Command Line Interface

This documentation covers the functionality of the ExtraView Command Line Interface (CLI).

This document is intended for the experienced computer user who has a good understanding of either the UNIX, Linux, Microsoft Windows NT or the Microsoft Windows 2000 environments from which they will use the Command Line Interface (CLI). To take maximum advantage of the features offered by the CLI, knowledge of languages such as Perl and HTML are helpful.

If you intend to modify the behavior of ExtraView with user custom programming, you will need to be skilled in coding with the Java language. In addition, ExtraView administration skills are required to configure many of the functions offered.

The CLI is implemented on top of the RESTful API which is described in the Application Programming Interface guide. The key features of the CLI are:

- Insert, update and delete records in the ExtraView database from remote applications
- Search the ExtraView database and return a set of records defined in a query
- Export information from the ExtraView database for input to a data warehouse
- Upload and download file attachments to and from the ExtraView database
- Provide limited administration access to create metadata and to manage user accounts
- Show the names of fields within the ExtraView database to which the user has access
- Provide high-level links via URLs from third-party applications that have no knowledge of ExtraView.

Downloadable PDF

The Command Line Interface Guide is downloadable as a single PDF by clicking here. You will need the Adobe Acrobat Reader to view this.

Installation & Configuration

Please consult the Installation & Upgrade Guide for details on how to install the CLI.

Configuring for Linux / UNIX

To configure the CLI, all users must configure their accounts with an environment variable named EVDIR. This should have a value that points to the directory that contains the file named evconfig.txt. For example, the appropriate command may be:

setenv EVDIR=/extraview/api

The file evconfig.txt contains information that points to the server components where ExtraView is located as well as other required information. An example of the file is shown below.

```
SERVER = http://myserver.mycompany.com/evj/ExtraView
POP3_USER = myname
POP3_PASSWORD = mypass
POP3_SERVER = mail.mycompany.com
SUBJ_REGEX = ^Subject
SMTP_SERVER = mail.mycompany.com
```

SERVER	This entry is the URL of the ExtraView API module as installed
POP3_USER	The user's email address on the POP3 server
POP3_PASSWORD	The user's email password on the POP3 server
POP3_SERVER	The address of the POP3 server on the network
SUBJ_REGEX	Regular expression used to identify the submitted issue and the action to be taken when the mail is processed
SMTP_CC	This address is used when delivery of mail through ExtraView fails. This avoids mail not being delivered to any address and just sitting in a queue
SMTP_FAIL_MSG_FNAM	This is the absolute path of a file that contains a template to be used in the event of a failure of the mail delivery system through ExtraView. If the message is not processed by ExtraView, this file is used as the body of the mail to be sent to the addresses in SMTP_FROM and SMTP_CC. If the body contains a tag __BODY__, then the tag is replaced by the email being processed. If the body contains a tag __REASON_, then the tag is replaced by the error message returned as an error message from ExtraView
SMTP_FROM	This address is used when delivery of mail through ExtraView fails. This avoids mail not being delivered to any address and just sitting in a queue
SMTP_SERVER	The SMTP server for email.

Configuring for Microsoft Windows

This is very similar to the Linux / UNIX installation. Please follow the instructions but with these differences:

- The .evrc file is named evrc.txt and is stored in the current directory
- Windows NT and Windows 2000 use AT commands as opposed to cron jobs to run tasks on a timed basis.

Configuring the CLI to work with HTTPS

The CLI supports HTTPS, SSL SMTP and SSL POP3. Configuring the CLI to use these features requires manual configuration.

Step 1

Modify the evconfig.txt configuration file by adding the following lines:

```
HTTPS = on

POP3_SSL = on
POP3_SSL_PORT = 995 # change to your POP3 port number

SMTP_SSL = on
SMTP_SSL_PORT = 465 # change to your SMTP port number
```

HB_SMTP_SSL=on
HB_SMTP_SSL_PORT=465 # change to your HB_SMTP port number

Step 2

Download and install OpenSSL from http://www.openssl.org/. This must be done before the next step in the installation procedure.

Step 3

Download and manually install the following Perl Modules. These cannot be automatically installed by ExtraView's installation procedures as ExtraView's installation is machine independent and these modules generate and install machine-specific code during their installation.

Net_SSLeay.pm
Perl extension for using OpenSSL
Version: 1.30
Author: Sampo Kellomaki
CPAN: http://search.cpan.org/dist/Net_SSLeay.pm-1.30/

IO-Socket-SSL
Nearly transparent SSL encapsulation for IO::Socket::INET.
Version: 1.08
Author: Steffen Ullrich & Peter Behroozi & Marko Asplund
CPAN: http://search.cpan.org/dist/IO-Socket-SSL-1.08/

Net-SMTP-SSL
SSL support for Net::SMTP
Version: 1.01
Author: Casey West
CPAN: http://search.cpan.org/dist/Net-SMTP-SSL-1.01/

Crypt-SSLeay
OpenSSL glue that provides LWP https support
Version: 0.53
Author: Joshua Chamas
CPAN: http://search.cpan.org/dist/Crypt-SSLeay-0.53/

Concepts

The Command Line Interface is a set of Perl scripts implemented using ExtraView's API. These commands provide an alternative to the Browser GUI, for accessing many of ExtraView's features.

Many of the CLI commands can be executed either interactively, where the CLI command prompts for parameters or it can be executed with a single command line, with all the required arguments provided as command line options.
Although the CLI provides access to most of the features that are available through the GUI, a few features are not supported through the CLI. These include:

- Repeating records may not be added or edited through the CLI
- Multi-value list fields may not be edited through the CLI. If more than one value for a multi-value list field is specified with the evadd command, only the last value entered will be used. evupdate is not able to modify multi-value list fields at all. The behavior setting CLI_EDIT_MULTI_VALUE_FIELDS should always be set to NO. If this setting has a value of YES,

issues containing multi-value field data will lose this data any time those issues are edited with evupdate.

CLI command summary

evadd	inserts new records into the ExtraView database
evadd2group	adds an existing user to a user role
evaddlist	provides an ordered list of fields that are used to insert records
evaddudfvalues	add one or more values to a UDF field
evadduser	adds a new user to ExtraView
evcheck	displays the existing configuration settings
evdebug	turns on debug mode
evdefaults	fields and values which have a default in the data dictionary
evdelete	deletes existing issues from the database
evdeleteuser	deactivate an existing user
evdownload	downloads a copy of a file from an issue attachment
eveditlist	displays a list of fields that are used to update records
evfields	displays a list of available fields and their screen names
evfiles	retrieves a list of file attachments against a single issue
evget	retrieves a single record from the database
evgetfields	download a specific field or fields from an existing issue
evheartbeat	checks the health of an ExtraView database
evhelp	help pages on the CLI
evhist	searches for records updated since a given date
evimport	imports records into ExtraView
evimportav	import allowed value combinations into ExtraView
evimportJIS	imports records with Japanese text into ExtraView
evmail	the email interface to ExtraView
evmeta	retrieves a list of valid metadata for an issue
evpasswd	updates a user's password
evproj	allows a user to reset their default area and project
evreport	runs a pre-defined ExtraView report or returns a list of reports
evrole	sets a user's role
evsearch	provides a general search and retrieval mechanism
evsearchlist	displays a list of the available column names and titles

evset	sets user details so they do are not entered with each command
evsh	a shell optimized to run CLI and system commands
evtemplate	produces a template for the evadd and evupdate commands
evupdate	updates existing records in the ExtraView database
evupload	uploads a file as a new issue attachment
evuserfields	displays an ordered list of fields for the user record
evusergroups	displays a list of user groups
evusers	displays a list of users
evversion	displays the version of ExtraView
evxmlc	neatly formats XML output in a columnar form

Authentication

CLI commands require an active ExtraView username and password. The CLI provides four different mechanisms for providing this information:

1. **Interactive prompt from the CLI command**

   If neither a username nor a password is provided, the CLI command will prompt for them. If a username is provided with no password, the CLI command will prompt for just the password

2. **Command line options**

   A username, or username and password may be provided as command line options using the -U and -P options, respectively. If only the -U option is provided, the CLI command will prompt for just the password.

3. **evsh**

   The evsh command requests the user to enter their "login" information once, when the command is first run. After authentication, the user may interactively run CLI commands without needing to re-enter their login information each time they run a CLI command. Refer to the evsh man page for complete information

4. **.evrc**

   The evset command may be used to create a configuration file containing the user's name and password. The most significant drawback to this approach is that this login information is stored in a plain text file, protected only by the operating system's file permissions. Refer to the evset man page for the complete description. Note that you may use an alternative user ID in place of a user ID.

   Throughout this document, all command line examples assume that the user name and password are stored in the $HOME/.evrc (evrc.txt on Windows) configuration file.

XML Data Returned From CLI Commands

Much of the data returned by a CLI/API call is in XML format. This has some significance to the user of the CLI, in that Extraview's XML data may embed your own XML within its results. To accommodate this, ExtraView uses Base64 encoding whenever it sees XML data returned from the application to ensure that

the XML returned by CLI commands through the API must be well-formed. This means that the contents of a CDATA string must not contain the character string "]]>", because that is the end sentinel for a CDATA section. So, if the original data contains this string, there must be some way to escape the data. For easy recognition of an escaped CDATA string, ExtraView prepend the characters %25S to the front of the string. These characters are merely a sentinel and are not part of the output string. The encoding used for the rest of the CDATA string is called Base64, and algorithms for encoding/decoding are widely available. Furthermore, ExtraView ensures that the %25S sentinel string does not appear in the CDATA raw character string by encoding any CDATA raw character string to Base64 as well. It is the responsibility of the receiver, therefore, to test each CDATA section for the sentinel characters %25S at the beginning of the CDATA, and, if present, perform the Base64 decode function on the remainder of the character data to get the raw character values in the field.

Fixed Database Names vs. Display Titles

In order to provide flexibility, many of the CLI commands allow users to refer to fields by either their fixed database names, or by their display titles. Every field defined in the Data Dictionary has both a fixed database name, and a display title. The documentation for the individual CLI commands indicates which naming convention should be used.

The CLI and Image Display Type Fields

As the CLI is a character-based interface, fields with a display type of image are not supported. These fields are ignored in the CLI.

The CLI and Business Rules

When in the ExtraView web-based interface business rules are executed as part of loading screens, making AJAX calls and refreshing screens, as well as when updating the database. When you are working in the CLI, there is no concept of screens, so business rules that are triggered in the web interface for loading and refreshing screens cannot be executed. Rules that are applied pre- and post-update and the email rules are applied however.

Rule Directive	Through Web Interface	Through CLI
Load	Yes	No
Onchange / Refresh	Yes	No
Preupdate	Yes	Yes
Postupdate	Yes	Yes
Email	Yes	Yes

Common CLI Options

Several command line options are common to almost every CLI command. For clarity, these options are not documented on the individual manual pages for the CLI commands.

-F alt-config-file

This option allows the user to specify an alternate evconfig.txt file, for the current CLI command. The full pathname must be specified. When used, this option must be the first option on the command line.

-S servername

Most of the CLI commands support the -s option:

evcommand -s servername

This option allows you to override the servername specified in the evconfig.txt file.

evupload and evdownload use a capitalized -S:

command -S servername

-H or -?

All commands support a -H option (or -?), which prints the documentation for that command.

-U and -P

These command line options are used for specifying a username and password on the command line.

-A

If ExtraView has been configured to support anonymous logins, this option specifies that the CLI command should be run using the ANONYMOUS_API_USER_ID login. This functionality is enabled when the behavior setting ALLOW_ANONYMOUS_API_ACCESS has a value of YES.

CLI Templates

Templates are stored on the server, and can be used to format the output of CLI commands. The following commands support an optional -T templatefile command line option:

CLI Command	API statevar (if using a direct API call)
evadd	INSERT
evadduser	INSERT_USER
evdelete	DELETE
evfiles	LIST_ATTACHMENT
evgetuser	GET_USER
evpasswd	UPDATE_USER_PASSWORD
evget	GET
evsearch	SEARCH
evupdate	UPDATE
evupload	ADD_ATTACHMENT

The second column contains statevar parameters that can be used when requests are submitted to ExtraView's RESTful API through a URL. The URL must include a p_template_file parameter with the

name of the template file. The API guide has additional information on the use of templates. Example usage:

evget -T get.html 12341

This will cause ExtraView to look for a file named get.html in the user_templates directory on the server. This directory is at the same level as the .../webapps/evj/WEB-INF/templates directory. Filling in the template get.html and then displaying the result will complete the output of the evget command. This template will have tags marked like so:

__NAME__

These entries are data dictionary names within ExtraView, as returned by evfields or equivalently those names that were placed in get.txt and search.txt by the CLI command evfields -g.

Exceptions to this convention are fields with a display type of text area and log area. These fields are stored within ExtraView with additional information and are used as follows, where DDNAME is replaced with the data dictionary name:

Field name	Requiredness	Purpose
__DDNAME__	Mandatory	No data is returned with this field, but it is required in order that ExtraView can set up the retrieval of the remainder of the fields. If you are designing an HTML template, it is recommended that you include this within a comment, e.g. <!-- __DESCRIPTION__ -->
__DDNAME.TEXT__	Optional	This field contains the actual text and is normally used in the template
__DDNAME.USER__	Optional	The user's name who created or last updated the text field
__DDNAME.TIMESTAMP__	Optional	The timestamp of the last update to the field

An HTML fragment that displays the field named DESCRIPTION may look like this:

```
<TABLE>
<TR>
<TD>Description</TD>
<!-- __DESCRIPTION__ -->
<TD>__DESCRIPTION.TEXT__</TD>
</TR>
</TABLE>
```

If the requested issue in the evget example command does not exist, we have this result:

ERR_RESULTS Problem #12341 does not exist.

