Africa/Bujumbura
	America/Chihuahua	Africa/Cairo
	America/Dawson_Creek	Africa/Gaborone
	America/Denver	Africa/Harare
	America/Edmonton	Africa/Johannesburg
	America/Hermosillo	Africa/Kigali
	America/Inuvik	Africa/Lubumbashi
	America/Mazatlan	Africa/Lusaka
	America/Phoenix	Africa/Maputo
	America/Shiprock	Africa/Maseru
	America/Yellowknife	Africa/Mbabane
	Canada/Mountain	Africa/Tripoli
	Etc/GMT+7	Asia/Amman
	MST	Asia/Beirut
	MST7MDT	Asia/Damascus
	Mexico/BajaSur	Asia/Gaza
	Navajo	Asia/Istanbul
	PNT	Asia/Jerusalem
	SystemV/MST7	Asia/Nicosia
	SystemV/MST7MDT	Asia/Tel_Aviv
	US/Arizona	CAT
	US/Mountain	EET
GMT -6:00	America/Belize	Egypt
	America/Cancun	Etc/GMT-2
	America/Chicago	Europe/Athens
	America/Costa_Rica	Europe/Bucharest
	America/El_Salvador	Europe/Chisinau
	America/Guatemala	Europe/Helsinki
	America/Managua	Europe/Istanbul
	America/Menominee	Europe/Kaliningrad
	America/Merida	Europe/Kiev
	America/Mexico_City	Europe/Minsk
	America/Monterrey	Europe/Nicosia
	America/North_Dakota/Center	Europe/Riga
	America/Rainy_River	Europe/Simferopol
	America/Rankin_Inlet	Europe/Sofia
	America/Regina	Europe/Tallinn
	America/Swift_Current	Europe/Tiraspol

	America/Tegucigalpa		Europe/Uzhgorod
	America/Winnipeg		Europe/Vilnius
	CST		Europe/Zaporozhye
	CST6CDT		Israel
	Canada/Central		Libya
	Canada/East-Saskatchewan		Turkey
	Canada/Saskatchewan	GMT +3:00	Africa/Addis_Ababa
	Chile/EasterIsland		Africa/Asmera
	Etc/GMT+6		Africa/Dar_es_Salaam
	Pacific/Easter		Africa/Djibouti
	Pacific/Galapagos		Africa/Kampala
	SystemV/CST6		Africa/Khartoum
	Mexico/General		Africa/Mogadishu
	SystemV/CST6CDT		Africa/Nairobi
	US/Central		Antarctica/Syowa
GMT -5:00	America/Bogota		Asia/Aden
	America/Cayman		Asia/Baghdad
	America/Detroit		Asia/Bahrain
	America/Eirunepe		Asia/Kuwait
	America/Fort_Wayne		Asia/Qatar
	America/Grand_Turk		Asia/Riyadh
	America/Guayaquil		EAT
	America/Havana		Etc/GMT-3
	America/Indiana/Indianapolis		Europe/Moscow
	America/Indiana/Knox		Indian/Antananarivo
	America/Indiana/Marengo		Indian/Comoro
	America/Indiana/Vevay		Indian/Mayotte
	America/Indianapolis		W-SU
	America/Iqaluit	GMT +3:30	Asia/Tehran
	America/Jamaica		Iran
	America/Kentucky/Louisville	GMT +4:00	Asia/Aqtau
	America/Kentucky/Monticello		Asia/Baku
	America/Knox_IN		Asia/Dubai
	America/Lima		Asia/Muscat
	America/Louisville		Asia/Oral
	America/Montreal		Asia/Tbilisi
	America/Nassau		Asia/Yerevan
	America/New_York		Etc/GMT-4
	America/Nipigon		Europe/Samara

	America/Panama	Indian/Mahe
	America/Pangnirtung	Indian/Mauritius
	America/Port-au-Prince	Indian/Reunion
	America/Porto_Acre	NET
	America/Rio_Branco	GMT +4:30 Asia/Kabul
	America/Thunder_Bay	GMT +5:00 Asia/Aqtobe
	Brazil/Acre	Asia/Ashgabat
	Canada/Eastern	Asia/Ashkhabad
	Cuba	Asia/Bishkek
	EST	Asia/Dushanbe
	EST5EDT	Asia/Karachi
	Etc/GMT+5	Asia/Samarkand
	Jamaica	Asia/Tashkent
	SystemV/EST5	Asia/Yekaterinburg
	SystemV/EST5EDT	Etc/GMT-5
	US/East-Indiana	Indian/Kerguelen
	US/Eastern	Indian/Maldives
	US/Indiana-Starke	PLT
	US/Michigan	GMT +5:30 Asia/Calcutta
GMT -4:00	America/Anguilla	IST
	America/Antigua	GMT +6:00 Antarctica/Mawson
	America/Aruba	Antarctica/Vostok
	America/Asuncion	Asia/Almaty
	America/Barbados	Asia/Colombo
	America/Boa_Vista	Asia/Dacca
	America/Caracas	Asia/Dhaka
	America/Cuiaba	Asia/Novosibirsk
	America/Curacao	Asia/Omsk
	America/Dominica	Asia/Qyzylorda
	America/Glace_Bay	Asia/Thimbu
	America/Goose_Bay	Asia/Thimphu
	America/Grenada	BST
	America/Guadeloupe	Etc/GMT-6
	America/Guyana	Indian/Chagos
	America/Halifax	GMT +6:30 Asia/Rangoon
	America/La_Paz	Indian/Cocos
	America/Manaus	GMT +7:00
	America/Martinique	Antarctica/Davis
	America/Montserrat	Asia/Bangkok

	America/Port_of_Spain	Asia/Hovd
	America/Porto_Velho	Asia/Jakarta
	America/Puerto_Rico	Asia/Krasnoyarsk
	America/Santiago	Asia/Phnom_Penh
	America/Santo_Domingo	Asia/Pontianak
	America/St_Kitts	Asia/Saigon
	America/St_Lucia	Asia/Vientiane
	America/St_Thomas	Etc/GMT-7
	America/St_Vincent	Indian/Christmas
	America/Thule	VST
	America/Tortola	GMT +8:00 Antarctica/Casey
	America/Virgin	Asia/Brunei
	Antarctica/Palmer	Asia/Chongqing
	Atlantic/Bermuda	Asia/Chungking
	Atlantic/Stanley	Asia/Harbin
	Brazil/West	Asia/Hong_Kong
	Canada/Atlantic	Asia/Irkutsk
	Chile/Continental	Asia/Kashgar
	Etc/GMT+4	Asia/Kuala_Lumpur
	PRT	Asia/Kuching
	SystemV/AST4	Asia/Macao
	SystemV/AST4ADT	Asia/Macau
GMT -3:30	America/St_Johns	Asia/Makassar
	CNT	Asia/Manila
	Canada/Newfoundland	Asia/Shanghai
GMT -3:00	AGT	Asia/Singapore
	America/Araguaina	Asia/Taipei
	America/Belem	Asia/Ujung_Pandang
	America/Buenos_Aires	Asia/Ulaanbaatar
	America/Catamarca	Asia/Ulan_Bator
	America/Cayenne	Asia/Urumqi
	America/Cordoba	Australia/Perth
	America/Fortaleza	Australia/West
	America/Godthab	CTT
	America/Jujuy	Etc/GMT-8
	America/Maceio	Hongkong
	America/Mendoza	PRC
	America/Miquelon	Singapore
	America/Montevideo	GMT +9:00 Asia/Choibalsan