The command evsearch is very similar. The difference is that evsearch may use three template files; a header file (optional), a body file that is used in a repeating fashion for each record returned by the search, and a footer file (optional). For example:

evsearch -T search.html keywords=test q

This will use the files hsearch.html, bsearch.html and fsearch.html for the display. The hsearch.html and fsearch.html files are optional. If there is no file named bsearch.html, ExtraView will search for a file named search.html for the display of each record. If this file is also not found an error is returned to the CLI user. The tag RECORD_COUNT is available in the header and footer template files. This contains the total record count for the search.

The following list describes all the supported template __NAME__ tags, organized by CLI command.

- evadd
  - The same fields for evget - see above.
  - ERR_RESULTS – "An error occurred while inserting an issue
- evupdate
  - ID – the ID of the inserted record
  - RESULTS – "Problem #12341 updated."
  - ERR_RESULTS – "An error occurred while updating the issue."
- evdelete
  - ID – the ID of the deleted record
  - RESULTS – "Problem #12341 deleted."
  - ERR_RESULTS – "You do not have permission to delete."
- evadduser
  - RESULTS – "User xxx inserted."
  - ERR_RESULTS – "Error during adding of user: <error message>"
- evpasswd
  - RESULTS – "Password changed for xxx."
  - ERR_RESULTS – "User xxx does not exist." or "Incorrect old password." or some other exception
- evfiles
  - ID – the ID of the record you requested
  - NFILES – the number of attached files
  - DATE_CREATED – repeating field
  - FILE_NAME – repeating field
  - FILE_SIZE – repeating field
  - CREATED_BY_USER – repeating field
  - ATTACHMENT_ID – repeating field
  - FILE_DESC – repeating field
  - Repeating fields are used within a __REPEAT_START__ ... __REPEAT_STOP__ block like this:

```
<TABLE>
<TR>
<TD>Date Created</TD>
<TD>File Name</TD>
<TD>File Size</TD>
<TD>Created by User</TD>
<TD>Attachment ID</TD>
<TD>File Description</TD>
</TR>
__REPEAT_START__
<TR>
```

```
<TD>__DATE_CREATED__</TD>
<TD>__FILE_NAME__</TD>
<TD>__FILE_SIZE__</TD>
<TD>__CREATED_BY_USER__</TD>
<TD>__ATTACHMENT_ID__</TD>
<TD>__FILE_DESC__</TD>
</TR>
__REPEAT_STOP__
</TABLE>
```

    - ERR_RESULTS "Problem #12341 does not exist."
- evupload
    - This uploads the file and then displays the record itself with the template file (see evget). It then prepends an 'f' to the template filename and displays all the attached files (including the one just attached) (see evfiles). Either of these can be an empty file but they both must exist.

      The URL statevar of ADD_ATTACHMENT requires some special care. Here is some sample HTML:

```
<FORM METHOD=post action="http://test.extraview.net/evj/ExtraView/ev_api.action?
p_user_id=mary&p_password=smith&
statevar=add_attachment&id=15948&p_template_file=get.html" enctype="multipart/form-
data">
File: <input type="file" size="40" name="file">
<BR>
Description: <input type="text" size="40" name="p_attach_desc"><BR>
<INPUT type="hidden" name="p_problem_id" value="15948">
<INPUT type="hidden" name="p_called_from" value="pr_edit_problem">
<INPUT type="hidden" name="p_user_id" value="system">
<input type="Submit">
</FORM>
</BODY>>
</HTML>
```

- evgetuser
    - There is actually no such API command... This is for internal purposes –- you pass the command get_user with an email address. You are returned the user ID and the company name with that email address.
        - ID – the user ID
        - COMPANY_NAME – the company name
        - ERR_RESULTS – "missing email parameter"

      For all the above commands the name ERR_RESULTS is not displayed using the specified template file. ExtraView will use the template file error.html if it exists or, if it does not, simply displays the string on the output. This obviates the need for normal display templates to deal with error conditions.

      The URL statevar of FILL_IN takes a p_template_file parameter and fills in the template with the other parameters passed. For example:

      http://www.extraview.net/evj/ExtraView/ev_api.action?
      user_id=myname&password=secret&statevar=fill_in&p_template_file=files.html
      &ID=1234&NAME=Harriette

      Where files.html contains the tags __ID__ and __NAME__.

evadd

Add a new issue into the ExtraView database. The command can be executed in several ways, depending on the options provided.

SYNOPSIS

evadd [-n]

evadd [-n][-i] [field1=value1 field2=value2 ... fieldn=valuen] [q]

evadd [-n] -a

evadd [-n] -e [-v] [-f]

DESCRIPTION

This command allows a user to insert a new issue into the ExtraView database. All permissions and business rules, including email notification, are implemented the same way through the CLI as they are when issues are added through the GUI.

This command may be run four different ways.

Interactive Menu

If neither the -a nor the -e command line options are specified, evadd will display a menu of all the fields available for adding a new issue. Each field is identified by a number. After the menu is displayed, the user will see the prompt, "Enter number or q to submit:". When a field number is entered, evadd will prompt for the value for that field. If the field is a list, the list of available values for that list will be displayed. If the field is a multiple line text field, the user may enter multiple lines of input text. These fields must be terminated by a line containing just a period.
In this menu, an asterisk (*) will identify all fields that are required but do not yet have values. After a value has been entered for a required field, the asterisk will be removed from the display.

After all the desired field values have been entered, press q to exit the menu and submit the issue to ExtraView.

Command-line only

If all required fields are included on the command line, and a trailing q ends the command line, the issue will be immediately submitted to ExtraView.

When field values are provided on the command line without a trailing q, the user will go into the interactive menu with those field values already populated.

Prompt Mode

With the -a option, the user will be prompted for a value for each field, in the same order as the fields are displayed with the evaddlist command. If the user tries to skip over a required field, a warning message is shown, "Must provide a value!", and the user is prompted for that field again.

To go back to a previous field in the list, enter a b at any prompt.

At the end of the list of fields, the issue is submitted to ExtraView. If there are fields which were not initially filled in, which are required as the result of a “required if” field attribute, the user will be notified of those missing values, and will be placed into the interactive menu, with all the field values preserved. In this case, the required fields will not be identified by an asterisk.

Edit Mode

The -e option invokes an editor to perform the data entry. The editor will use a template file as the basis for collecting the field values for adding the issue.

In UNIX / Linux systems, the editor defaults to /bin/vi. In Microsoft Windows environments, the editor defaults to c:winnt40notepad.exe. An alternate editor may be specified by either setting up an EVEDITOR environment variable, by specifying the EDITOR line in the evconfig.txt file, or by using the EDITOR environment variable.

The template file will be created by one of five methods:

1. The -f <mytemplate> command line option
2. With the EVADD environment variable
3. By the existence of the file $HOME/.evadd (evadd.txt on Windows)
4. By the existence of the file evcli/add.txt
5. It will automatically be generated from the list of fields returned by evaddlist

The format of a template file is similar to the following:

```
Title:

Product:

Comments:
.
Attachments:
```

The period following the Comments field in the example above signifies that a multiple line input is allowed. Text for multi-line fields should be inserted immediately after the line containing the multi-line field title (Comments in this example). A period on a line by itself terminates a multi-line field. If a multi-line field is not terminated with a period, evadd will treat the rest of the template file as part of the multi-line field, when the issue is submitted to ExtraView.

Similarly, when entering multi-line fields either through the interactive menu, or with the -a option, the field must be terminated with a line containing just a period.

When entering text into a multiple line field, the user may read in the contents of an existing file, as if that file had been typed it directly in, by using the ~r fname command on a line by itself. This is very useful for entering information into fields with display types of Text Area, Log Area, or Print Text.

When adding an attachment (through any of the methods described above, type in the attachment file name, followed by a space, and then the attachment description, on the lines following the keyword ‘Attachments’. Only one attachment may be specified per line.

OPTIONS

-n	suppresses the email notification sent whenever a new issue is created
-i	indicates that fields will be identified by their fixed database names instead of by their display titles
-a	runs the command in prompt mode
-e	runs the command in edit mode
-v	adds additional field list value information in the initial template file used within edit mode
-f templatefile	use templatefile as the initial template file
fieldn=valuen	assigns an initial value to a field. ‘fieldn’ is the Display Title of a field. ‘valuen’ must be a valid value for that field
q	ends a list of ‘fieldname=value’ pairs, the user will not be prompted for additional input.

NOTES

Repeating record data may not be added to an issue with the evadd command. First enter the base issue with evadd, and then use evupdate to add the repeating record data.

EXAMPLE 1

Entering an issue with evadd and no parameters –

```
$ evadd
*1. Title *10. Priority
*2. Product 11. Severity
3. Category 12. Owner
4. Component 13. OS
5. Platforms 14. Customer
6. Description 15. Originator
7. Comments 16. View
8. Workaround 17. Test Case ID
9. Release Notes 18. Test Case Location

Enter number or q to submit? 1
Title? This is the title
1. Title *10. Priority
*2. Product 11. Severity
3. Category 12. Owner
4. Component 13. OS
5. Platforms 14. Customer
6. Description 15. Originator
7. Comments 16. View
8. Workaround 17. Test Case ID
9. Release Notes 18. Test Case Location

Title => This is the title
Enter number or q to submit? 2

1. Global Interaction
2. Net Transactions
```

```
3. SMS

Choice for Product? 3

1. Title *10. Priority
2. Product 11. Severity
3. Category 12. Owner
4. Component 13. OS
5. Platforms 14. Customer
6. Description 15. Originator
7. Comments 16. View
8. Workaround 17. Test Case ID
9. Release Notes 18. Test Case Location

Title => This is the title, Product => SMS
Enter number or q to submit? 3

1. Enhancement
2. Hardware
3. Software

Choice for Category? 3

1. Title *10. Priority
2. Product 11. Severity
3. Category 12. Owner
4. Component 13. OS
5. Platforms 14. Customer
6. Description 15. Originator
7. Comments 16. View
8. Workaround 17. Test Case ID
9. Release Notes 18. Test Case Location

Title => This is the title, Product => SMS
Category => SOFTWARE
Enter number or q to submit? 6
Multiple Line field. Finish with . or EOF
Description?
> Here is the long description of the problem.
> This is a second line.
> And another one.
> .

1. Title *10. Priority
2. Product 11. Severity
3. Category 12. Owner
4. Component 13. OS
5. Platforms 14. Customer
6. Description 15. Originator
7. Comments 16. View
8. Workaround 17. Test Case ID
9. Release Notes 18. Test Case Location

Title => This is the title, Product => SMS
Category => SOFTWARE
Description => Here is the long description of the problem
This is a second line.
```

```
And another one.

Enter number or q to submit? 10

1. Low
2. Medium
3. High

Choice for Priority? 3

1. Title 10. Priority
2. Product 11. Severity
3. Category 12. Owner
4. Component 13. OS
5. Platforms 14. Customer
6. Description 15. Originator
7. Comments 16. View
8. Workaround 17. Test Case ID
9. Release Notes 18. Test Case Location

Title => This is the title, Product => SMS
Category => SOFTWARE
Description => Here is the long description of the problem
This is a second line.
And another one.
Priority => HIGH
Enter number or q to submit? q
Bug # 12345 added.
$
```

EXAMPLE 2

This shows entering all the fields and values at the command line. Note that the field names such as short_descr, owner, status, etc. can be ascertained by using the evfields command.

```
$ evadd short_descr="This is the title of the problem"
owner=username status=OPEN priority=HIGH
description="This is the description of the problem" customer=GE
```

EXAMPLE 3

This example uses the -a option. You are prompted for the value for each field, in the order of fields that appear with the evaddlist command. If you try to skip over a field that is required, you will see a prompt that reminds you that you "Must provide a value!".

```
% evadd -a
Title? This is the title of the problem

1. Net Transactions
2. Dev Tools
3. Widget
```

```
4. Global Interaction

Choice for Product? 3

1. P0
2. P1
3. P2
4. P3
5. P4
6. P5

Choice for Priority? 2
Originator? carl.koppel
Owner? jon.bjornstad
Choice for Platforms? 1

1. NetBSD
2. LINUX
3. ALL
4. FREEBSD
5. WINDOWS 98
6. WINDOWS NT
7. SOLARIS

Choice for OS? 3
Multiple Line field. Finish with . or EOF
Description?
> This is the description for the item. You can enter many lines for
> the description, or even read the description from a file.
> .
Bug # 12497 added.
%
```

evadd2group

Adds an existing user to one or more existing user roles

SYNOPSIS

evadd2group username [role]

DESCRIPTION

username must be an existing User ID or alternative User ID within the ExtraView database.

role may be either the fixed database name or the display title of an existing role.

When the role is provided on the command line, the user specified in username is given permission to change roles to the new role specified.

When the role is not provided, a list of available roles is displayed, and new roles may be selected from a menu. An asterisk (*) marks the roles in which username is already a member. At the prompt, "Which group", users may enter one or more numbers, indicating the new roles to be added. Multiple numbers

may be separated with either white space, or a comma.

The list of user roles may be displayed with the evusergroups command.

NOTES

Write permission for the SE_SECURITY_GROUP security key is required to successfully run this command.