	America/Paramaribo		Asia/Dili
	America/Recife		Asia/Jayapura
	America/Rosario		Asia/Pyongyang
	America/Sao_Paulo		Asia/Seoul
	Antarctica/Rothera		Asia/Tokyo
	BET		Asia/Yakutsk
	Brazil/East		JST
	Etc/GMT+3		Japan
GMT -2:00	America/Noronha		Etc/GMT-9
	Atlantic/South_Georgia		Pacific/Palau
	Brazil/DeNoronha		ROK
	Etc/GMT+2	GMT +9:30	ACT
GMT -1:00	America/Scoresbysund		Australia/Adelaide
	Atlantic/Azores		Australia/Broken_Hill
	Atlantic/Cape_Verde		Australia/Darwin
	Etc/GMT+1		Australia/North
GMT +0:00	Africa/Abidjan		Australia/South
	Africa/Accra		Australia/Yancowinna
	Africa/Bamako	GMT +10:00	AET
	Africa/Banjul		Antarctica/DumontDUrville
	Africa/Bissau		Asia/Sakhalin
	Africa/Casablanca		Asia/Vladivostok
	Africa/Conakry		Australia/ACT
	Africa/Dakar		Australia/Brisbane
	Africa/El_Aaiun		Australia/Canberra
	Africa/Freetown		Australia/Hobart
	Africa/Lome		Australia/Lindeman
	Africa/Monrovia		Australia/Melbourne
	Africa/Nouakchott		Australia/NSW
	Africa/Ouagadougou		Australia/Queensland
	Africa/Sao_Tome		Australia/Sydney
	Africa/Timbuktu		Australia/Tasmania
	America/Danmarkshavn		Australia/Victoria
	Atlantic/Canary		Etc/GMT-10
	Atlantic/Faeroe		Pacific/Guam
	Atlantic/Madeira		Pacific/Port_Moresby
	Atlantic/Reykjavik		Pacific/Saipan
	Atlantic/St_Helena		Pacific/Truk
	Eire		Pacific/Yap

		GMT +10:30	Australia/LHI
	Etc/GMT		
	Etc/GMT+0		Australia/Lord_Howe
	Etc/GMT-0	GMT +11:00	Asia/Magadan
	Etc/GMT0		Etc/GMT-11
	Etc/Greenwich		Pacific/Efate
	Etc/UCT		Pacific/Guadalcanal
	Etc/UTC		Pacific/Kosrae
	Etc/Universal		Pacific/Noumea
	Etc/Zulu		Pacific/Ponape
	Europe/Belfast		SST
	Europe/Dublin	GMT +11:30	Pacific/Norfolk
	Europe/Lisbon	GMT +12:00	Antarctica/McMurdo
	Europe/London		Antarctica/South_Pole
	GB		Asia/Anadyr
	GB-Eire		Asia/Kamchatka
	GMT		Etc/GMT-12
	GMT0		Kwajalein
	Greenwich		NST
	Iceland		NZ
	Portugal		Pacific/Auckland
	UCT		Pacific/Fiji
	UTC		Pacific/Funafuti
	Universal		Pacific/Kwajalein
	WET		Pacific/Majuro
	Zulu		Pacific/Nauru
GMT +1:00	Africa/Algiers		Pacific/Tarawa
	Africa/Bangui		Pacific/Wake
	Africa/Brazzaville		Pacific/Wallis

Character Set Values

The following table represents the ISO Latin 1 Character Entities as defined by the ASCII-ISO 8859-1 standard. Note that not all browsers can display all these characters accurately.

Decimal	Symbol	HTML Code	Decimal	Symbol	HTML Code
32		&#32	144	□	&144
33	!	&#33	145	'	&145
34	"	&#34	146	'	&146
35	#	&#35	147	"	&147
36	$	&#36	148	"	&148

37	%	&#37	149	•	&149
38	&	&#38	150	–	&150
39	'	&#39	151	—	&#151
40	(	&#40	152	˜	&#152
41	)	&#41	153	™	&#153
42	*	&#42	154	š	&#154
43	+	&#43	155	›	&#155
44	,	&#44	156	œ	&#156
45	-	&#45	157	□	&#157
46	.	&#46	158	ž	&#158
47	/	&#47	159	Ÿ	&#159
48	0	&#48	160		&#160
49	1	&#49	161	¡	&#161
50	2	&#50	162	¢	&#162
51	3	&#51	163	£	&#163
52	4	&#52	164	¤	&#164
53	5	&#53	165	¥	&#165
54	6	&#54	166	¦	&#166
55	7	&#55	167	§	&#167
56	8	&#56	168	¨	&#168
57	9	&#57	169	©	&#169
58	:	&#58	170	ª	&#170
59	;	&#59	171	«	&#171
60	<	&#60	172	¬	&#172
61	=	&#61	173		&#173
62	>	&#62	174	®	&#174
63	?	&#63	175	¯	&#175
64	@	&#64	176	°	&#176
65	A	&#65	177	±	&#177
66	B	&#66	178	²	&#178
67	C	&#67	179	³	&#179
68	D	&#68	180	´	&#180
69	E	&#69	181	µ	&#181
70	F	&#70	182	¶	&#182
71	G	&#71	183	•	&#183
72	H	&#72	184	¸	&#184
73	I	&#73	185	¹	&#185
74	J	&#74	186	º	&#186
75	K	&#75	187	»	&#187

76	L	&#76	188	¼	&#188
77	M	&#77	189	½	&#189
78	N	&#78	190	¾	&#190
79	O	&#79	191	¿	&#191
80	P	&#80	192	À	&#192
81	Q	&#81	193	Á	&#193
82	R	&#82	194	Â	&#194
83	S	&#83	195	Ã	&#195
84	T	&#84	196	Ä	&#196
85	U	&#85	197	Å	&#197
86	V	&#86	198	Æ	&#198
87	W	&#87	199	Ç	&#199
88	X	&#88	200	È	&#200
89	Y	&#89	201	É	&#201
90	Z	&#90	202	Ê	&#202
91	[	&#91	203	Ë	&#203
92	\	&#92	204	Ì	&#204
93	]	&#93	205	Í	&#205
94	^	&#94	206	Î	&#206
95	_	&#95	207	Ï	&#207
96	`	&#96	208	Ð	&#208
97	a	&#97	209	Ñ	&#209
98	b	&#98	210	Ò	&#210
99	c	&#99	211	Ó	&#211
100	d	&#100	212	Ô	&#212
101	e	&#101	213	Õ	&#213
102	f	&#102	214	Ö	&#214
103	g	&#103	215	×	&#215
104	h	&#104	216	Ø	&#216
105	i	&#105	217	Ù	&#217
106	j	&#106	218	Ú	&#218
107	k	&#107	219	Û	&#219
108	l	&#108	220	Ü	&#220
109	m	&#109	221	Ý	&#221
110	n	&#110	222	Þ	&#222
111	o	&#111	223	ß	&#223
112	p	&#112	224	à	&#224
113	q	&#113	225	á	&#225
114	r	&#114	226	â	&#226

115	s	&#115	227	ã	&#227
116	t	&#116	228	ä	&#228
117	u	&#117	229	å	&#229
118	v	&#118	230	æ	&#230
119	w	&#119	231	ç	&#231
120	x	&#120	232	è	&#232
121	y	&#121	233	é	&#233
122	z	&#122	234	ê	&#234
123	{	&#123	235	ë	&#235
124	|	&#124	236	ì	&#236
125	}	&#125	237	í	&#237
126	~	&#126	238	î	&#238
127		&#127	239	ï	&#239
128	€	&#128	240	ð	&#240
129	□	&#129	241	ñ	&#241
130	,	&#130	242	ò	&#242
131	ƒ	&#131	243	ó	&#243
132	„	&#132	244	ô	&#244
133	…	&#133	245	õ	&#245
134	†	&#134	246	ö	&#246
135	‡	&#135	247	÷	&#247
136	ˆ	&#136	248	ø	&#248
137	‰	&#137	249	ù	&#249
138	Š	&#138	250	ú	&#250
139	‹	&#139	251	û	&#251
140	Œ	&#140	252	ü	&#252
141	□	&#141	253	ý	&#253
142	Ž	&#142	254	þ	&#254
143	□	&#143			

The HTML Area Utility

The HTML Area is available to as an editor for HTML, allowing you to enter rich text or HTML without knowing the HTML language. It offers many features similar to a word processor, but it is not a full word processor, die to the limitations of internet browsers. Advanced users may switch to edit HTML directly. HTML Area fields may contain images which are pasted into the field from the user's clipboard. Likewise, entire files may be embedded. Embedded files are represented by an icon, once they have been placed within the field.