EXAMPLE

```
$ evadd2group bill.smith
groups: ???
1. Administrator
*2. Development Manager
*3. Engineer
4. Guest
5. QA Engineer
6. QA Manager
7. Release Management
8. TAC Engineer
9. TAC Manager
10. Tech. Pubs. Manager
11. Technical Writer

Which group? 2 3
User bill.smith added to group Development Manager.
User bill.smith is already part of group Engineer.
$
```

evaddlist

Displays the list of fields on the add screen

SYNOPSIS

evaddlist [-r]

DESCRIPTION

This command displays the list of fields that are used when adding an issue to the ExtraView database. The field names are displayed in two columns. The first column is the field's fixed database name. The second column is the field's display title. The field names are printed out in the order that they appear on the ExtraView Add Issue screen, top to bottom, left to right.
When the -r option is used, the fixed database names and display titles are each terminated by a colon, with no white space in between the two names. Fields that are required on the add screen are preceded by an asterisk (*); multi-line input fields are preceded by a plus sign (+). Additionally, when a field has an allowed value parent, the fixed database name of the parent field is printed after the child field's colon-terminated display title. This parent field is not terminated by a colon.

EXAMPLE

```
$ evaddlist -r
+SHORT_DESCR:Title
+PRODUCT_NAME:Product
CATEGORY:Category
COMPONENT:Component
PLATFORM:Platforms
+*DESCRIPTION:Description
*COMMENTS:Comments
WORKAROUND:Workaround
RELEASE_NOTES:Release Notes
PRIORITY:Priority
SEVERITY_LEVEL:Severity
ASSIGNED_TO:Owner
OS:OS
CUSTOMER:Customer
OWNER:Originator
PRIVACY:View

$
```

evaddudfvalues

Create one or more new UDF metadata values in a UDF list field

SYNOPSIS

evaddudfvalues [-f] [-s] dd_name value1[;value2;valuen]

DESCRIPTION

This command adds one or more values to an existing UDF field list. The display type of the field must be one of:

- List
- Tab
- Pop-up

Values may optionally be embedded within double quote marks. This is mandatory if you want the semi-colon (;) character to appear within the metadata.

The default separator value of a semi-colon (;), can be altered to any other separator, with the -s option.

The -f option allows the input for the values to be derived from a text file. The format of list values in the text file uses white space as a delimiter between each data dictionary name (the DD_NAME) and the list value (the LIST_VALUE). You may have one DD_NAME and one LIST_VALUE per line in the text file, or you may have one DD_NAME with more than one LIST_VALUE per line. In this case, each LIST_VALUE is separated by the separator specified with the -s <separator> parameter.

When you use the -f <filename> parameter, you do not need to specify the DD_NAME in the command line.

EXAMPLE 1

```
DD_1 DD_1_LIST_VALUE1
DD_1 DD_1_LIST_VALUE2
DD_2 DD_2_LIST_VALUE1
DD_2 DD_2_LIST_VALUE2
```

EXAMPLE 2

```
DD_1 DD_1_LIST_VALUE1
DD_2 DD_2_LIST_VALUE1;DD_2_LIST_VALUE2
```

NOTES

- You may only load values into UDF fields with this command. You may not load values into the database fields named ALT_ID, AREA_ID, ASSIGNED_TO, CATEGORY, ORIGINATOR, OWNER, PRIORITY, PRIVACY, PRODUCT_LINE, PRODUCT_NAME, PROJECT_ID, RELEASE_FIXED, RELEASE_FOUND, RESOLUTION, SEVERITY_LEVEL, and STATUS. At the present time, you must load values for these fields through the web-based interface
- You may load any number of values with one invocation of the command
- For each value successfully loaded, you will receive the message "value loaded OK"
- Duplicate values are not allowed in UDF lists, and any duplicates will be rejected. Any rejects will be shown with the message "value not loaded"
- The default separator of a semi-colon (;) must be escaped, if you are using the CLI from a Linux, Solaris or Unix operating system, as most shells interpret the semi-colon as the end of the command, with a new command beginning on the same line. Use a back-slash () before the semi-colon to achieve this, or use the -s option to set the separator to a different character.

EXAMPLE

Load a single value into the UDF named CUSTOMER from the command prompt –

```
DD_1 DD_1_LIST_VALUE1
$ evaddudfvalues CUSTOMER Big & Brash Inc.
Processing values for CUSTOMER:
Big & Brash Inc. loaded OK
$
```

evadduser

Creates a new ExtraView user account

SYNOPSIS

evadduser ["User Id"=value0 "Alternative User Id "=value1 "First Name"=value2 "Last Name"=value3 Password=value4 "Email Address"=value5 Timezone=value6 Language=value7 "Job Title"=value8 "Company Name"=value9 "Address 1"=value10 "Address 2"=value11 City=value12 State/Province=value13 "Zip/Postal Code"=value14 Country=value15 Region=value16 "Work Phone"=value17 "Home Phone"=value18 "Cell Phone"=value19 Fax=value20 Pager=value21 "Area Id"=value22 "Project Id"=value23 "Start Page"=value24 "User Group"=value25] q

evadduser -a

evadduser

DESCRIPTION

There are three different ways this command may be run, command-line only, with an interactive menu, or in prompt mode.

Command-line only

If the ExtraView behavior setting named ENFORCE_DETAILED_USER_INFO has a value of NO, the following fields are required –

User Id
Password
First Name
Last Name
Email Address
Timezone

If the ExtraView behavior setting named ENFORCE_DETAILED_USER_INFO has a value of YES, then the following fields also become required –

Company Name
Address 1
City
State
Zip/Postal Code
Work Phone

If all required fields are included on the command line, and a trailing q terminates the command line, the new user will be immediately added to the ExtraView database.

The evuserfields command displays all field names associated with creating new user accounts.

Interactive Menu

With no options specified, evadduser will display a menu of all the fields available for adding a new user account. An asterisk (*) will identify required fields. After a value has been provided for a required field, the asterisk will be removed from the menu.

At the prompt, enter the number of the field to be input. After all desired fields have been entered, type a q at the prompt, and the user account will be submitted to ExtraView to be added to the database.

Prompt Mode

Users are prompted for a value for each field. If the user tries to skip over a field that is required, a warning message is printed, "Must provide a value!".

Entering b at the prompt will ‘back-up’ to the previous field.

NOTES

If there are errors after a new user account has been submitted, they will be reported. If a new user is successfully created, the message, "user username inserted", is printed.

Write permission for the SE_SECURITY_USER security key is required to successfully run this command.

After adding a new user, they must separately be assigned to a minimum of one user role, using the evadd2group command, else they will be given the minimum privileges associated with the user role named in the behavior setting named DEFAULT_USER_GROUP.

The Timezone setting should be given as a number, relative to UTC or GMT time. For example, enter -8 for Pacific Standard Time.

EXAMPLE

```
$ evadduser
* 1. User Id 14. Zip/Postal Code
* 2. First Name 15. Country
* 3. Last Name 16. Region
* 4. Password 17. Work Phone
* 5. Email Address 18. Home Phone
* 6. Timezone 19. Cell Phone
7. Language 20. Fax
8. Job Title 21. Pager
9. Company Name 22. Area Id
10. Address 1 23. Project Id
11. Address 2 24. Start Page
12. City 25. User Group
13. State/Province

Enter number or q to submit: 1
User Id? bill.smith
1. User Id 14. Zip/Postal Code
* 2. First Name 15. Country
* 3. Last Name 16. Region
* 4. Password 17. Work Phone
* 5. Email Address 18. Home Phone
* 6. Timezone 19. Cell Phone
7. Language 20. Fax
8. Job Title 21. Pager
9. Company Name 22. Area Id
10. Address 1 23. Project Id
11. Address 2 24. Start Page
12. City 25. User Group
13. State/Province

User Id => bill.smith
Enter number or q to submit: 2
First Name? William
1. User Id 14. Zip/Postal Code
2. First Name 15. Country
* 3. Last Name 16. Region
* 4. Password 17. Work Phone
* 5. Email Address 18. Home Phone
* 6. Timezone 19. Cell Phone
7. Language 20. Fax
8. Job Title 21. Pager
9. Company Name 22. Area Id
10. Address 1 23. Project Id
11. Address 2 24. Start Page
```

12. City 25. User Group
13. State/Province

User Id => bill.smith, First Name => William
Enter number or q to submit: 3
Last Name? Smith
1. User Id 14. Zip/Postal Code
2. First Name 15. Country
3. Last Name 16. Region
* 4. Password 17. Work Phone
* 5. Email Address 18. Home Phone
* 6. Timezone 19. Cell Phone
7. Language 20. Fax
8. Job Title 21. Pager
9. Company Name 22. Area Id
10. Address 1 23. Project Id
11. Address 2 24. Start Page
12. City 25. User Group
13. State/Province

User Id => bill.smith, First Name => William
Last Name => Smith
Enter number or q to submit: 4
Password?
Confirm Password?
1. User Id 14. Zip/Postal Code
2. First Name 15. Country
3. Last Name 16. Region
4. Password 17. Work Phone
* 5. Email Address 18. Home Phone
* 6. Timezone 19. Cell Phone
7. Language 20. Fax
8. Job Title 21. Pager
9. Company Name 22. Area Id
10. Address 1 23. Project Id
11. Address 2 24. Start Page
12. City 25. User Group
13. State/Province

User Id => bill.smith, First Name => William
Last Name => Smith, Password => *****
Enter number or q to submit: 5
Email Address? bill@extraview.com
1. User Id 14. Zip/Postal Code
2. First Name 15. Country
3. Last Name 16. Region
4. Password 17. Work Phone
5. Email Address 18. Home Phone
* 6. Timezone 19. Cell Phone
7. Language 20. Fax
8. Job Title 21. Pager
9. Company Name 22. Area Id
10. Address 1 23. Project Id
11. Address 2 24. Start Page
12. City 25. User Group
13. State/Province

User Id => bill.smith, First Name => William
Last Name => Smith, Password => *****, Email Address =>
bill@extraview.com
Enter number or q to submit: 6
1. (GMT -12) Eniwetok 15. (GMT 1) Amsterdam
2. (GMT -11) Midway Island 16. (GMT 2) Athens
3. (GMT -10) Hawaii 17. (GMT 3) Moscow
4. (GMT -9) Alaska 18. (GMT 4) Abu Dhabi
5. (GMT -8) Pacific 19. (GMT 5) Ekaterinburg
6. (GMT -7) Mountain 20. (GMT 5:30) Bombay
7. (GMT -6) Central 21. (GMT 6) Colombo
8. (GMT -5) Eastern 22. (GMT 7) Bangkok
9. (GMT -4) Atlantic 23. (GMT 8) Beijing
10. (GMT -3:30) New Foundland 24. (GMT 9) Tokyo
11. (GMT -3) Buenos Aires 25. (GMT 9:30) Darwin
12. (GMT -2) Mid-Atlantic 26. (GMT 10) Sydney
13. (GMT -1) Azores 27. (GMT 11) Magadan
14. (GMT 0) Greenwich 28. (GMT 12) Fiji
Timezone? 5
1. User Id 14. Zip/Postal Code
2. First Name 15. Country
3. Last Name 16. Region
4. Password 17. Work Phone
5. Email Address 18. Home Phone
6. Timezone 19. Cell Phone
7. Language 20. Fax
8. Job Title 21. Pager
9. Company Name 22. Area Id
10. Address 1 23. Project Id
11. Address 2 24. Start Page
12. City 25. User Group
13. State/Province

User Id => bill.smith, First Name => William
Last Name => Smith, Password => *****, Email Address =>
bill@extraview.com
Timezone => -8
Enter number or q to submit: 25

1. Administrator
2. Customer
3. Customer Support
4. Engineer
5. IT Support
6. Quality Assurance

Which groups? 3
1. User Id 14. Zip/Postal Code
2. First Name 15. Country
3. Last Name 16. Region
4. Password 17. Work Phone
5. Email Address 18. Home Phone
6. Timezone 19. Cell Phone
7. Language 20. Fax
8. Job Title 21. Pager
9. Company Name 22. Area Id
10. Address 1 23. Project Id

11. Address 2 24. Start Page
12. City 25. User Group
13. State/Province

```
User Id => bill.smith, First Name => William
Last Name => Smith, Password => *****, Email Address =>
bill@extraview.com
Timezone => -8
Enter number or q to submit: q
User BILL.SMITH inserted.
User BILL.SMITH added to group.
$
```

evcheck

Displays various configuration settings of the CLI Installation

SYNOPSIS

evcheck [-m][-d]

DESCRIPTION

This command checks that the current configuration of the CLI is able to connect to the specified server. It displays the name of the CLI configuration file, the URL of the ExtraView server, the version number and modification time of the Java API class, and the current CLI user's username. If a .evrc file was used for authentication, the pathname of that file will also be displayed.

NOTES

If the -m option is used, evmail settings are also displayed.

If the -d option is used, warning messages are displayed when fields in the data dictionary have the same display title as other fields with different names. These messages are informational only, and do not necessarily indicate an error.

EXAMPLE

```
$ evcheck
config file: /home/bill/evj52_evcli/evcli/evconfig.txt
server: http://avalon.extraview.net/exp/ExtraView/ev_api.action
evapi version: 15
modtime: 8/05/04 11:33a
user: bill.smith - from /home/bill/.evrc
delimiter: :
$
```

evdebug

Specifies the amount of debugging information written to the ExtraView log file

SYNOPSIS

evdebug -n

DESCRIPTION

This command may be used to increase or decrease the amount of debugging information written to the ExtraView application log file. The default debugging level is 6. Values of n may be in the range of 1 through 12; 1 writes the least amount of information to the log file, and 12 writes the most information. Setting the debugging level higher than 6 may reduce end-user performance, as considerably more information gets written to the log file.

evdefaults

Displays fields from the data dictionary that have default values

SYNOPSIS

evdefaults [-r]

DESCRIPTION

This command displays a list of the fixed database names of all the fields in the data dictionary that have default values, with their default values.

Using the -r option, the name and default value are separated by a colon instead of a tab.

EXAMPLE

```
$ evdefaults
STATUS NEW
EDIT_BUTTON EditButton.gif
CSR_CUST_SEVERITY 227
HISTORY_BUTTON HistoryButton.gif
AREA 1
VIEW_BUTTON ViewButton.gif
IT_DATE_REQUESTED 04/10/11
STATUS_HIST UNASSIGNED
DELETE_BUTTON DeleteButton.gif
$
```

evdelete

Delete an existing issue

SYNOPSIS

evdelete [-B <pathname>] id1 [. . . idn] [idx-idy]

DESCRIPTION

This command allows users to delete one or more issues from the ExtraView database. Once an issue is deleted, there is no way to access that issue from either the CLI nor the web interface. However, the history of the issue is not deleted. Contact ExtraView support if a deleted issue needs to be recovered. The file(s) to be deleted can either be specified on the command line, or may be stored in a file. Use one of the three options with any invocation of the command, i.e. either delete issues from a file, from a list of issues entered on the command line, or from a range of issues entered on the command line.

-B <pathname>

<pathname> points to a file containing a list of issue numbers to be deleted, one issue per line. The filename must end in **.txt** and the full path where the file is stored must be given.

id1 [id2 ... idn]

One or more individual issues may be specified on the command line, separated by white space.

[idx-idy]

A range of issues may be specified, e.g. 21345-21357. USE CAUTION WITH RANGES - it is very easy to accidentally delete a large number of issues.

All ID's between the starting and the ending numbers must exist for this command to function correctly. If you are not certain of the existence of all ID's within the range, you can write a script that deletes the issues one-by-one in a loop. For example, in Windows, a script like this will do the job:

FOR /L %i IN (start,step,end) DO evdelete %i

You can write a similar shell script under UNIX or Linux.

NOTES

Write permission for the PR_RESOLUTION.DELETE_BUTTON security key is required to successfully run this command.

EXAMPLE

```
$ evdelete 12352
Issue # 12352 deleted.
$
```

The term Issue # in the example is replaced with the data dictionary title for the field named ITEM_ID.

If the issue you attempt to delete does not exist, the message "Issue # nnnnn does not exist" is returned.

evdeleteuser

Deactivate an active user account

SYNOPSIS

evdeleteuser user_id

DESCRIPTION

This command deactivates the user account user_id. The user_id may be the alternative user ID for the account. The ExtraView never deletes user accounts. When an account is deactivated, that user may not sign in any longer, and that username cannot be saved as the value for any field with the display type of User. The reason accounts are not deleted is that there may be historic data attached to a user account and the user's details are required when accessing historic information.

Issues are unaffected by deactivating a user. User fields may contain the name of a deactivated user. However, user fields can no longer be assigned a deactivated username.

NOTES

Write permission for the SE_SECURITY_USER security key is required to successfully run this command.

evdownload

Get a copy of an existing issue attachment from the server

SYNOPSIS

evdownload [[-d] [-r] | -a] ID filename [newfilename|-] | [[-d] -B idfile]

DESCRIPTION

This command allows users to download a local copy of an attachment from the server.

The ID must be a valid issue ID, and the issue must have one or more attachments available to download.

The -d option displays just the filenames of the files attached to the issue ID.

The -r option displays this same information in raw format, plus an additional field (the Attachment ID), in a colon-separated list, containing one attachment per line, without a column header line. The fields are also returned in a different order.

The -a option downloads all the attachments for the specified ID.

The -B idfile option allows the user to create a file of issue ID's, with one ID per line, and all the attachments for all the issue ID's are downloaded.

filename must be the file name of an existing attachment stored within the issue # ID

If newfilename is specified, the attachment will be saved with this new name, instead of saving it with it’s original name, filename.

If a minus character (-) is specified as the final command line option, the attachment will be printed on

standard out, instead of saving it to a local file. This is typically used for text that you want to display on your terminal.

If more than one attachment exists with the name, filename, a table will be displayed with the file descriptions, file sizes, dates, and owners, of all instances of filename, and the user is asked to choose which file to download.

NOTES

A complete list of attachments for any issue can be displayed with the evfiles command.

EXAMPLE

```
$ evdownload 12345 abc.xls newname.xls
$
```

eveditlist

Displays the list of fields used on the edit screen for the current Business Area and Project

SYNOPSIS

eveditlist [-r]

DESCRIPTION

This command displays the list of fields that are used when editing an issue in the ExtraView database. The field names are displayed in two columns. The first column is the field's fixed database name. The second column is the field's display title. The field names are printed out in the order that they appear on the ExtraView Edit screen, top to bottom, left to right.

When the -r option is used, the fixed database names and display titles are each terminated by a colon, with no white space in between the two names.

There may be one or more characters preceding each line. These have the following meanings –

Character	Meaning
*	This indicates the field can span multiple lines when being input
~	This indicates that the field is a member of a repeating record
%	This indicates that the field has a display type of user
^	This indicates that the field supports multiple values
+	This is a field with a display type of multi-value list

Additionally, when a field has an allowed value parent, the fixed database name of the parent field is printed after the child field's colon-terminated display title. This parent field is not terminated by a colon.

EXAMPLE

```
$ eveditlist -r
MODULE_ID:Module:PRODUCT_NAME
DEV_DOC_CHANGE:Doc Change:
+*DOCUMENT_IMPACT:Documentation Impact:
SIMILAR_SEARCH:Search for Similar:
PRIORITY:Priority:
+PRODUCT_NAME:Product:
SEVERITY_LEVEL:High:
+SHORT_DESCR:Title:
~RELEASE_FIXED:Committed Release:
+STATUS:Open:
DEV_REPRODUCIBLE:Reproducible:
*COMMENTS:Internal Comments:
CATEGORY:Category:
*DEV_RELEASE_NOTES:Release Notes:
RESOLUTION:Resolution:
RELATED_ISSUES:Related Issues:
PARENT_ID:Parent ID:
~RELEASE_FOUND:Requested Release:
%ASSIGNED_TO:Assigned To:
*DESCRIPTION:Description:
DEV_RELEASE_FOUND:Release Found:
~RELEASE_STATUS:Release Status:
*DEV_TEST_CASE:Test Case:
$
```

evfields

Display a list of all data dictionary fields and their title

SYNOPSIS

evfields [-r | -g]

DESCRIPTION

This command displays a list of all fields in the ExtraView. The field names are displayed in two columns. The first column is the field's fixed database name, the second column is the field's display title. The field names are alphabetized by display titles.
When the -r option is used, the fixed database names and display titles are separated by a colon instead of a tab, and the list is not alphabetized. Additionally, one or more characters may precede the field name with their meaning as follows:

Character	Meaning
*	This indicates the field can span multiple lines when being input

~	This indicates that the field is a member of a repeating record
%	This indicates that the field has a display type of user
^	This indicates that the field supports multiple values
+	This is a field with a display type of multi-value list

The -g option generates two identical files named get.txt and search.txt, which may be used by the evget and evsearch commands, respectively. These files are saved in the directory where the CLI commands are stored, if the user has write access to that directory.

If the user does not have write access to that directory, the files are saved in the users home directory as defined by the environment variable, $HOME. The -g option will be ignored if the -r option is used.

These files contain only the fixed database names, and names are not capitalized in these files. This does not have any functional effect; it is just a different format. The last line of these two files is a line containing, __END__. When evget or evsearch read these files, they will ignore this line and anything after it.

NOTES

This command displays field names independent of any layouts. As such, the concept of required fields does not exist in this command. To see which commands are required on the add screen or edit screen, use the command evaddlist –r or eveditlist –r, respectively, and look for lines beginning with a plus sign (+).

EXAMPLE

```
$ evfields
user * Current User Name *
promo
active Active
alt_id Alt ID
assigned_to Assigned To
attach_content_type Attachment Content Type
attach_created_by_user Attachment Created By
attach_date_created Attachment Date Created
attach_file_desc Attachment File Description
attach_file_name Attachment File Name
attach_file_size Attachment File Size
attachment_id Attachment ID
attach_path Attachment Path
attachment Attachments
area Business Area
cc_email CC Email
category Category
last_change_user Changed by
date_code_freeze Code Freeze
release_fixed Committed Release
company_name Company
contact Contact
start_date Created Start Date
```

stop_date Created Stop Date
sysdate Current Date
sysday Current date at midnight
custom_email Custom Email
customer Customer
date_closed Date Closed
date_created_trunc Date Created
date_created Date Created
date_last_status_change Date of Last Status Change
date_created_day Day Created
timestamp_day Day Last Updated
days_open Days Open
date_closed_since Days Since Closed
date_created_since Days Since Created
date_last_status_change_since Days Since Last Status Change
timestamp_since Days Since Last Updated
days_in_status Days in Status
problem_release_delete Delete
description Description
document_impact Documentation Impact
release_directory Download File Directory
due_date Due by
email E-Mail Address
email_address Email Address
administration ExtraView Administration
search_report ExtraView Search / Report
sign_on ExtraView User Sign On
add_problem Extraview Add Issue
file_size File Size
filter_child_values Filter Children
date_first_customer_ship First Customer Shipment
generate_email Generate Email
severity_level High
product_name_hist Historical Product Reference
release_found_hist Historical Release Reference
status_hist Historical Status
item_id ID #
id ID #
comments Internal Comments
problem_release_id Internally used Id in the problem_release_view
problem Issue
problem_summary_edit Issue Summary Edit
keyword Keywords
timestamp Last Modified
timestamp_trunc Last Modified
layout Layout
layout_type Layout Type
mailing_list Mailing List
module_id Module
module_name Module
module_title Module
module_assigned Module Assigned
module_date_created Module Date Created
module_product Module Product
module_status Module Status
module_timestamp Module Timestamp
module_type Module Type

date_created_month Month Created
timestamp_month Month Last Modified
months_open Months Open
months_in_status Months in Status
add_problem_summary New Issue Summary
open_assigned_to Open Status Assigned To
open_originator Open Status Originator
open_owner Open Status Owner
originator Originator
owner Owner
parent_id Parent ID
priority Priority
privacy Privacy
product_name Product
product_line Product Line
project Project
quick_list Quick List
date_release_to_qa RQA
page_size Records per Page
relationship_group Relationship Group
relationship_group_id Relationship Group
relationship_group_link Relationship Group Link
relationship_group_owner Relationship Group Owner
relationship_grp_parent_id Relationship Group Parent Problem ID #
relationship_group_title Relationship Group Title
relationship_group_type Relationship Group Type
release Release
release_assigned_to Release Assigned To
release_date_created Release Date Created
release_doc_filename Release File
dev_release_found Release Found
dev_release_notes Release Notes
release_owner Release Owner
release_priority Release Priority
release_product Release Product
release_resolution Release Resolution
release_severity Release Severity
release_status Release Status
release_type Release Type
report Report By
report_by Report By
report_output Report Output
report_type Report Type
release_found Requested Release
resolution Resolution
similar_search Search
for Similar
sort Sort By
report_name Sort Order
spell_check Spell Check
status_change State Change Rules
status Status
system_log_type System Log Types
security_permission System Object Access
security_keys System Security Object Summary
short_descr Title
start_update Updated Start Date

```
stop_update Updated Stop Date
user_accounts User Accounts Summary
udf User Defined Fields
security_group User Groups Summary
module_version Version
date_created_week Week Created
timestamp_week Week Last Modified
weeks_open Weeks Open
weeks_in_status Weeks in Status
date_created_year Year Date Created
timestamp_year Year Timestamp
home_page ExtraView Home Page
$
```

evfiles

Display file attachment information for a given Issue ID

SYNOPSIS

evfiles [-r] ID

DESCRIPTION

This command displays a tab-formatted table, with column headers, containing a set of metadata for every file attached to the specified issue. This metadata includes Attachment Created date, Attachment File Size, Attachment Created By, Attachment File Name, and the Attachment File Description.

The ID must be a valid issue ID.

The -r option displays this same information, plus an additional field (the Attachment ID), in a colon-separated list, containing one attachment per line, without a column header line. The fields are also returned in a different order:

Created date:File name:File size:Created by:Attachment ID:File Description

EXAMPLE

```
$ evfiles 12345
Created File Size Owner Name Description
======= ========== ===== ==== ===========
01-Jan-2001 1.2MB jstorr spec.xls New GUI layout
07-Feb-2001 3KB mdrewar listbutton.gif New button
$
```

evget

Displays the data for one issue

SYNOPSIS

evget [-r] [-c "field1, field2, ... fieldn"] [-f myfile.txt] [-a] ID

DESCRIPTION

This command displays field data for a single issue. By default, all fields on the **Detailed Report** layout to which a user has read permission, will be displayed. The **Detailed Report** for the user's current Business Area and Project is selected. There are three ways of specifying a subset of these fields to be displayed -

1. The -f option
2. The -c option
3. The existence of a file named $HOME/.evget (evget.txt on Windows). However, only fields on the **Detailed Report** layout can be displayed with this command.

NOTE: Even though attachments may be displayed on the Detailed Report layout, evget cannot display attachments.