Your administrator will have selected one of three available toolbars or might have customized a toolbar solely for your company's use. The built-in toolbars are shown in the following screenshots. The HTML Area utility makes use of scripts that run within the browser, and according to the security settings of the browser, you may need to acknowledge that it is OK to run these scripts. If you are using a mobile client, only the basic toolbar is initialized, to conserve space on the screen.

Basic Toolbar

Standard Toolbar
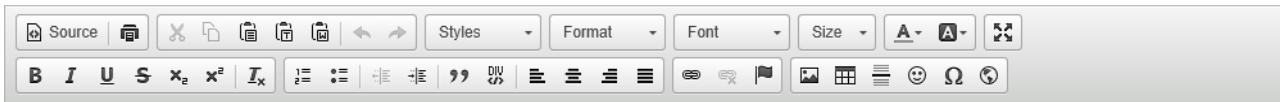
Full Toolbar
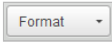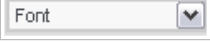
The use of the HTML Area field is intuitive. Just enter text and use the buttons on the toolbar to provide the formatting.

Note that there are some limitations when entering and updating text within the utility. A key limitation is that you cannot click on a link when the field is being updated. This functionality only works when the field is in read-only mode.

The available toolbar buttons are:

Basic	Standard	Full	Button	Function
✓	✓	✓	**B**	Toggle bold on and off for the selected text
✓	✓	✓	*I*	Toggle italic text on and off for the selected text
✓	✓	✓	U	Toggle underlined text on and off for the selected text
✓	✓	✓	I_x	Remove text formatting from the selected text
✓	✓	✓	≔	Toggle a numbered list for the selected text
✓	✓	✓	≔	Togle a bulleted list for the selected text
✓	✓	✓	∞	Add a link to the selected text
✓	✓	✓	∞x	Remove the link from the selected text
✓	✓	✓	🖶	Print the contents of the HTML Area
	✓	✓	Source	Toggle the display mode between "what-you-see-is-what-you-get" and HTML source views
	✓	✓	✂	Cut the selected text to the clipboard
	✓	✓	🗐	Copy the selected text to the clipboard
	✓	✓	📋	Paste the text from the clipboard to the cursor position
	✓	✓	📋	Paste the text, as plain unformatted text, from the clipboard to the cursor position
	✓	✓	📋	Paste text from Microsoft Office documents - this will remove the obscure formatting that Microsoft Office documents often contain
	✓	✓	↶	Undo the last operation
	✓	✓	↷	Redo the last operation that was undone

Basic	Standard	Full	Button	Function
	✓	✓	Styles ▾	Use a style from the list. Note that there are interdependencies with the Style, Format, Font and Size lists. Not all combinations work with all computers, as your browser, or computer operating system may not support the combination you choose
	✓	✓	Format ▾	Format the text from the selections within the list. Note that there are interdependencies with the Style, Format, Font and Size lists. Not all combinations work with all computers, as your browser, or computer operating system may not support the combination you choose
	✓	✓	Font ▾	Select a font from the selections within the list. Note that there are interdependencies with the Style, Format, Font and Size lists. Not all combinations work with all computers, as your browser, or computer operating system may not support the combination you choose
	✓	✓	5 ▾	Set a size for the text from the selections within the list. Note that there are interdependencies with the Style, Format, Font and Size lists. Not all combinations work with all computers, as your browser, or computer operating system may not support the combination you choose
	✓	✓	A	Set a color for the selected text
	✓	✓	A	Set a background color for the selected text
	✓	✓	⤢	Mximize / Restore the size of the HTML Area within the browser
	✓	✓	S	Strikeout the selected text
	✓	✓	x₂	Subscript the selected text
	✓	✓	x²	Superscript the selected text
	✓	✓	⇤	Outdent the selected text
	✓	✓	⇥	Indent the selected text
	✓	✓	❞	Place the selected text in blockquotes
	✓	✓	DIV	Place the selected text in an HTML DIV
	✓	✓	≡	Left-justify the selected text
	✓	✓	≡	Center-justify the selected text
	✓	✓	≡	Right-justify the selected text
	✓	✓	≡	Full-justify the selected text
	✓	✓	⚑	Insert an HTML text anchor tag
	✓	✓	🖼	Insert an image
	✓	✓	⊞	Insert a table
	✓	✓	≡	Insert a horizontal rule
	✓	✓	☺	Insert a smiley face
	✓	✓	Ω	Insert a character symbol
	✓	✓	🌐	Insert an HTML Iframe
		✓	🔍	Search for text within the HTML Area field
		✓	ᵇₐ	Search and replace text within the HTML Area field

Basic	Standard	Full	Button	Function
		✓		Select all the text within the HTML Area field
		✓		Text direction goes from left-to-right (the default)
		✓		Text direction goes from right-to-left
		✓		Insert a page-break for printing

Spell Checker

The ExtraView Spell Checker is an alternative to the spell checking built into most browsers. It offers these advantages:

- Consistency of use for all your users
- A central dictionary with the same vocabulary for all users
- Many languages and variants
  - US English
  - UK English
  - Canadian English
  - Australian English
  - Spanish
  - French
  - Dutch
  - English Medical Dictionary
- The ability to add words to the central dictionaries

The dictionary is enabled with the behavior setting named ENABLE_SPELLCHECK. When this is turned on, all text area, log area, print area and text fields are spell checked as the user types, when the user enters or modifies data within them. You can also spell check HTML area fields, by clicking on the spell check button on the toolbar. Spell checking as the user types is not supported in HTML Area fields.

There are occasions when you might not want to spell check a field. For example, if the field contains an account number, or some arbitrary text that never spells a word, you might want to skip the spell check for that field. This is achieved by adding an HTML Modifier layout cell attribute to the field on its layout within the Design Center. The HTML Modifier is:

```
nospell=true
```

Version Limitations

The 3 ExtraView versions have increasing levels of features and sophistication as shown in the following table. There are no limitations in the **Enterprise** version.