If the file $HOME/.evget exists (evget.txt on Windows), and neither the -c option nor the -f option are specified, evget will use the contents of this file as the list of field names whose values will be displayed. The format of this file is:

- Only one field name may be placed per line in this file
- Only fixed database names may be used in this file
- Blank lines are ignored
- Lines whose first non-blank character is a hash (#) are ignored
- A line containing only the string, __END__ marks the end of the field list. This line, and all lines after this line in the file are ignored.

OPTIONS

-r

Data for all fields is returned in XML format. All other command line options are ignored when -r is used.

-c "field1, field2, ... , fieldn"

Allows the user to provide a comma-separated list of fields to be displayed, as command line options. Field names must be identified using the fixed database names. If the -f option is used in combination with the -c option, all the fields included in myfile.txt will be displayed in addition to the fields specified with the -c option. When the -c option is used, the file $HOME/.evget is ignored.

-f myfile.txt

Allows the user to specify a file that contains a list of all the fields to be displayed. The format of this file is identical to the format of the file $HOME/.evget, described above. When the -f option is used, the file $HOME/.evget is ignored.

-a left|normal

Controls the alignment of the displayed data. With -a normal, the colons which separate the display titles from their values are aligned vertically. With -a left, all the output is left justified.

ID

Must be a valid Issue number.

NOTES

Use the evfields command with the -g option to generate a sample get.txt file. This file may then be modified and saved as $HOME/.evget, to specify the default fields to be displayed by evget.

EXAMPLE 1

```
$ evget 10200
Assigned To: Internal IS Support
Category:
Changed by: Bill Smith
Comments: 6/30/04 System Administrator
This needs to be fixed quickly
Date Created: 1/7/04
Description: Round corners, square edges, measure once
Documentation Impact:
ID #: 10200
Last Modified: 7/23/04
Operating System:
Originator: Guest User
Priority: P 3
Product: Tracker Enterprise
Reproducible:
Resolution:
Severity: Low
Status: Open
Title: Metal mounting bracket is too short
$
```

EXAMPLE 2

```
$ evget -r 10200
<?xml version="1.0" encoding="UTF-8"?>
```

]]>

This needs to be fixed quickly

```
]]>

]]>


$
```

evgetfields

Displays a set of field values for one issue

SYNOPSIS

evgetfields ID field1 [field2 ... ... fieldn]

DESCRIPTION

ID must be a valid issue number

Fieldn must be a fixed database name to which the user has read or write access.

This command displays a specific set of field values for an existing issue, one field per line. If a multi-line field is requested, the second and subsequent lines will be indented by two spaces. Neither fixed database names nor display titles are printed, only the field values. If a field has no value assigned, a blank line is printed.

This command is not limited to only accessing fields on the **Detailed Report** layout for the user's current Business Area and Project, as is the case with the evget command. This command will display the value of any field for which the user has read permissions.

NOTES

evfields generates a list of valid fields that can be used with this command.

EXAMPLE

```
$ evgetfields 10200 status priority description
OPEN
P3
Round corners, square edges, measure once
$
```

evheartbeat

This command checks the status of an ExtraView database, to indicate whether it is functioning correctly.

SYNOPSIS

evheartbeat [-r] [-m]

DESCRIPTION

This command checks the status of several components of the ExtraView installation, and provides immediate feedback to the user. Additionally, this command is designed to send optional email output, so that a system administrator can be automatically notified if there is a system outage.

The command may be configured with several entries in the evconfig.txt file as shown in this example:

```
HB_SMTP_SERVER = .com
HB_TO = you@<your-domain>.com[,your-buddy@<your-domain>.com]
HB_FROM = someone@thisplace.com
```

- The -r option outputs the results from the command in raw XML format.

  Note: The XML returned also includes tags named DB_DATETIME, FREE_MEMORY, TOTAL_MEMORY, SERVICE_COUNT and HEARTBEAT_EXEC_TIME along with values. This information is useful if you are developing scripts that need to accurately know the current time of the server or scripts that monitor the performance of ExtraView running on the server. The time returned is the time on the database server. The memory statistics are in megabytes and the execution time is in milliseconds

- The -m option signifies that email is to be sent when the command is run, and the command detects an error condition in the ExtraView installation. In this case, the above evconfig.txt options must have been set. No email is sent if the ExtraView installation returns a success code. The intention is that evheartbeat -m is used on a job, to monitor the health of the ExtraView installation. It is suggested that this script is run at least every 15 minutes throughout the day and night.

If the configuration file has not been set correctly, then a message or warning is displayed on the screen, informing the user of the error.

When sending email, evheartbeat places a header into the email message, entitled ExtraView-Audit.

```
ExtraView-Audit:
Sent by system user <login-name-of-sender>,
from host <name-of-host-running-evheartbeat> (<ip-address-of-host>),
on Tue Sep 30 17:04:17 2003, client program == evheartbeat.
EXAMPLE 1
$ evheartbeat

Heartbeat Results
-----------------
ExtraView status : EXTRAVIEW ALIVE
DataBase status : DB CONNECTION CONFIRMED
DataBase Timestamp: 2004-10-11 22:15:09.000 -0700

$
```

EXAMPLE 2

```
$ evheartbeat -r
<?xml version="1.0" encoding="UTF-8"?>
```

```
<EV_HEARTBEAT>
<EV_STATUS>EXTRAVIEW ALIVE</EV_STATUS>
<DB_STATUS>DB CONNECTION CONFIRMED</DB_STATUS>
<DB_DATETIME>2004-10-11 22:15:22.000 -0700</DB_DATETIME>
<FREE_MEMORY>106</FREE_MEMORY>
<TOTAL_MEMORY>128</TOTAL_MEMORY>
<SERVICE_COUNT>1</SERVICE_COUNT>
<HEARTBEAT_EXEC_TIME>396</HEARTBEAT_EXEC_TIME>
</EV_HEARTBEAT>

$
```

evhelp

Display on-line help for a command

SYNOPSIS

evhelp [-l | <command>]

DESCRIPTION

This command will display on-line help for any CLI command, and for a small set of additional subsections.

-l only displays a list of the available commands.

<command> displays the on-line help for <command>. This is the same information that is printed when using the -H option with all CLI commands. <command> may also be the name of one of the subsections described by evhelp -H.

EXAMPLE

```
$ evhelp evgetfields
evgetfields
This command provides the values for specific fields enumerated in a list,
for an existing problem in the ExtraView database.
Syntax:
evgetfields ID field1 field2 ... ... fieldn
Notes:
* The ID must exist in the ExtraView database.

* Each field in the list must exist.

* The command evfields generates a list of valid fields that can be used with
this command.

The following example retrieves the status, priority and description for a
problem.

$ evgetfields 12345 status priority description
Sumitted
```

High
This problem can be seen when you set the control to setting A.
$

If the query returns a null value for a field, there will be an empty line in the output.
$

evhist

Display field data for issues modified since the specified date. The primary purpose of this command is to provide an efficient and convenient method of providing all the changed data since a specific date (and time), so that this data may be used to load another database, such as a company's data warehouse. The options on the command allow the dumping of a specific field list, using a specific set of filters.

SYNOPSIS

evhist date [time] [-i] [-c field1,field2, ... fieldn] [-f myfile.txt] [-p reportID] [field1=value1[;value2[;valuen]]] [field2=value1[;value2[;valuen]]] ... ... [q]

DESCRIPTION

This command returns all field data (for which the user has read permission), for all issues that have been modified since the specified date, in XML format. The default list of fields that is dumped is the fields on the detailed report layout for the user's current area and project. The list of fields can be modified by using your own field list as parameters to the command. In addition, you can set a set of filters. For example, you may return only the data for a specific business area, or for a specific product.

ID numbers, and 'last modified' dates are also returned for any issues that were deleted since the specified date.

The format for date and time within the output is not the current user's own date / time format, but always uses the following format –

yyyy-MM-dd HH:mm:ss.SSS ZZZZ

As an example, this is a valid date and time to use to extract information from midnight on May 1st 2009 in the pacific standard time zone:

2009-05-01 00:00:00.0 PST

This ensures that a script created by one user can be used within other scripts and CRON jobs, so that many other users can share the output. It also ensures that the format for date and time is unambiguous and can be used by other systems that may read and interpret the XML in the command output. This format is also known as ExtraView internal timestamp format.

In using this command to output data for use within a separate data warehouse you should take care to prepare a list of the fields needed, as opposed to outputting all the fields. In most warehousing applications, you will not require the text area, log area and other similar fields within ExtraView, yet these fields may occupy more than 90% of the volume of data in an issue. The time to execute a data dump from ExtraView with the evhist command may be dramatically reduced if you only output the fields required in the warehouse.

Similarly, the filters you apply to the command allow you to only output the data for any specific criteria you require. Typically this will be for a business area, or project, but it may be any other segment of data, such as the data for a specific product.

Search Criteria

The user may specify search criteria in two ways, either on the command line, or through an interactive menu.

On the command line, the user may specify fields and field values to use as search criteria. If the command line is terminated by the letter q, the export will be executed immediately. If the command line does not terminate with the letter q, any search criteria specified on the command line will be passed along to the interactive menu, for additional filter criteria specification.

By default, command-line-specified fields must be identified by their Display Titles. For fields with a display type of List, values must be identified by their display titles. When the -i option is used, fields are identified by their fixed database names, and list values must be identified by their fixed names (for non-UDF lists), or by their internal list ID's (for UDF lists). The evmeta command displays both of these list names for all list items.

A search may be performed for more than one value of a field by using a semi-colon to separate the values. For example:

STATUS=OPEN;FIXED;CLOSED

will search for issues with a status of Open, Fixed or Closed.

List of Fields to Export

There are three ways of specifying the list of fields to be displayed -

1. The -f option
2. The -c option
3. The existence of a file named $HOME/.evsearch (evsearch.txt on Windows). One of these three approaches must be used. If no field list is provided, no field data will be displayed.

By default, only fields on the **Detailed Report** layout for the user's current Business Area and Project, and to which the user has read permission, may be requested. You cannot request fields that do not appear on the detailed report for the currently selected business area, project and user role. However, the -p option may be used to specify a different report to be used as the source of field data.

Note: Even though attachments may be displayed on a report layout, evhist does not export attachments.

If the file $HOME/.evsearch exists, and neither the -c option nor the -f option is provided, evsearch will use the contents of this file as the list of field names whose values will be displayed. The format of this file is:

- This file may only contain printable, ASCII characters
- Tabs, spaces, and blank lines are ignored
- Only one field name may be placed per line
- Only fixed database names may be used
- Lines whose first non-blank character is a '#' are ignored
- A line containing only the string, __END__ marks the end of the field list. This line, and all lines after this line in the file are ignored

OPTIONS

-c "field1, field2, ..., fieldn"

Allows the user to provide a comma-separated list of fields to be displayed, as command line options. Field names must be identified using the fixed database names. If the -f option is used in combination with the -c option, all the fields included in myfile.txt will be displayed in addition to the fields specified with the -c option. When the -c option is used, the file $HOME/.evsearch is ignored.

Again, the fields selected must exist on the detailed report for the currently selected business area and project for the user.

field1=value1[;value2[;valuen]]] [field2=value1[;value2[;valuen]]] ...

This is a set of name value pairs that specify the filters to be used as search criteria for the output. You may either have a simple name value pair, such as field1=value1, or you can use the longer form field1=value1;value2;valuen to set up a search that performs an "or" operation on all the values provided.

-f myfile.txt

Allows the user to specify a file that contains a list of all the fields to be displayed. The format of this file is identical to the format of the file $HOME/.evsearch, described above. When the -f option is used, the file $HOME/.evsearch is ignored. As with the -c option, the fields selected must exist on the detailed report for the currently selected business area and project for the user.

-p reported

By default, evhist only retrieves field data for fields that are on the **Detailed Report**. This option allows the user to specify an alternate Report, from which field data may be selected.

EXAMPLE

```
$ evhist 2004-10-11 12:00:00
<?xml version="1.0" encoding="UTF-8"?>
<HISTORY>
<PROBLEM_RECORD>
<ID TITLE="ID #"><![CDATA[10205]]></ID>
<DATE_CREATED TITLE="Date
Created"><![CDATA[8/25/04]]></DATE_CREATED>
<TIMESTAMP TITLE="Last
Modified"><![CDATA[10/11/04]]></TIMESTAMP>
<CATEGORY
TITLE="Category"><![CDATA[Documentation]]></CATEGORY>
<ORIGINATOR TITLE="Originator"><![CDATA[Carl
Koppel]]></ORIGINATOR>
<LAST_CHANGE_USER TITLE="Changed by"><![CDATA[Carl
Koppel]]></LAST_CHANGE_USER>
<SHORT_DESCR TITLE="Title"><![CDATA[This is a new
issue]]></SHORT_DESCR>
<PRODUCT_NAME
TITLE="Product"><![CDATA[Tracker]]></PRODUCT_NAME>
<PRIORITY TITLE="Priority"><![CDATA[]]></PRIORITY>
<STATUS TITLE="Status"><![CDATA[Fixed]]></STATUS>
<ASSIGNED_TO TITLE="Assigned To"><![CDATA[Carl
Koppel]]></ASSIGNED_TO>
<RESOLUTION TITLE="Resolution"><![CDATA[]]></RESOLUTION>
<OS TITLE="Operating System">AIX:Mac</OS>
```

```
<SEVERITY_LEVEL
TITLE="Severity"><![CDATA[Low]]></SEVERITY_LEVEL>
<DEV_REPRODUCIBLE
TITLE="Reproducible"><![CDATA[No]]></DEV_REPRODUCIBLE>
<DEV_PERFORCE_JOB_NAME TITLE="Perforce Job
Name"><![CDATA[]]></DEV_PERFORCE_JOB_NAME>
<DESCRIPTION TITLE="Description"><![CDATA[
]]></DESCRIPTION>
<COMMENTS TITLE="Comments"><![CDATA[

]]></COMMENTS>
<DOCUMENT_IMPACT TITLE="Documentation Impact"><![CDATA[
]]></DOCUMENT_IMPACT>
<FULL_TIMESTAMP>2004-10-11 22:25:15.000 -0700</FULL_TIMESTAMP>
</PROBLEM_RECORD>
<PROBLEM_RECORD>
<ID TITLE="ID #"><![CDATA[10200]]></ID>
<DATE_CREATED TITLE="Date
Created"><![CDATA[1/7/04]]></DATE_CREATED>
<TIMESTAMP TITLE="Last
Modified"><![CDATA[10/11/04]]></TIMESTAMP>
<CATEGORY TITLE="Category"><![CDATA[Hardware]]></CATEGORY>
<ORIGINATOR TITLE="Originator"><![CDATA[Guest
User]]></ORIGINATOR>
<LAST_CHANGE_USER TITLE="Changed by"><![CDATA[Bill
Smith]]></LAST_CHANGE_USER>
<SHORT_DESCR TITLE="Title"><![CDATA[Metal mounting bracket is too
short]]></SHORT_DESCR>
<PRODUCT_NAME TITLE="Product"><![CDATA[Tracker
Enterprise]]></PRODUCT_NAME>
<PRIORITY TITLE="Priority"><![CDATA[P 3]]></PRIORITY>
<STATUS TITLE="Status"><![CDATA[Open]]></STATUS>
<ASSIGNED_TO TITLE="Assigned To"><![CDATA[Internal IS
Support]]></ASSIGNED_TO>
<RESOLUTION TITLE="Resolution"><![CDATA[]]></RESOLUTION>
<OS TITLE="Operating System"></OS>
<SEVERITY_LEVEL
TITLE="Severity"><![CDATA[Low]]></SEVERITY_LEVEL>
<DEV_REPRODUCIBLE
TITLE="Reproducible"><![CDATA[No]]></DEV_REPRODUCIBLE>
<DESCRIPTION TITLE="Description"><![CDATA[Round corners, square
edges, measure once
]]></DESCRIPTION>
<COMMENTS TITLE="Comments"><![CDATA[6/30/04 System Administrator
This needs to be fixed quickly
]]></COMMENTS>
<DOCUMENT_IMPACT TITLE="Documentation Impact"><![CDATA[
]]></DOCUMENT_IMPACT>
<FULL_TIMESTAMP>2004-10-11 22:25:38.000 -0700</FULL_TIMESTAMP>
</PROBLEM_RECORD>
</HISTORY>
$
```

evimport

This command enables the user to import a tab-delimited file from which records can be parsed and added to the database.

SYNOPSIS

evimport [-c] [-n] [-a attachment_dir -o field_name] file.txt

evimport -u file.txt

evimport -ui file.txt

evimport -n file.txt

NOTES

- Reads in a text file composed of tab delimited fields. Saving a spreadsheet in a tab-delimited format typically composes this file
- The first (non blank non comment) line contains titles for database items that must match exactly what is returned from evaddlist. This line specifies which column should be inserted into which database field inside ExtraView
- The rest of the tab-separated rows in the file are read and an issue record is created within ExtraView for each row. If a line contains only __END__ all lines after it are ignored. This allows for storing titles and other values without affecting the running of the command.
- If a title in the first line begins with an asterisk (*), that column is skipped entirely. This is useful in case the original file has additional columns of data irrelevant to the import
- If a line contains only the text __END__, then this line and all following lines will be ignored
- Blank lines and those beginning with a hash mark (#) are ignored
- If one or more records cannot be imported because of data errors, they are written to a file named errors.txt. This file may be edited, renamed and used again with the evimport program. Note that the same filename is used for each set of errors, so be sure to rename this before running the command again
- The -n option suppresses the sending of email to users as each issue is entered into the database
- When the -c option is used, the file is processed, without updating ExtraView. This is good for error verification before actually doing the inserts. Typical errors are values that are not valid for an unknown release or category
- The -u option presents a list of titles to the user and asks which ones you wish to find unique fields for. Then the file is parsed in its entirety. The parser will search for uniquely occurring values within the requested fields. This is useful for preparing ExtraView metadata before doing the actual evimport
- The -ui option is similar to the -u option, except that after showing the set of unique values, the program asks if the invalid ones (marked with an asterisk) should be inserted into ExtraView as valid values for this field. After saying y to all of these questions evimport -c should succeed
- If the data being imported contains text fields with multiple lines, the end of line character(s) within the import file must be changed before the import is commenced. If this is not done, the end of the first line is interpreted as the end of the record. To overcome this, change the end of line character(s) within the text fields to <<EV_DELIM_EOL>>. The evimport command will alter the data as it is being imported to reinstate the new lines within the data. The <<EV_DELIM_EOL>> value can be altered with an assignment, to be a different character string like so:

  EOL = **

- If the data being imported contains tab characters they can be handled in a similar fashion to the EOL character. If a field value contains <<EV_DELIM_TAB>>, this string or an equivalent value will be translated into a tab character as the import is performed. To replace the <<EV_DELIM_TAB>> value, use:

TAB = REPLACE_ME_WITH_A_TAB

- The -a option is always used with the -o option. Should these be present, they indicate that attachments will be uploaded with the issues in file.txt. The attachment_dir is the name of the directory where the attachments are stored. Within this directory, there will be sub-directories whose name is derived from the value of the import field, field_name. For example, if the attachment_dir is /mydir/attachments, and the import field is this_field then the attachment files for the issue will be found under the directory named /mydir/attachments/this_field

evimportJIS

This command enables the user to import a tab-delimited Japanese language text file from which records can be parsed and added to the database.

SYNOPSIS

evimportJIS [-c][-u][-d] file.txt fields.txt

NOTES

Reads in a Shift JIS text file composed of tab delimited fields. This file is typically composed by saving a spreadsheet in a tab-delimited format from a Japanese OS or computer and saving the .txt file on that computer. It may be imported from an English OS computer once it is saved from a Japanese OS.

The tab-separated rows in the file are read and an issue record is created within ExtraView for each row. If a line contains only __END__ all lines after it are ignored. This allows for storing titles and other values without affecting the running of the command. Blank lines and those beginning with a hash mark (#) are ignored.

If a title in the first line begins with an asterisk (*), that column is skipped entirely. This is useful in case the original file has additional columns of data irrelevant to the import.

The -c option is a check option that allows the file to be checked for integrity, without adding the records to the database.

The -u option presents a list of titles to the user and asks which ones you wish to find unique fields for. Then the file is parsed in its entirety. The parser will search for uniquely occurring values within the requested fields. This is useful for preparing ExtraView meta-data before doing the actual import.

The -d option allows the user to debug the file being imported. It will display messages when an error is encountered from within the API.

evimportav

This command imports sets of allowed values (parent-child values) from a file into the ExtraView database.

SYNOPSIS

evimportav -file av_filename -area area_id -project project_id -parent parent_field_name -child child_field_name

NOTES

-file av_filename specifies the name of the file containing the allowed value combinations. The file to be imported is tab-delimited and is similar to the following example:

parent_value1	child_value1
parent_value1	child_value2
parent_value2	child_value3
parent_value2	child_value4

-area area_id specifies the area_id into which the allowed values will be imported

-project project_id specifies the project_id into which the allowed values will be imported

-parent parent_field_name specifies the data dictionary name of the parent field which has the allowed value relationship with the specified child

-child child_field_name specifies the data dictionary name of the child field which has the allowed value relationship with the specified parent.

evmail

Processes responses emailed to ExtraView

SYNOPSIS

evmail [-n] [-r]

DESCRIPTION

With the inception of ExtraView 6.1, this command has been redeveloped as an internal, server-based ExtraView task that resides within the administration user interface. It is recommended that you use the new method for future development, although for backwards compatibility, the evmail CLI command will remain functional.

This command will process email messages that were sent to either create a new issue or in response to an ExtraView email notification. The body of the email is typically entered into the DESCRIPTION field of the issue created. An email message sent in response to an ExtraView email notification will be added into the COMMENTS or other field of the issue that triggered the initial notification. In addition, other fields may have values set with this command, such as setting the status of an issue, or setting an assigned to user. A reply may be sent to the submitter of the email, containing verification of the mail they sent.

The evmail utility can be installed on the same server as the ExtraView application, on the POP3 email or any other server on your network. Normally, this command will be setup as a job on a UNIX-based server, or as an AT command on Window-based server. External to ExtraView, you must have a POP3 mail server, and a dedicated POP3 mail account.

Attachments to email messages will be added as attachments to the issue being created or updated. The -n flag stops ExtraView from sending mail, on both insert and update operations.

The -r flag is used to send a reply to the sender of the email received by ExtraView. Typically this is an

acknowledgement of their submission of a new issue. Email is only sent as a reply if the email "from" and the email "to" addresses are different to avoid recursive, spoofed emails. The email body is defined in the evconfig.txt file, within the section named REPLY_TEMPLATE.

If the evmail command fails for any reason when it attempts to update the ExtraView database, an email reply is sent to the submitter, giving the reason for the failure. The email body is defined in the evconfig.txt file, within the section named SMTP_FAIL_MSG_FNAM.

The following table describes the entries that are used within evconfig.txt:

POP3_USER	The name of the email account used as an address for ExtraView
POP3_PASSWORD	The password of the email account
POP3_SERVER	The address of the POP3 server
POP3_BODY_UDF	This is the field into which the body of the email is placed, when a new issue is inserted into ExtraView. The field named DESCRIPTION is typically used to receive the text from the body of the incoming email
POP3_BODY_UPDATE_UDF	This is the field into which the body of the email is placed, when an issue is updated. The default for this is the field named COMMENTS, but it can be set to any field with a display type of text area, log area or print text
POP3_fieldname	A field name in ExtraView's data dictionary to which data will be added. These fields can be inbuilt or user defined fields. Note that if such entries exist, then the Perl code in EV::Mail::Message.pm, must be altered to support this field and the value associated with it. Contact ExtraView support with questions about making changes within EVMailMessage.pm
EMAIL_SUBJ_REGEX	The regular expression to be parsed on the subject line of the email. The first captured sub-expression is interpreted as the issue ID. EMAIL_SUBJ_REGEX is used to detect whether the message is an issue update and to determine the ID of the issue to be updated
DELIM_REGEX	The delimiter used within the body of the email. Text following this delimiter is discarded.
REPLY_TEMPLATE	This setting contains the filename of text that is returned to the submitter of the mail. The file may contain tags, i.e. references to fields within ExtraView. evmail will replace the tags with the values from the issue that was created or updated. For example, a tag __ID__ will place the issue number into the mail body, while a tag of __STATUS__ would enter the value of the field STATUS into the mail.
SMTP_FAIL_MSG_FNAM	This setting contains the filename that contains a failure message that is returned to the submitter, if evmail could not complete its insert or update to ExtraView. Within the file, the tags named __REASON__ and __BODY__ can be used with the template as the placeholder for the specific failure reason

All actions taken by this command are logged to a file named mail_log.txt in the $EVDIR directory.

If the user name in the "from" address of the email matches the email address of a valid user in ExtraView, the ORIGINATOR field in ExtraView is set to that user.

If the email contains one or more file attachments of any type, the attachment(s) will be inserted into the issue.

To set up and configure evmail, you will need to identify the user role, business area and project that email-submitted issues will belong to. For example, you may decide that emails entered into ExtraView will be created as Customer Support issues, and will be entered into the system as if a user in the Customer role had signed in and created the issue - without the requirement of every potential Customer user needing a unique user ID within ExtraView.

To achieve this, create a Customer account, and sign in with that user name. Go through the process to add a new issue as this user. Identify the fields that are required, and the values that should be set for issues that are entered via email (e.g. Status = Submitted, Priority = Medium).

Once you have identified the fields and the values that you wish to set, the next step is to set those options in the evmail configuration file. If you are self-hosted, you will do this yourself. If you have chosen ExtraView's ASP hosted solution, contact ExtraView support with this information and this will be set up for you.

The following is a sample configuration of evmail from the config.txt file.

```
SERVER = www.my_company.com/evj/ExtraView
POP3_USER = evmailuser
POP3_PASSWORD = my_password
POP3_SERVER = mail.my_company.com
EVMAIL_ID_REGEX = [ExtraView-(d+)]
DELIM_REGEX = ----Original Message-----
SMTP_SERVER = mail.my_company.com
SMTP_FROM = support@my_company.com
SMTP_CC = int-support@my_company.com
SMTP_FAIL_MSG_FNAM = tmpl8.txt
#
# tmpl8.txt might contain something like:
#
#__REASON__
#-------- Original Message --------
#__BODY__
#
# evmail will substitute EVs failure
# message for __REASON__, and the body
# of the email for __BODY__.

# required fields
# need to do reverse lookup
POP3_BODY_UDF = description
POP3_CATEGORY = QUESTION
POP3_CUST_SEVERITY = 6821
POP3_PRODUCT_NAME = MY_PROD
POP3_STATUS = SUBMIT
#POP3_ASSIGNED_TO = dcrane
POP3_ORIGINATOR = evmailuser
#POP3_SUBMITTED_YES = 8821
HB_SMTP_SERVER=mail.my_company.com
HB_FROM=evmailuser@my_company.com
HB_TO=leeann@my_company.com, donald@my_company.com
```

evmeta

Display list field metadata

SYNOPSIS

evmeta

DESCRIPTION

This command displays all existing list values, for all list fields in the data dictionary, as a colon separate list, with one list item per line. For fields in the Database section of the Data Dictionary, the format of these lines is:

FIXED_DATABASE_NAME:FIXED_NAME:Display title

For fields in the UDF section of the Data Dictionary, the format of these lines is:

FIXED_DATABASE_NAME:Internal ID#:Display title

Only fields to which you have read permission are returned in the data.