Feature	Standard	Professional	Enterprise
General			
Number of Business Areas supported	1	3	Unlimited
Number of Projects supported	1	2 per Area	Unlimited
Number of user roles	3	5	Unlimited
Number of user-defined fields	25	150	Unlimited
Number of public reports	5	15	Unlimited
Number of private reports	5 per user	10 per user	Unlimited
Total number of issues in database	5,000	15,000	Unlimited
Support for multiple languages	No	No	Yes
Support for multiple time zones	No	No	Yes

Feature	Standard	Professional	Enterprise
Support fo repeating rows	No	Yes	Yes
Support for relationships	No	No	Yes
Workspaces	No	No	Yes
LDAP connections	No	Yes	Yes
SSO connections	No	No	Yes
RESTful API	No	No	Yes
Command Line Interface	No	No	Yes
Web Services Interface	No	No	Yes
Custom Programming Interfaces	No	No	Yes
Interfaces to 3rd party products	No	No	Yes

User-Defined Field Types

Feature	Standard	Professional	Enterprise
Checkbox	No	No	Yes
Currency	No	No	Yes
Custom	No	Yes	Yes
Date	Yes	Yes	Yes
Day	No	No	Yes
Decimal	No	Yes	Yes
Document	No	No	Yes
HTML	No	Yes	Yes
Image	No	No	Yes
Label	No	Yes	Yes
List	Yes	Yes	Yes
Log Area	Yes	Yes	Yes
Number	No	Yes	Yes
Popup	No	No	Yes
Print Text	No	No	Yes
Radio Button	No	No	Yes
Tab	No	Yes	Yes
Text Area	Yes	Yes	Yes
Text	Yes	Yes	Yes
User	No	Yes	Yes

Reporting Features

Feature	Standard	Professional	Enterprise
Column Reports	Yes	Yes	Yes
Summary Reports	Yes	Yes	Yes
Matrix Reports	Yes	Yes	Yes
Charts	No	Yes	Yes

Feature	Standard	Professional	Enterprise
Dashboard Reports	No	Yes	Yes
Calendar Reports	No	Yes	Yes
Aging Reports	No	No	Yes
Taskboard Reports	No	No	Yes
Planning Reports	No	No	Yes
Container Reports	No	No	Yes
Custom URL Reports	No	No	Yes

Administrative Capabilities

User Account Maintenance	Yes	Yes	Yes
Manage List Values	Yes	Yes	Yes
Email Templates	Yes	Yes	Yes
Task Manager	Yes	Yes	Yes
Design Center	Yes	Yes	Yes
Data Dictionary	Yes	Yes	Yes
Behavior Settings	Yes	Yes	Yes
User Sign On Log	No	No	Yes
System Log	No	No	Yes
End-User Session Management	No	No	Yes
Disable and Enable User Access	No	No	Yes
Manage Scheduled Reports	No	No	Yes
Administrative Report Management	No	No	Yes
Manage User Groups	No	No	Yes
Interest Lists	No	No	Yes
Escalation	No	No	Yes
Business Calendars	No	No	Yes
Business Rules	No	No	Yes
Security Permissions	No	Limited	Yes
Allowed Values	No	Yes	Yes
Layout Types	No	No	Yes
Status Change Rules	No	No	Yes
Relationship Group Maintenance	No	No	Yes
Service Level Agreement Maintenance	No	No	Yes
Reporting Hierarchies	No	No	Yes
Privacy Groups	No	No	Yes
User Role Maintenance	Yes	Yes	Yes
Adobe PDF Report Mapping	No	No	Yes
Navigation Bar Maintenance	Yes	Yes	Yes

Feature	Standard	Professional	Enterprise
Sign On Message	Yes	Yes	Yes
Upload Company Logo	Yes	Yes	Yes
User Interface Themes	No	Yes	Yes
Metadata Import / Export	No	No	Yes
File Import - Issue Data	No	Yes	Yes
File Import - User Information	No	No	Yes
File Import - Data Dictionary	No	No	Yes
Allowed Locale Maintenance	No	No	Yes
Message Translation	No	No	Yes
Server Side Templates	No	No	Yes
Admin Statistics	Yes	Yes	Yes
System Log Type Maintenance	No	No	Yes
Start Page Maintenance	No	No	Yes
Pre-cache Layouts	No	No	Yes
Edit Administration Tab Labels	No	No	Yes
Encryption Key Management	No	No	Yes

Attachment Migration

This appendix describes an ExtraView command-line utility that moves file attachments from internal storage within the database, to the file system.

By default, file attachments uploaded to ExtraView are stored as binary objects in the database. The ExtraView Administrator has the choice to configure the system to store some or all of the uploaded attachments to a file system accessible by the application server.

Within ExtraView, the program keeps track of where all particular attachment are located. Prior to the conversion, all attachments are marked as being stored "internally" – within the database. As part of the migration process, as each individual file is extracted and copied to the final location on the external file system, the system marks that attachment as now being stored "externally". This allows the migration process to be done as a background task, without impacting the availability of ExtraView or its attachments to the end users.

This provides flexibility – the Administrator can choose to have all files stored internally in the database, all files stored externally in the file system, or have a mixture of some file types stored within the database, and some file types stored on the file system.

Configuring External Storage

Note: It is strongly recommend that you back up your database prior to performing any of the procedures described below.

To configure attachment storage options, log in to ExtraView with the Administrator role. Click on the **Admin** navigation button, then the **System Controls** tab, then select **All Behavior Settings**, and edit the following settings:

- **ATTACHMENT_REPOSITORY_OPT** - This setting controls whether attachments, document field and image fields are stored internally within the database, externally on the file system, or in some combination of the two methods.

  If the value of this setting is **INTERNAL** (the default value), then all attachments, document fields and image fields are stored internally within the database. If the value is set to **EXTERNAL**, then all the attachments, document fields and image fields are stored on the external file system.

Alternatively, you may provide a comma separated list of file extensions and then all files with these extensions are stored in the database while all others are stored externally in the file repository.

For example, if you set the value of this setting to **txt, doc, docx, htm, html** then files of these types will be stored within the database. This strategy leaves the files stored internally in the database as searchable by keywords, while image and video files with extensions such as **jpg** and **mp4** are stored externally.

Regardless of the setting, ExtraView will continue to store dynamically generated thumbnail-sized image files in the database as these are relatively small files and for performance reasons must be more readily available.

- **ATTACHMENT_REPOSITORY_ROOT** – This setting defines the location outside the database where attachments, document fields and image fields will be stored. You must ensure that the path is valid from the application server(s) that are running ExtraView and that you have all the permissions to read and write to the storage. In a clustered or load-balanced environment, you must ensure that all instances of ExtraView can read and write to the same file system using this path.

  Please note – if you change this setting, new attachments, document field and image fields will be saved to the new location, and existing attachments, document fields and image fields will still reference the old location. You will need to ensure that the previous file path is still accessible by the ExtraView instance, or these existing attachments will not be available to end users.

  Set up a separate backup method for this external storage as backing up your database will no longer backup the file attachments

- **ATTACHMENT_REPOSITORY_DMAX** - This setting defines the maximum number of files or directories that will be created under one node of the external directory structure. The default for this value is 999. It is not likely that this value needs to be altered.

Sizing the file system

You should ensure that there is adequate space for future growth when sizing the file system to be used for storing attachments externally. To do this, have your DBA connect to your ExtraView database using Query Analyzer (MSSQL) or SQL Plus (Oracle) and run the following queries:

-- bring back the total size of the attachments stored in the database, in bytes.
select sum(file_size) from attachment;
-- bring back the number of attachments in the database
select count(*) from attachment;

Migration Utility Options

There are several forms of the doAttachmentMigration command. The utility is found in the directory named WEB-INF/data. All require one or more parameters passed as part of the ExtraView command line utility:

- Move a single attachment by providing a single attachment ID:

  doAttachmentMigration instanceName -id 10000

- Move a batch of attachments by providing a range of attachment IDs:

  doAttachmentMigration instanceName -from 10000 -to 50000

- Move attachments according to their content type:

  doAttachmentMigration instanceName -type *content-type* ...

  The content type may contain valid database wildcard characters. This typically means that a percentage sign (%) will match multiple characters and an underscrore (_) will match a single character. Further, you can specify multiple -type parameters in a single command. For example, the following will migrate all the files that contain **binary** and **zip** within their content types:

  doAttachmentMigration instanceName -type %binary% -type %zip%

- Migrate a list of attachments, from a file containing the list of ID's:

  doAttachmentMigration instanceName -id_list *filename*

The filename contains a list of the IDs that are to be migrated.