This command is used within other CLI scripts. Its use is limited as a standalone utility, except for very specific purposes.

evpasswd

Change a user's password

SYNOPSIS

evpasswd user_id

DESCRIPTION

This command allows the user to change the password of an existing ExtraView user. The user will be prompted for both the old password and the new password. The user_id may be the alternative User ID.

NOTES

Write permission for the SE_SECURITY_USER security key is required to successfully run this command.

EXAMPLE

```
$ evpasswd username
Old password: yyyyy
New password: xxxxx
Re-enter new Password: xxxxx
$
```

evproj

Modify the current users' Business Area and Project

SYNOPSIS

evproj

DESCRIPTION

This command allows the user to change their current (default) Business Area and Project. This command does not have any affect if Business Areas are not enabled, and should not be used in that case.

NOTES

This command may affect the fields available for adding or updating issues, as each area and project may have its own layout defined.

EXAMPLE

```
$ evproj
1. Areaname A
*2. Areaname B
3. Areaname C
4. Areaname D
Number of new Area? 1
1. Project a
2. Project b
*3. Project c
4. Project d
Number of new Project 4
$
```

evreport

Runs an existing ExtraView report or obtains a list of reports to which the current user has access.

SYNOPSIS

evreport [-c|-t] [-n] [-r] [-s skipcount] [-l pagelength]

evreport <Report Name> [-s skipcount] [-l pagelength]

DESCRIPTION

This command allows the user to run an existing Report or to obtain a list of all the reports that the current user can run.

To obtain the list of reports that can be run, the user simply enters the command evreport, without any

parameters. The command will return a list of the reports, sorted by the user's personal reports, then the public reports that can be run. After the list of reports, the user is prompted for the report number that they want to run. Entering the report number executes that report.

The parameters to the command are explained as follows:

-c	The report will be produced in a columnar format. This option has no effect on summary reports. This option cannot be used with the -t option. Column widths default to 15 characters. These defaults can be modified with the existence of a file named report.txt, either in $HOME/.evreport (evreport.txt on Windows) or publicly in evcli/report.txt. This file has the following format: Status = -10 Title = 20.18 Assigned To = 12 EOL = ':' Negative numbers cause left justification. The ".18" suffix in the above example indicates that the field has a maximum length of 18 characters. If it is longer than 18 characters, the field contents will be truncated. Fields with embedded newlines will have their newlines converted to a single blank space. An alternate replacement string may be specified by defining EOL in the report.txt file.
-t	The report will be produced in tabular format. This is similar to columnar format, except that fields do not have a fixed width, and a tab character is used as a separator between fields. This option cannot be used with -c
-n	Suppresses the report header and the record total at the end of the report
-r	Output data in XML format
-l pagelength	Causes the output to paginate pagelength records at a time. The default for pagelength is 10
-s skipcount	Causes the output to skip over 'skipcount-1' issues, and begin the report starting with record number skipcount

EXAMPLES

Run the evreport command and choose a report to run:

```
$ evreport

Public:
1. Assigned to you - Issues Assigned to you - Summary
2. Originated by you - Issues Originated by you - Summary
3. Owned by you - Issues you own - Summary
4. Summary of Issues by Business Area - Open Issues Only - Summary

Report? 2
```

```
Originated by you
Issues Originated by you

Originator System Status Total
================= ====== ====== =====
Gregor McPherson Atlas New 2
Open 1
Main Fixed 1
New 1
Open 1
-----
6
=====
6
$
```

Run an existing report with the name of **All Feature Requests**. Note the use of the quote marks around the report name.

```
$ evreport "All Feature Requests"
---------- 1
ID #: 10215
Severity: Low
Date Created: 1/21/05 2:48 PM
Title: We need to add a new prompt to the screen to ask for the user name
Repeating Rows:
1 record found.
```

evrole

Change the current user's role to a different role

SYNOPSIS

evrole [-r]

DESCRIPTION

The evrole command displays a menu list of all of the roles a user is permitted to use. The user's current role is preceded by an asterisk (*). The user can change their role by entering the number of a new role displayed by the command, at the prompt. Pressing q, or just pressing return exits this command without changing roles.

The -r option is not interactive. It displays the list of roles and their titles, separated by a colon, with the current role preceded by an asterisk.

EXAMPLE

```
$ evrole
```

```
*1. Administrator
2. Guest

Return to continue or number of new role? 2
User 'currentuser' role set to Guest.
$
```

evsearch

Search for issues with a combination of filters and display the search results.

SYNOPSIS

evsearch [-n] [-i] [-c field1,field2, ... fieldn] [-f myfile.txt] [-s skipcount] [-l pagelength] [-p reportID] [-T templatefile] [field1=value1[;value2[;valuen]]] [field2=value1[;value2[;valuen]]] ... ... [q]

evsearch -r [-n] [-s skipcount] [-l pagelength]

[field1=value1[;value2[;valuen]] [field2=value1[;value2[;valuen]]] ... .[q]

evsearch -d

DESCRIPTION

This command allows the user to search for issues that meet specified filter criteria, and displays results (field values) for issues that match the filter criteria. The user must provide both search criteria, and a list of fields to display. Note that any field selected must be on the detailed report layout for the business area and project currently selected, and that the user must have permission to read the fields.

Search Criteria

The user may specify search criteria in two ways, either on the command line, or through an interactive menu.

On the command line, the user may specify fields and fields values to use as search criteria. If the command line is terminated by the letter q, the search will be immediately executed. If the command line does not terminate with the letter q, any search criteria specified on the command line will be passed along to the interactive menu, for additional filter criteria specification.

By default, command-line-specified fields must be identified by their Display Titles. For fields with a display type of List, values must be identified by their display titles. When the -i option is used, fields must be identified by their fixed database names, and list values must be identified by their fixed names (for non-UDF lists), or by their internal list ID's (for UDF lists). The evmeta command displays both of these list names for all list items.

A search may be performed for more than one value of a field by using a semi-colon to separate the values. For example:

STATUS=OPEN;FIXED;CLOSED

will search for issues with a status of Open, Fixed or Closed.

A variety of formats are supported for searching date fields:

datefield=6/24/2003	Searches for a date equal to 6/24/2003
datefield=−6/24/2003	Searches for dates up to and including 6/24/2003
datefield=6/24/2003-6/30/2003	Searches for all dates between 6/24/2003 and 6/30/2003 inclusive

These formats may be used either on the CLI, or through the interactive menu.

Keyword searching is also supported through the CLI. On the command line, use the -k option, and provide one or more keywords as a space-delimited list. When performing a keyword search, all text fields in the issue are searched for occurrences of the keywords entered. When multiple keywords are entered, the search will return issues that match any of the keywords. For example:

evsearch -k ‘core stack’ q

will search for issues containing either the word ‘core’ or the word ‘stack’. Keyword search criteria must be specified on the command line. This option is not available through the interactive menu.
If a command line is not terminated with the letter q, evsearch will display a menu of all the fields available as search criteria, with each field identified by a number. After the menu is displayed, the user will see the prompt, Enter number or q to submit:. When a field number is entered, evsearch will prompt for the search value. If the field is a list, the list of available values for that list will be displayed.

After all desired search criteria have been entered, press q to exit the menu and submit the search query to ExtraView.

List of Fields to Display

There are three ways of specifying the list of fields to be displayed:

1. The -f option
2. The -c option
3. The existence of a file named $HOME/.evsearch (evsearch.txt on Windows). One of these three approaches must be used. If no field list is provided, no field data will be displayed.

By default, only fields on the **Detailed Report** layout of the user's current Business Area and Project, to which the user has read permission, may be requested. You cannot request fields that do not appear on the detailed report for the currently selected business area, project and user role. However, the -p option may be used to specify a different report to be used as the source of field data.

Note: Even though attachments may be displayed on a report layout, evsearch cannot display attachments. To view attachments, users may use the evfiles command.

If the file $HOME/.evsearch exists, and neither the -c option nor the -f option are provided, evsearch will use the contents of this file as the list of field names whose values will be displayed. The format of this file is:

- This file may only contain printable, ASCII characters
- Tabs, spaces, and blank lines are ignored
- Only one field name may be placed per line
- Only fixed database names may be used
- Lines whose first non-blank character is a hash character (#) are ignored
- A line containing only the string, __END__ marks the end of the field list. This line, and all lines after this line in the file are ignored.

OPTIONS

-r	Returns search results in XML format. When this option is specified, the -c, -f, -i, -p, and -T options are ignored
-c "field1, field2, ... , fieldn"	Allows the user to provide a comma-separated list of fields to be displayed, as command line options. Field names must be identified using the fixed database names. If the -f option is used in combination with the -c option, all the fields included in myfile.txt will be displayed in addition to the fields specified with the -c option. When the -c option is used, the file $HOME/.evsearch is ignored. Again, the fields selected must exist on the detailed report for the currently selected business area and project for the user
-f myfile.txt	Allows the user to specify a file that contains a list of all the fields to be displayed. The format of this file is identical to the format of the file $HOME/.evsearch, described above. When the -f option is used, the file $HOME/.evsearch is ignored. As with the -c option, the fields selected must exist on the detailed report for the currently selected business area and project for the user
-k "keyword1 keyword2 ... keywordn"	Allows the user to provide a space-delimited list of keywords to search for. When performing a keyword search, all text fields in the issue are searched for occurrences of the keywords entered. When multiple keywords are entered, the search will return issues that match all of the keywords
-n	Causes the search to ignore case
-i	causes the search to use the ExtraView data dictionary names and list value names, as opposed to the data dictionary titles and the list titles for the search. For example, there is a data dictionary field named STATUS that may have a title of ‘Status of Issue’. This field may have as one of its values a name of NEW that has a title of ‘Newly Submitted’. The following searches are synonymous: evsearch -i status=NEW q evsearch 'Status of Issue=Newly Submitted' q
-l pagelength	causes the output to paginate pagelength records at a time. The default for pagelength is 10
-s skipcount	causes the output to skip over 'skipcount-1' issues, and begin the report starting with record number skipcount
-d	shows a list of available reports by Name (also referred to as Title), with their report ID’s. The report ID’s may be used with the –p option. When this flag is specified, all other flags are ignored
-p reportID	by default, evsearch can only retrieve field data for fields that are on the system-wide Detailed Report. This option allows the user to specify an alternate Report, from which field data may be selected
-T templatefile	allows the user to specify a server side template file to be used for formatting the output of the search.

NOTES

- The output from evsearch may be piped to the evxmlc command, as an alternate method for formatting of the output
- Keywords can be used as a name in the search although the name is not a database field. This is used to provide an unlimited number of keywords in a space-delimited list to ExtraView. When you enter a keyword for the search, all text fields in the issue record are scanned for occurrences of the keyword entered
- If no records are returned by the search, the message "No records found." will be displayed.

EXAMPLE 1

```
$ evsearch status=Open priority="Priority 1" q
--------
ID: 12449
Priority: Priority 1
Title: SM crashes when TMX allocates address
--------
ID: 12446
Priority: Priority 1
Title: My phone does not work
--------
ID: 12426
Priority: Priority 1
Title: Lever arm broken
--------
ID: 12424
Priority: Priority 1
Title: new color scheme inconsistent
4 records found.
$
```

EXAMPLE 2

The output from a search can be piped to the evxmlc command, to give control over the formatting of the output.

```
$ evsearch -r Status=Open q | evxmlc -c "ID(8),ASSIGNED_TO(-
13),SHORT_DESCR(-48)"

Defect # Assigned Title
-------- ------------ ------------------------------------------------
10152 Don Powell User interface does not refresh before menu is
displayed
10132 Don Powell On pressing the enter key on the home page, nothing
happens
10120 Steve Hoydic Architecture Diagram Needs Review
10118 Jom Smith There is a typo in the doc. On page 234
10108 Don Powell I found a problem with calculation on stairs
10105 Dave Green When the timer goes past zero, an error is generated
10034 Don Powell When you click on the Product edit box it throws an
error

7 records processed.
```

```
$
```

evsearchlist

Display fields that may be used as filter criteria by evsearch.

SYNOPSIS

evsearchlist [-r]

DESCRIPTION

This command displays a list of fields that can be used as filter criteria with evsearch. Both the fixed database names, and their matching display titles, are printed.

When the -r option is used, the fixed database names and display titles are each terminated by a colon, with no white space in between the two names. When a field has an allowed value parent, the fixed database name of the parent field is printed after the child field's colon-terminated display title. This parent field is not terminated by a colon.

EXAMPLE 1

```
$ evsearchlist
ASSIGNED_TO Assigned To
CATEGORY Category
CLARIFY_ID Clarify ID
COMMENTS Comments
COMPONENT Component
DATE_CLOSED Date Closed
DATE_CREATED Created
DAYS_IN_STATUS Days in Queue
DAYS_OPEN Days Open
... ... ...

$
```

EXAMPLE 2

```
$ evsearchlist -r
+ASSIGNED_TO:Assigned To
CATEGORY:Category
CLARIFY_ID:Clarify ID
*COMMENTS:Comments
COMPONENT:Component
DATE_CLOSED:Date Closed
DATE_CREATED:Created
DAYS_IN_STATUS:Days in Queue
DAYS_OPEN:Days Open
... ... ...
$
```

evset

Saves the current user's ID and password in a configuration file

SYNOPSIS

evset

DESCRIPTION

This command allows a user to store their ExtraView User ID and password in a configuration file, which is automatically used by most CLI commands for authentication. If this file does not exist, the individual CLI commands will prompt the user for their username and password. Having this configuration file removes the need for a user to type in their ExtraView User ID and password every time they run a CLI command. The User ID may be an alternative User ID.

In Unix environments, this configuration file is saved as $HOME/.evrc. In a Microsoft Windows environment, the file is named evrc.txt, and is saved in the same directory as the CLI commands.

This command will prompt for the User ID and password, and will verify the information before creating the configuration file. Unless a User ID or password changes, it is not necessary to re-run this command. After the configuration file has been created, it may be edited directly.

CLI commands will use any combination of User ID and/or password. If the User ID and/or password are not included in the configuration file, the CLI commands will prompt for the missing information.

NOTES

This configuration file is stored as a plain ASCII text file. Users should be aware of the security implications of having authentication information stored in an ASCII file, and they should make an effort to at least use operating system file permissions to limit access to this file.

EXAMPLE

```
$ evset
User: user_id
Password: xxxxxx
$
```

evsh

Provides a CLI "shell" for accessing CLI commands

SYNOPSIS

evsh [-x]

DESCRIPTION

This command provides a CLI "shell" as a convenient way to run CLI commands. It allows the user to log

into ExtraView once, then enter unlimited CLI commands without the need to re-authenticate for each command. This is most useful in environments where users prefer not to save their usernames and passwords in an ASCII file with the evset command.

The default evsh prompt is ev%. This can be changed by defining a different value for PROMPT in the evconfig.txt file, or, individually, in a user's .evrc file.

Operating system commands may be entered within evsh. A command line history is also made available, using the up and down arrow keys.

evtemplate

Produces an editor template for the evadd and evupdate CLI commands.

SYNOPSIS

evtemplate [-u]

DESCRIPTION

This command produces an editor template for the evadd and evupdate commands. This template may contain comment blocks and the metadata will be neatly enumerated.

The generated file contains field titles, followed by a colon and a space, similar to what is obtained within your editor when you use evadd -e. The difference is that, for metadata fields with multiple values such as PRIORITY and STATUS, the choices are presented in comment blocks. This gives the user a list of all possible valid values that can be entered, eliminating the need to know in advance the values.

The -u flag generates the template for the update screen as opposed to the add screen.

The following example shows the generation of the template file for the add screen, and then invokes the editor with the template.

EXAMPLE

The following example shows the generation of the template file for the add screen then invokes the editor with the template.

```
$ evtemplate > my_input_file.dat
$ evadd -e -f mytemplate.dat
$
```

Here is the raw output from the evtemplate command.

```
$ evtemplate
############################
# Type in value for Title.
############################

Title:
```

```
############################
# Type in value for Product.
#
# Values for Product:
# -----------------
# Globe Power
# Cell Power
# Web Power
############################

Product:

############################
# Type in value for Status.
#
# Values for Status:
# -----------------
# Closed
# Fixed
# Open
# Pending
# Unassigned
#
# Default value provided.
############################

Status: Unassigned

############################
# Type in value for Category.
#
# Values for Category:
# -----------------
# Software
# Documentation
# Hardware
############################

Category:

############################
# Type in value for Priority.
#
# Values for Priority:
# -----------------
# P 1
# P 2
# P 3
############################

Priority:

############################
# Type in value for Module.
#
# Values for Module:
# -----------------
```

```
# Logic
# User Interface
# Database
# Phone
############################

Module:

############################
# Type in value for Assigned.
############################

Assigned:

############################
# Type in value for Description.
############################

Description:

$
```

evupdate

Modify the field values of an existing issue

SYNOPSIS

evupdate [-n]

evupdate [-n][-i] [field1=value1 field2=value2 ... fieldn=valuen] [q]

evupdate [-n] -e [-v] [-f <mytemplate>]

DESCRIPTION

This command allows a user to modify an existing issue in the ExtraView database. All permissions and business rules, including email notification, are implemented the same way through the CLI as they are when issues are modified through the GUI.

This command may be run three different ways.

Interactive Menu

When the -e command line option is not specified, and the user does not terminate a list of fields on the command line with the letter q, evupdate will display a menu of all the fields available for modifying an issue. Each field is identified by a number. After the menu is displayed, the user will see the prompt, "Enter number or q to submit:. When a field number is entered, evupdate will prompt for the value for that field. If the field is a list, the list of available values for that list will be displayed. If the field is a multiple line text field, the user may enter multiple lines of input text. These fields must be terminated by a line containing just a period.

In this menu, an assterisk (*) will identify any field that is required but does not have a value. After a

value has been entered for a required field, the asterisk will be removed from the display.

After all the desired field values have been entered, press q to exit the menu and submit the update to ExtraView.

Command-line only

If all desired fields are included on the command line, and a trailing q ends the command line, the issue will be immediately submitted to ExtraView for update.

When field values are provided on the command line without a trailing q, the user will go into the interactive menu with those field values already populated.

Edit Mode

The -e option invokes an editor to perform the data entry. The editor will use a template file as the basis for collecting the field values for modifying the issue.

In UNIX, the editor defaults to /bin/vi. In Microsoft Windows environments, the editor defaults to c:winnt40notepad.exe. An alternate editor may be specified by either the EVEDITOR environment variable, by an EDITOR line in the evconfig.txt file, or by the EDITOR environment variable.

The template file will be created by one of five methods:

1. The -f <mytemplate> command line option
2. The EVADD environment variable
3. The existence of the file $HOME/.evadd (evadd.txt on Windows)
4. The existence of the file evcli/add.txt
5. The file will automatically be generated from the list of fields returned by evaddlist.

The format of a template file is similar to the following:

```
Title:


Product:


Comments:
.
Attachments:
```

The period following the Comments field in the example above signifies that a multiple line input is allowed. Text for multi-line fields should be inserted immediately after the line containing the multi-line field title (Comments in this example). A period on a line by itself terminates a multi-line field. If a multi-line field is not terminated with a period, evupdate will treat the rest of the template file as part of the multi-line field, when the issue is submitted to ExtraView.

Similarly, when entering multi-line fields either through the interactive menu, or with the -a option, the field must be terminated with a line containing just a period.

When entering text into a multiple line field, the user may read in the contents of an existing file, as if that file had been typed in directly, by using the ~r fname command on a line by itself. This is very useful for entering information into fields with display types of Text Area, Log Area, or Print Text.

When adding an attachment (through any of the methods described above, type in the attachment file name, followed by a space, and then the attachment description, on the lines following the keyword **Attachments**. Only one attachment may be specified per line.

OPTIONS

-n	suppresses the email notification sent whenever an issue is updated
-i	indicates that fields will be identified by their fixed database names instead of by their display titles
-a	runs the command in prompt mode
-e	runs the command in edit mode
-v	adds additional field list value information in the initial template file used with in edit mode
-f templatefile	use templatefile as the initial template file
fieldn=valuen	assigns an initial value to a field. fieldn is the Display Title of a field. valuen must be a valid value for that field
q	ends a list of 'fieldname=value' pairs, the user will not be prompted for additional input.

NOTES

Repeating row data may not be added to an issue through the CLI.

EXAMPLE

```
% evupdate 12256
1. Title 11. Severity
2. Product 12. Assigned To
3. Category 13. OS
4. Component 14. Customer
5. Platforms 15. Clarify ID
6. Description 16. Owner
7. Comments 17. View
8. Workaround 18. Test Case ID
9. Release Notes 19. Test Case Location
10. Priority

Enter number or q to submit: 10
Current value of Priority: P3

1. P0
2. P1
3. P2
4. P3
5. P4
6. P5

Choice for Priority? 2
1. Title 11. Severity
```

2. Product 12. Assigned To
3. Category 13. OS
4. Component 14. Customer
5. Platforms 15. Clarify ID
6. Description 16. Owner
7. Comments 17. View
8. Workaround 18. Test Case ID
9. Release Notes 19. Test Case Location
10. Priority

```
Priority => P1
Enter number or q to submit: q
Bug # 12256 updated.
%
```

evupload

Adds an attachment to an existing issue

SYNOPSIS

evupload [-s userID] [ID] [-a ALT_ID] filename description

DESCRIPTION

This command allows the user to add an attachment to an existing issue in the ExtraView database. The may be an alternative User ID.

The -s option allows the user to specify a different username to be used as the person who created the attachment. This is useful when automated scripts are used to attach files to issues.

You must either have the ID or the -a ALT_ID parameters to the command, but not both.

If you are migrating data from a remote system into ExtraView, a useful strategy to move attachments is to use the -a ALT_ID option to match attachments to issues migrated with the File Import Utility within ExtraView. In this circumstance, the internal unique identifier from the system from which you are migrating data is typically loaded into the ALT_ID field. The -a flag allows you to use the unique identifier from the legacy system without knowing the new ExtraView ID of the migrated issue.

EXAMPLE

This example uploads an attachment with a name of abc.xls into an existing issue with an ID of 12345.

```
$ evupload 12345 abc.xls "This is a spreadsheet"
Uploaded
$
```

This example uploads an attachment with a name of abc.xls into an existing issue with an ALT_ID of 65432.

```
$ evupload -a 65432 abc.xls "This is a spreadsheet"
Uploaded
$
```

evuserfields

Displays the list of fields used when adding a new user account.

SYNOPSIS

evuserfields [-r]

DESCRIPTION

This command displays the list of fields that are used when adding a new user account to the ExtraView database. The field names are displayed in two columns. The first column is the field's fixed database name. The second column is the field's display title.

When the -r option is used, the fixed database names and display titles separated by a colon, with no white space in between the two names. Fields that are required on the add user screen are preceded by an asterisk (*).

EXAMPLE

```
$ evuserfields
SECURITY_USER_ID Security User Id
SECURITY_PASSWORD Security Password
FIRST_NAME First Name
LAST_NAME Last Name
JOB_TITLE Job Title
COMPANY_NAME Company
ADDRESS_LINE1 Address Line1
ADDRESS_LINE2 Address Line2
CITY City
STATE State
POSTAL_CODE Postal Code
COUNTRY Country
EMAIL E-Mail Address
WORK_TELEPHONE Work Telephone
HOME_TELEPHONE Home Telephone
CELL_PHONE Cell Phone
FAX Fax
PAGER Pager
$
```

evusergroups

Displays the list of defined user roles.

SYNOPSIS

evusergroups [-r]

DESCRIPTION

This command displays the list of defined user roles in the ExtraView database. The names are displayed in two columns. The first column is the role's fixed database name. The second column is the role's display title.

When the -r option is used, the fixed database names and display titles separated by a colon, with no white space in between the two names.

EXAMPLE

```
$ evusergroups -r
ADMIN:Administrator
CUSTOMER:Customer
HW_ENG:Hardware Engineering
QA:Quality Aassurance
SW_ENG:Software Engineering
SUPPORT:Support
$
```

evusers

Outputs a list of ExtraView users

SYNOPSIS

evusers [-r]

evusers -s

evusers [-r] [-l] [-m 1|2] [-s filters]

DESCRIPTION

This command outputs a list of ExtraView users to the console. The list appears in the following format:

user id, first name, last name

The -r option provides the list in a raw format, using the system delimiter between the values in the following way:

user_id:first_name:last_name

The -s option restricts the output of records that match the wildcard pattern supplied in <format>. For example:

evusers -s "id=%OB%"

will output only names which contain the character pattern OB, such as BOB and ROBERT

The -m 1|2 option will provide role information about the users.

- If -m 1 specified, the command returns the user's current role –

  USERID : First Name : Last Name : Current Role

- If -m 2 is specified, the command returns a delimited list of all roles –

  USERID : First Name : Last Name : Current Role : Second Role : Third Role : .....

- The -l option displays the alternate User ID

EXAMPLE

```
$ evusers
User ID First Name Last Name
------- ---------- ---------
SYSTEM Super User
ADMIN System Administrator
GUEST Guest User
DEV Jim Developer
TEST George Test
QA Alison Tester
IT Internal IS Support
CSR Customer Support Rep
EVP4D ExtraView Perforce Daemon
DHOYDIC David Hoydic
SHOYDIC steven hoydic
GRANT Grant Gongaware
$
```

evversion

Displays the CLI version

SYNOPSIS

evversion

DESCRIPTION

This command displays the CLI version information.

EXAMPLE

```
$ evversion
Release 4.2.3.1 Build 70
$
```

evxmlc

Formats the XML output of a CLI command to become readable text

SYNOPSIS

evxmlc [-i] [-t] [-f filename] [-s] [-c field1, field2, ... fieldn]

DESCRIPTION

This command formats the XML output of a CLI command into either a tab-delimited report, or a columnar report. If a specific list of fields is not provided with the -c option, all field data sent to evxmlc will be output.

OPTIONS

-i	outputs fixed database names, instead of display titles
-t	causes fields names and data to be separated by a tab character, instead of spaces
-f filename	specifies the name of the file containing the XML to be formatted
-s	removes non-printable characters and character sequences from the output
-c field1[([-]nn)], field2[([-]nn)], ... fieldn[([-]nn)]	specifies the list of fields to be output. Field names must be identified using fixed database names. Field widths may be specified by including a numeric value enclosed in parenthesis immediately after the field name. Additionally, if the field width is preceded by a minus sign, the data will be left justified. Without a minus sign, the data will be right justified.