Attachment ID values can be found by having your DBA run the following SQL query against the ExtraView database:

select min(attachment_id), max(attachment_id) from attachment;

Optional parameters can be added inside the double quotes:

-report <filename> - Generates a specific location for the report file. By default the report will be dumped into ExtraView log file.

-delete - If specified, deletes the attachment content blob from the database after migration. We would not generally recommend this if you have a large number of attachments, as it is much more efficient to simply have your DBA truncate the table when you have completed the migration and you have validated your results.

-testOnly - This allows a dry run migration of the attachments without any modification to the database or the repository.

-noValidation - This prevents the CRC-32 check on the stored migrated attachments, and will be somewhat faster. Use of this parameter is not recommended, because the extra time to validate the file content vis-à-vis the blob content is nominal, and the consequences of storing a blob in the file system incorrectly due to I/O error or network error could include a loss of valuable data. If the CRC-32 validation fails, the attachment is not migrated and an error is posted to the log and/or report file.

Running the Migration Utility

The migration command line utility is included in the ExtraView application package. It requires Java be installed on the application server to run. We provide shell and batch file scripts that you can use to run the utility.

If you have deployed ExtraView in a WAR/EAR file, you will need to set up an exploded directory to run this script. The scripts are located in the evj/WEB-INF/data folder on your application server.

Microsoft Windows with Apache Tomcat

Edit the doAttachmentMigration.bat file

Locate the lines

set JAVA_HOME=C:\ExtraView\java\jdk1.7.0
set TOMCAT_HOME=C:\ExtraView\Tomcat7.0
set EV_BASE=C:\ExtraView\Tomcat7.0\webapps\evj

and modify the paths to match your installation

Save and exit the edit session.

Open a Command window in the evj\WEB-INF\data folder and type the following command:

doAttachmentMigration instanceName "parameters"

where "parameters" is a double-quote enclosed list of parameters as per the options listed above.

Linux / Unix with Tomcat

Edit the doAttachmentMigration.sh file

Locate the lines

TOMCAT_HOME=/usr/local/extraview/tomcat
JAVA_HOME=/usr/local/extraview/java
EV_BASE = $TOMCAT_HOME/webapps/evj

Set TOMCAT_HOME, JAVA_HOME and EV_BASE to the correct paths for your installation.

Save and exit the edit session.

At the command prompt, type the following command:

sh doAttachmentMigration.sh instanceName "parameters"

where "parameters" is a double-quote enclosed list of parameters as per the options listed above.

For All Platforms

Run the script you just edited. You must provide some additional parameters as shown here:

```
REM required:
REM -from (the starting attachment id for the migration) and -to (the ending attachment
id for the migration)
REM or
REM -id (an attachment id for the migration)
REM or
REM -type content_type [-type content_type2[ ...]] (content type may be a wild card using
 database wild cards -- i.e. '%')
REM or
REM -id_list <file> (the name of a file containing attachment id's - one per line)
```

If you do not know the attachment IDs you may run this query:

```
select min(attachment_id), max(attachment_id) from attachment;
```

You might also simply substitute 1 and a very high number.

Linux / Unix with Weblogic

Edit the doAttachmentMigrationWeblogic.sh file

Locate the lines JAVA_HOME=/usr/local/extraview/java
WL_HOME=/usr/local/bea/weblogic81
EV_BASE=/path/to/extraview/application/evj52-x.jar

Set JAVA_HOME, WL_HOME and EV_BASE to the correct paths for your installation.

Save and exit the edit session.

At the command prompt, type the following command:

sh doAttachmentMigrationWeblogic.sh instanceName "parameters"

where "parameters" is a double-quote enclosed list of parameters as per the options listed above.

Validation and Completion

While the migration utility runs, it will print out information to the console. It will also write detailed log entries to a file in the evj/WEB-INF/log folder or to the file defined in the -report parameter option. Once the process has completed, you can check the log file to determine if there were any problems.

If all attachments were migrated, you can verify that the attachments are available to download and view through the ExtraView web interface. Once you are satisfied, you can have your DBA remove the binary attachments from the database.

If you have selected the option to have all attachments stored in the file system, your DBA can simply run the command

update attachment set thumbnail_id = null where thumbnail_id is not null;
truncate table attachment_content;

Note: this will remove all thumbnails. The thumbnail will be regenerated dynamically the next time the attachment is referenced, e.g., as part of a detailed report.

If you have selected the option of having some types of file stored in the database and some files stored externally, your DBA will need to run some variant of

delete from attachment_content
where attachment_id in
(select attachment_id from attachment where STORED_INTERNAL = 'N');

Note: We strongly recommend that you back up your database prior to performing any of the procedures described above (particularly before deleting any attachments). Please contact support@extraview.com if you would like assistance with this step.

Document & Image Field Migration

This appendix describes an ExtraView command-line utility named DocumentImageMigration that moves fields with the display types of *document* and *image* from internal storage within the database, to the file system. The utility will typically be run with the support of doDocumentMigration.bat (Windows) or doDocumentMigration.sh (Linux).

By default, the contents of document and image fields uploaded to ExtraView are stored as binary objects in the database. The ExtraView Administrator has the choice to configure the system to store the uploaded field contents on the file system, where they will be accessible by the application server.

The program must be run stand-alone, and the ExtraView application server(s) should be stopped while the documents and images are being migrated.

Configuring External Storage

Note: It is strongly recommend that you back up your database prior to performing any of the procedures described below.

To configure the storage options, log in to ExtraView with the Administrator role. These settings are shared with storing file attachments on the file system. Click on the **Admin** navigation button, then the **System Controls** tab, then select **All Behavior Settings**, and edit the following settings:

- **ATTACHMENT_REPOSITORY_OPT** - This setting controls whether attachments, documents and images are stored internally within the database, externally on the file system, or in some combination of the two methods.

  If the value of this setting is **INTERNAL** (the default value), then all attachments, documents and images are stored internally within the database. If the value is set to **EXTERNAL**, then all the attachments, documents and images are stored on the external file system.

  Alternatively, you may provide a comma separated list of file extensions and then all files with these extensions are stored in the database while all others are stored externally in the file repository.

  For example, if you set the value of this setting to **txt, doc, docx, htm, html** then files of these types will be stored within the database. This strategy leaves the files stored internally in the database as searchable by keywords, while image and video files with extensions such as **jpg** and **mp4** are stored externally

- **ATTACHMENT_REPOSITORY_ROOT** – This setting defines the location outside the database where attachments, documents and images will be stored. You must ensure that the path is valid from the application server(s) that are running ExtraView and that you have all the permissions to read and write to the storage. In a clustered or load-balanced environment, you must ensure that all instances of ExtraView can read and write to the same file system using this path.

  Please note – if you change this setting, new attachments will be saved to the new location, and existing attachments will still reference the old location. You will need to ensure that the previous file path is still accessible by the ExtraView instance, or these existing attachments will not be available to end users.

  Set up a separate backup method for this external storage as backing up your database will no longer backup the file attachments

- **ATTACHMENT_REPOSITORY_DMAX** - This setting defines the maximum number of files or directories that will be created under one node of the external directory structure. The default for this value is 999. It is not likely that this value needs to be altered.

Running the Migration Utility

The DocumentImageMigration utility is a Java program. This is run from Windows with the doDocumentMigration.bat file, and from Linux with the doDocumentMigration.sh file.

The migration command line utility is included in the ExtraView application package. It requires Java be installed on the application server to run. We provide shell and batch file scripts that you can use to run the utility.

If you have deployed ExtraView in a WAR/EAR file, you will need to set up an exploded directory to run this script. The scripts are located in the evj/WEB-INF/data folder on your application server. **Syntax:**

doDocumentMigration [-instanceName] [-testOnly] [-migrate] [-updateSize]

You must provide one of the optional parameters. The meaning of the parameters is:

- -instanceOnly - the name of your ExtraView instance, e.g. *evj*. This parameter is required
- -testOnly - perform a dry run of the migration, but without making any changes to the database or to the file system
- -migrate - perform the migration
- -updateSize - update the size of the external attachments in the database (not normally used)

Microsoft Windows with Apache Tomcat

Edit the doDocumentMigration.bat file.

Locate the lines

set JAVA_HOME=%1
set TOMCAT_HOME=%2
set EV_BASE=%3

replace %1 with your path to Java, e.g. D:ExtraView\java\jdk-1.4.2
replace %2 with your path to Tomcat, e.g. D:ExtraView\jakarta-tomcat-5.0.28
replace %3 with your path to ExtraView, e.g. D:ExtraView\jakarta-tomcat-5.0.28\webapps\evj

Save and exit the edit session.

Open a Command window in the evj\WEB-INF\data folder and type the following command:

doDocumentMigration instanceName "parameter"

where

instanceName and "parameter" are defined by the options listed above.

Linux / Unix with Tomcat

Edit the doDocumentMigration.sh file.

Locate the lines

TOMCAT_HOME=/usr/local/extraview/tomcat
JAVA_HOME=/usr/local/extraview/java

Set TOMCAT_HOME and JAVA_HOME to the correct paths for your installation.

Save and exit the edit session.

At the command prompt, type the following command:

sh doDocumentMigration.sh instanceName "parameter"

where

instanceName and "parameter" are defined by the options listed above.

Completing the Migration

You must use the Java utility named runPatches when you have completed the above steps. The setup for this is identical to the setup for doDocumentMigration, except you edit the file named either runPatches.bat or runPatches.sh. Once you have set up the file, simply run the runPatches from the command line.

Again: We strongly recommend that you back up your database prior to performing any of the procedures described above. Please contact support@extraview.com if you would like assistance with this step.

VPAT for ExtraView

ExtraView Corporation is committed to the delivery of software that meets or exceeds the requirements of Section 508.

In January 2009, the Information Technology Industry Council (ITI) asked for the development of a specification that could be used for ITI's Voluntary Product Accessibility Template (VPAT). This VPAT has been embraced by the U.S. General Services Administration as a way to simplify government market research on IT products with accessible features. This page indicates how the VPAT is supported by ExtraView's web-based, end-user application.

The first table of the Template provides a summary view of the section 508 Standards. The subsequent tables provide more detailed views of each subsection. There are three columns in each table. Column one of the Summary Table describes the subsections of subparts B and C of the Standards. The second column describes the supporting features of the product or refers you to the corresponding detailed table, "e.g., equivalent facilitation." The third column contains any additional remarks and explanations regarding ExtraView. In the subsequent tables, the first column contains the lettered paragraphs of the subsections. The second column describes the supporting features of the product with regard to that paragraph. The third column contains any additional remarks and explanations regarding the product.

ExtraView is reliant on the correct functioning and support of an Internet browser. Supported browsers are Microsoft Internet Explorer, Mozilla Firefox, Google Chrome and Apple Safari. If there are exceptions to any of the VPAT criteria within any of these browsers, then ExtraView will only meet the criteria to the extent that the browser meets the criteria. The VPAT for these browsers should be consulted for any and all limitations that may exist.

Summary Table **VPAT™** **Voluntary Product Accessibility Template**®		
Criteria	**Supporting Features**	**Remarks and explanations**
Section 1194.21 Software Applications and Operating Systems		
Section 1194.22 Web-based Internet Information and Applications		
Section 1194.23 Telecommunications Products		
Section 1194.24 Video and Multi-media Products		
Section 1194.25 Self-Contained, Closed Products		
Section 1194.26 Desktop and Portable Computers		
Section 1194.31 Functional Performance Criteria		
Section 1194.41 Information, Documentation and Support		

Section 1194.21 Software Applications and Operating Systems – Detail **VPAT™** **Voluntary Product Accessibility Template**®		
Criteria	**Supporting Features**	**Remarks and explanations**
(a) When software is designed to run on a system that has a keyboard, product functions shall be executable from a keyboard where the function itself or the result of performing a function can be discerned textually.	Supports with exceptions that may exist within the browser software that is required for ExtraView to function	ExtraView supports all standard keyboard operations of the user interface provided by the browser software that is required to operate ExtraView.

(b) Applications shall not disrupt or disable activated features of other products that are identified as accessibility features, where those features are developed and documented according to industry standards. Applications also shall not disrupt or disable activated features of any operating system that are identified as accessibility features where the application programming interface for those accessibility features has been documented by the manufacturer of the operating system and is available to the product developer.	ExtraView does not interfere with any keyboard accessibility features built into the operating system or the browser that supports ExtraView.	ExtraView does not interfere with Mouse Keys, Sticky Keys, Filter Keys or Toggle Keys
(c) A well-defined on-screen indication of the current focus shall be provided that moves among interactive interface elements as the input focus changes. The focus shall be programmatically exposed so that Assistive Technology can track focus and focus changes.	ExtraView utilizes a visual focus indicator that moves among interactive objects as in the input focus changes is provided. This visual focus indicator is provided by the browser that supports ExtraView and ExtraView is therefore subject to limitations and extensions of any individual browser type.	
(d) Sufficient information about a user interface element including the identity, operation and state of the element shall be available to Assistive Technology. When an image represents a program element, the information conveyed by the image must also be available in text.	ExtraView provides information to the extent that the underlying browser provides this information and is subject to any limitations that may exist in the supported browser required for operation.	ExtraView extends functionality by providing a means to document each element of the user interface into which data is entered and stored with a tooltip. The ExtraView administrator is able to define a context and function sensitive message which appears on the screen and is available to assistive technology.
(e) When bitmap images are used to identify controls, status indicators, or other programmatic elements, the meaning assigned to those images shall be consistent throughout an application's performance.	There is a consistent set of bit map images used throughout the user interface.	There are a number of complete sets of images, or themes, that are selectable by the administrator, to offer the user of ExtraView. All images that are members of a set are consistent throughout the set, in regard to color, style and font.
(f) Textual information shall be provided through operating system functions for displaying text. The minimum information that shall be made available is text content, text input caret location, and text attributes.	Again, ExtraView is compliant with this criteria to the extent that the required browser is compliant.	
(g) Applications shall not override user selected contrast and color selections and other individual display attributes.	Again, ExtraView is compliant with this criteria to the extent that the required browser is compliant. Note that different browsers offer a different level of compliance with this criteria.	In practice this is not a limitation for ExtraView's end users unless an end user's browser has been configured with unusual settings.
(h) When animation is displayed, the information shall be displayable in at least one non-animated presentation mode at the option of the user.	The only place where animation is used is for progress bars for lengthy data import, and mass update operations. In these cases, the progress is indicated with both an animation and with text-based numeric counters that show how far the operation has progressed and how far there is to go to reach completion.	The user interface never relies only on animation for user feedback and always offers text information.
(i) Color coding shall not be used as the only means of conveying information, indicating an action, prompting a	Firstly, ExtraView works within any constraints imposed by the supported browsers. Secondly, ExtraView uses color	Many of the distinguishing elements may be set by the ExtraView administrator, so much of the control becomes the

response, or distinguishing a visual element.	to differentiate different elements of information on the screens generated to end users. However, this color embellishment is for aesthetics only and does not impart any essential information to the end user. When there is a requirement to distinguish information, ExtraView uses non-color elements such as underlines, bolding and font sizes.	responsibility of the administrator to ensure the meeting of the criteria.
(j) When a product permits a user to adjust color and contrast settings, a variety of color selections capable of producing a range of contrast levels shall be provided.	The only area of ExtraView where end users may choose color settings is when preparing charts for viewing. A number of different color preconfigured palettes are available, from high-contrast color to monochrome.	End users may also define their own color palettes for charts, and may choose from any colors that their computer browser and computer operating system will generate.
(k) Software shall not use flashing or blinking text, objects, or other elements having a flash or blink frequency greater than 2 Hz and lower than 55 Hz.	ExtraView does not utilize any flashing or blinking objects.	
(l) When electronic forms are used, the form shall allow people using Assistive Technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues.	ExtraView offers highly customizable forms. Indeed this feature is at the core of ExtraView's functionality. Each form can be designed with a variable number of fields, of varying types, and this information is programmable by the ExtraView administrator. ExtraView offers a complete toolbox (the Design Center) with which to create the electronic forms, and it is the administrator's responsibility to ensure that compliance with this criteria is met. For example, every field has the capability of text being available to assistive technology, but it is the administrator's responsibility to compose relevant text for the field.	The text offered to assistive technology is fully customizable by the ExtraView administrator, and each message may be of any length up to 4,000 characters.

Section 1194.22 Web-based Internet information and applications – Detail VPAT™ Voluntary Product Accessibility Template®

Criteria	Supporting Features	Remarks and explanations
(a) A text equivalent for every non-text element shall be provided (e.g., via "alt", "longdesc", or in element content).	Non-text elements are implemented in the form of bitmap images. All such images are complemented with a text equivalent which can be interpreted by assistive technology.	It is possible for the ExtraView administrator to implement non-text elements. When they do this, it is their responsibility to provide a text equivalent.
(b) Equivalent alternatives for any multimedia presentation shall be synchronized with the presentation.	Not applicable	
(c) Web pages shall be designed so that all information conveyed with color is also available without color, for example from context or markup.	Color is used for aesthetics only and does not impart any essential information.	
(d) Documents shall be organized so they are readable without requiring an associated style sheet.	It is the responsibility of the ExtraView administrator to comply with this criteria when they configure an ExtraView installation. For example, it is possible to	It is ExtraView Corporation's recommendation that administrators do not utilize the HTML Area field display type if compliance with Section 508 is

	implement a field display type known as an *HTML Area* field. In this case, markup information within the field might be inserted by an end user that would not satisfy this criteria.	required. There are two other field display types that may be used as alternatives.
(e) Redundant text links shall be provided for each active region of a server-side image map.	Not applicable	
(f) Client-side image maps shall be provided instead of server-side image maps except where the regions cannot be defined with an available geometric shape.	Only end user charts which are generated with drilldown links utilize client-side image maps. In all cases the regions are defined within either a rectangular or pie-slice shape.	
(g) Row and column headers shall be identified for data tables.	These are implemented throughout the reporting function and all data tables that present information in the form of results to an end user have the appropriate column and/or row headers.	There are a number of generated data tables that only use column headers and row headers are not applicable in their context.
(h) Markup shall be used to associate data cells and header cells for data tables that have two or more logical levels of row or column headers.	Only reports known as Matrix Reports fall within this category, and there are two logical levels of row and column headers. Each of these is supported with markup that indicates to assistive devices what the underlying data means.	
(i) Frames shall be titled with text that facilitates frame identification and navigation	There are two possible frames used through ExtraView, that present information to the end user. In both cases, these are identified with titles	There is a third frame that appears on some screens. This is hidden from the end user and is only used to store transactional data that is invisible to the end user. Purposely, this frame is not titled.
(j) Pages shall be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz.	Not applicable	
(k) A text-only page, with equivalent information or functionality, shall be provided to make a web site comply with the provisions of this part, when compliance cannot be accomplished in any other way. The content of the text-only page shall be updated whenever the primary page changes.	Not applicable	
(l) When pages utilize scripting languages to display content, or to create interface elements, the information provided by the script shall be identified with functional text that can be read by Assistive Technology.	JavaScript is used within various pages to display user alerts. These alerts are text-based and are visible to assistive technology. Ajax techniques are frequently used to retrieve parts of screen forms from the server. When this occurs, the appropriate text information that is required by assistive technology is automatically downloaded to the user's browser.	
(m) When a web page requires that an applet, plug-in or other application be present on the client system to interpret page content, the page must provide a link to a plug-in or applet that complies with §1194.21(a) through (l).	There is a single function within ExtraView (file upload) that utilizes an applet, as opposed to HTML and JavaScript. This applet only contains a single message, supported by alternative	

	text, and is therefore compliant with this criteria.	
(n) When electronic forms are designed to be completed on-line, the form shall allow people using Assistive Technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues.	ExtraView offers highly customizable forms. Indeed this feature is at the core of ExtraView's functionality. Each form can be designed with a variable number of fields, of varying types, and this information is programmable by the ExtraView administrator. ExtraView offers a complete toolbox (the Design Center) with which to create the electronic forms, and it is the administrator's responsibility to ensure that compliance with this criteria is met. For example, every field has the capability of text being available to assistive technology, but it is the administrator's responsibility to compose relevant text for the field.	The text offered to assistive technology is fully customizable by the ExtraView administrator, and each message may be of any length up to 4,000 characters.
(o) A method shall be provided that permits users to skip repetitive navigation links.	Not applicable	

Section 1194.23 Telecommunications Products – Detail VPAT™ Voluntary Product Accessibility Template®		
Criteria	**Supporting Features**	**Remarks and explanations**
(a) Telecommunications products or systems which provide a function allowing voice communication and which do not themselves provide a TTY functionality shall provide a standard non-acoustic connection point for TTYs. Microphones shall be capable of being turned on and off to allow the user to intermix speech with TTY use.	Not applicable	
(b) Telecommunications products which include voice communication functionality shall support all commonly used cross-manufacturer non-proprietary standard TTY signal protocols.	Not applicable	
(c) Voice mail, auto-attendant, and interactive voice response telecommunications systems shall be usable by TTY users with their TTYs.	Not applicable	
(d) Voice mail, messaging, auto-attendant, and interactive voice response telecommunications systems that require a response from a user within a time interval, shall give an alert when the time interval is about to run out, and shall provide sufficient time for the user to indicate more time is required.	Not applicable	
(e) Where provided, caller identification and similar telecommunications functions shall also be available for users of TTYs, and for users who cannot see displays.	Not applicable	

(f) For transmitted voice signals, telecommunications products shall provide a gain adjustable up to a minimum of 20 dB. For incremental volume control, at least one intermediate step of 12 dB of gain shall be provided.	Not applicable	
(g) If the telecommunications product allows a user to adjust the receive volume, a function shall be provided to automatically reset the volume to the default level after every use.	Not applicable	
(h) Where a telecommunications product delivers output by an audio transducer which is normally held up to the ear, a means for effective magnetic wireless coupling to hearing technologies shall be provided.	Not applicable	
(i) Interference to hearing technologies (including hearing aids, cochlear implants, and assistive listening devices) shall be reduced to the lowest possible level that allows a user of hearing technologies to utilize the telecommunications product.	Not applicable	
(j) Products that transmit or conduct information or communication, shall pass through cross-manufacturer, non-proprietary, industry-standard codes, translation protocols, formats or other information necessary to provide the information or communication in a usable format. Technologies which use encoding, signal compression, format transformation, or similar techniques shall not remove information needed for access or shall restore it upon delivery.	Not applicable	
(k)(1) Products which have mechanically operated controls or keys shall comply with the following: Controls and Keys shall be tactilely discernible without activating the controls or keys.	Not applicable	
(k)(2) Products which have mechanically operated controls or keys shall comply with the following: Controls and Keys shall be operable with one hand and shall not require tight grasping, pinching, twisting of the wrist. The force required to activate controls and keys shall be 5 lbs. (22.2N) maximum.	Not applicable	
(k)(3) Products which have mechanically operated controls or keys shall comply with the following: If key repeat is supported, the delay before repeat shall be adjustable to at least 2 seconds. Key repeat rate shall be adjustable to 2 seconds per character.	Not applicable	
(k)(4) Products which have mechanically operated controls or keys shall comply	Not applicable	

with the following: The status of all locking or toggle controls or keys shall be visually discernible, and discernible either through touch or sound.		

Section 1194.24 Video and Multi-media Products – Detail VPAT™
Voluntary Product Accessibility Template®

Criteria	**Supporting Features**	**Remarks and explanations**
(a) All analog television displays 13 inches and larger, and computer equipment that includes analog television receiver or display circuitry, shall be equipped with caption decoder circuitry which appropriately receives, decodes, and displays closed captions from broadcast, cable, videotape, and DVD signals. As soon as practicable, but not later than July 1, 2002, widescreen digital television (DTV) displays measuring at least 7.8 inches vertically, DTV sets with conventional displays measuring at least 13 inches vertically, and stand-alone DTV tuners, whether or not they are marketed with display screens, and computer equipment that includes DTV receiver or display circuitry, shall be equipped with caption decoder circuitry which appropriately receives, decodes, and displays closed captions from broadcast, cable, videotape, and DVD signals.	Not applicable	
(b) Television tuners, including tuner cards for use in computers, shall be equipped with secondary audio program playback circuitry.	Not applicable	
(c) All training and informational video and multimedia productions which support the agency's mission, regardless of format, that contain speech or other audio information necessary for the comprehension of the content, shall be open or closed captioned.	Not applicable	
(d) All training and informational video and multimedia productions which support the agency's mission, regardless of format, that contain visual information necessary for the comprehension of the content, shall be audio described.	Not applicable	
(e) Display or presentation of alternate text presentation or audio descriptions shall be user-selectable unless permanent.	Not applicable	

Section 1194.25 Self-Contained, Closed Products – Detail VPAT™
Voluntary Product Accessibility Template®

Criteria	Supporting Features	Remarks and explanations
(a) Self contained products shall be usable by people with disabilities without requiring an end-user to attach Assistive Technology to the product. Personal headsets for private listening are not Assistive Technology.	Not applicable	
(b) When a timed response is required, the user shall be alerted and given sufficient time to indicate more time is required.	Not applicable	
(c) Where a product utilizes touchscreens or contact-sensitive controls, an input method shall be provided that complies with §1194.23 (k) (1) through (4).	Not applicable	
(d) When biometric forms of user identification or control are used, an alternative form of identification or activation, which does not require the user to possess particular biological characteristics, shall also be provided.	Not applicable	
(e) When products provide auditory output, the audio signal shall be provided at a standard signal level through an industry standard connector that will allow for private listening. The product must provide the ability to interrupt, pause, and restart the audio at anytime.	Not applicable	
(f) When products deliver voice output in a public area, incremental volume control shall be provided with output amplification up to a level of at least 65 dB. Where the ambient noise level of the environment is above 45 dB, a volume gain of at least 20 dB above the ambient level shall be user selectable. A function shall be provided to automatically reset the volume to the default level after every use.	Not applicable	
(g) Color coding shall not be used as the only means of conveying information, indicating an action, prompting a response, or distinguishing a visual element.	Not applicable	
(h) When a product permits a user to adjust color and contrast settings, a range of color selections capable of producing a variety of contrast levels shall be provided.	Not applicable	
(i) Products shall be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz.	Not applicable	
(j) (1) Products which are freestanding, non-portable, and intended to be used in one location and which have operable controls shall comply with the following:	Not applicable	

The position of any operable control shall be determined with respect to a vertical plane, which is 48 inches in length, centered on the operable control, and at the maximum protrusion of the product within the 48 inch length on products which are freestanding, non-portable, and intended to be used in one location and which have operable controls.		
(j)(2) Products which are freestanding, non-portable, and intended to be used in one location and which have operable controls shall comply with the following: Where any operable control is 10 inches or less behind the reference plane, the height shall be 54 inches maximum and 15 inches minimum above the floor.	Not applicable	
(j)(3) Products which are freestanding, non-portable, and intended to be used in one location and which have operable controls shall comply with the following: Where any operable control is more than 10 inches and not more than 24 inches behind the reference plane, the height shall be 46 inches maximum and 15 inches minimum above the floor.	Not applicable	
(j)(4) Products which are freestanding, non-portable, and intended to be used in one location and which have operable controls shall comply with the following: Operable controls shall not be more than 24 inches behind the reference plane.	Not applicable	

Section 1194.26 Desktop and Portable Computers – Detail VPAT™ Voluntary Product Accessibility Template®		
Criteria	**Supporting Features**	**Remarks and explanations**
(a) All mechanically operated controls and keys shall comply with §1194.23 (k) (1) through (4).	Not applicable	
(b) If a product utilizes touchscreens or touch-operated controls, an input method shall be provided that complies with §1194.23 (k) (1) through (4).	Not applicable	
(c) When biometric forms of user identification or control are used, an alternative form of identification or activation, which does not require the user to possess particular biological characteristics, shall also be provided.	Not applicable	
(d) Where provided, at least one of each type of expansion slots, ports and connectors shall comply with publicly available industry standards	Not applicable	

Section 1194.31 Functional Performance Criteria – Detail VPAT™
Voluntary Product Accessibility Template®

Criteria	Supporting Features	Remarks and explanations
(a) At least one mode of operation and information retrieval that does not require user vision shall be provided, or support for Assistive Technology used by people who are blind or visually impaired shall be provided.	Assistive Technology supports this criteria.	Note that each user has the ability to alter the text size of all information presented to them when using ExtraView. This setting allows visually impaired users to better use ExtraView, without the limitations that are often inherent when using browser settings alone.
(b) At least one mode of operation and information retrieval that does not require visual acuity greater than 20/70 shall be provided in audio and enlarged print output working together or independently, or support for Assistive Technology used by people who are visually impaired shall be provided.	Both modes are provided within ExtraView.	
(c) At least one mode of operation and information retrieval that does not require user hearing shall be provided, or support for Assistive Technology used by people who are deaf or hard of hearing shall be provided	Not applicable	
(d) Where audio information is important for the use of a product, at least one mode of operation and information retrieval shall be provided in an enhanced auditory fashion, or support for assistive hearing devices shall be provided.	Not applicable	
(e) At least one mode of operation and information retrieval that does not require user speech shall be provided, or support for Assistive Technology used by people with disabilities shall be provided.	Not applicable	
(f) At least one mode of operation and information retrieval that does not require fine motor control or simultaneous actions and that is operable with limited reach and strength shall be provided.	Not applicable	

Section 1194.41 Information, Documentation and Support – Detail VPAT™
Voluntary Product Accessibility Template®

Criteria	Supporting Features	Remarks and explanations
(a) Product support documentation provided to end-users shall be made available in alternate formats upon request, at no additional charge	Product support documentation is provided as standard and at no charge to all end users in an online format, accessed through a standard web browser, where Assistive Technology may be utilized. The documentation may also be dowloaded by any end user in the Adobe PDF format. Other formats may be obtained by copying the text of the product documentation into a word	

	processor or text editor for reading or conversion into alternate formats.	
(b) End-users shall have access to a description of the accessibility and compatibility features of products in alternate formats or alternate methods upon request, at no additional charge.	This VPAT is available online as part of the product documentation. In addition the product documentation refers to many parts of this VPAT to highlight specific criteria.	
(c) Support services for products shall accommodate the communication needs of end-users with disabilities.	Support services for the product are available in many forms, telephone, email, fax and writing are the common forms of accessing the support services.	

More information on how ExtraView meets the needs to comply with Section 508 can be seen here.