

Netcool/OMNIbus
Version 7 Release 3

*Web GUI Administration and User's
Guide*



Netcool/OMNIbus
Version 7 Release 3

*Web GUI Administration and User's
Guide*



Note

Before using this information and the product it supports, read the information in "Notices" on page 421.

This edition applies to version 7, release 3, modification 1 of IBM Tivoli Netcool/OMNIBus (product number 5724-S44) and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 2001, 2011.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this publication vii

Intended audience	vii
What this publication contains	vii
Publications	viii
Accessibility	x
Tivoli technical training.	x
Support information.	x
Conventions used in this publication	x

Chapter 1. Administering the Web GUI . . . 1

Administering the Web GUI server	1
Stopping and starting the application server	1
Restarting the server.	1
Viewing the application server profile	2
Changing the timeout setting	3
Adding ObjectServers as user registries	5
Changing the password for the connection to the ObjectServer	6
Administering timed tasks.	7
Querying the event database	8
Copying data between Web GUI servers.	10
Backing up and restoring data	21
Administering the GUI framework.	24
Tivoli Integrated Portal layout	24
How to customize a Tivoli Integrated Portal setup	25
How to customize a Tivoli Integrated Portal setup	27
Administering pages	28
Administering views	36
Administering portlets.	40
Administering console preference profiles	45
Exporting and importing Tivoli Integrated Portal data	47
Resource types	63
Manage Global Refresh	63
Administering users, roles, and groups	65
Web GUI users, roles, and groups	65
Changing passwords	69
Creating the Web GUI administrative user	69
Modifying the preferences of a Web GUI user	70
Administering users.	73
Administering roles	80
Administering groups	84
Assigning roles to users and groups	92
Administering a load balancing cluster	92
The Web GUI in a load balancing environment	93
Maintaining the list of files to monitor	97
Cluster administration tools	97
Load balancing best practices	98
Troubleshooting	100
Remotely administering the Web GUI server	100
The Web GUI Administration API (WAAPI)	101
Using WAAPI	104
WAAPI requests	106

Installing the WAAPI client on a remote host	193
WAAPI security	194
Troubleshooting	202
Checking system information	202
Viewing TIPProfile logs for login errors	203
Editing a properties file	203
Setting the log level	203
Generating performance logs	205
Setting a trace	205
Auditing the usage of objects	206
Web GUI log files	207
Harmless authentication messages	208
No user role assigned	208
Slow network response	208
AEL fails to start and displays message W0025	209
Event Dashboard unable to get data from server	209
Maps are slow to display severity colors	210
Changing the height setting of a Map portlet has no effect.	210
The connection to the ObjectServer is lost or the AEL issues a timeout.	211
Performance tuning tips for the Web GUI	211
Increasing JVM memory on the Web GUI server	211
Increasing memory on Web GUI client Java Virtual Machines	212
Performance tuning	212
CGI support.	214

Chapter 2. Administering a load balancing cluster. 215

The Web GUI in a load balancing environment	215
Structure of a cluster	216
Configuration data	216
Updating configuration data	217
Conditions necessary for changing configuration data	218
Administering a load balancing cluster	218
Using a load-balanced cluster	219
Maintaining the list of files to monitor	219
Cluster administration tools	220
Monitoring a load balancing cluster	220
Load balancing best practices	221
Troubleshooting	222
Resynchronizing a node with the cluster	222
Recovering from a database corruption.	222

Chapter 3. Setting portlet preferences 225

Setting AEL portlet preferences	225
Setting Chart portlet preferences	227
Setting Event Dashboard portlet preferences and defaults	228
Setting gauge portlet preferences	231
Changing the General settings for all gauges	232
Selecting the data sources and ObjectServers	232
Generating HTML for mobile devices	232

Changing the properties of gauges	233
Adding gauges	234
Removing gauges	234
Rearranging gauges	234
Changing the bi-directional text settings	235
Setting Inline Frame portlet preferences	235
Setting LEL portlet preferences	236
Setting preferences for the Map portlet	237
Setting Table View portlet preferences	238
Setting Web widget portlet preferences	239

Chapter 4. Customizing Active Event Lists 241

Setting the appearance and behavior of the Active Event List	241
Changing the AEL refresh rate.	241
Turning data row caching on or off	242
Changing event severity icons.	242
Restricting the number of rows displayed in the Active Event List	243
Enabling notifications and setting notification criteria	244
Enabling flashing and setting flash speed and brightness	245
Modifying the AEL font and window settings	245
Modifying the AEL date and time format	246
Changing how event severity is depicted in the AEL	248
Changing which areas of the AEL are displayed	249
Changing the font color for an event after it has been acknowledged	250
Adding sounds to use for notifications	250
Creating event management tools	250
Event management tools overview	251
Prompt types	253
Setting up CGI and URL tools.	255
Creating SQL tools	261
Creating command-line tools	263
Creating script tools	265
Modifying tools	267
Renaming tools.	270
Copying tools	271
Deleting tools	274
Tool runtime parameters	274
Customizing AEL Menus	276
The Alerts and Tools menus	276
Adding tools to a menu	277
Modifying existing tools.	277
Creating submenus	278
Adding submenus to a menu	279
Deleting submenus	279
Removing items from a menu	280
Changing the order of items in a menu.	280
Renaming menu items	280

Chapter 5. Filtering event information 283

Filters	283
Views	285
Setting up filters for event data	286
Filter Builder overview	286

Creating and editing filters	287
Copying a filter from another filter group	296
Deleting filters	297
Setting up views for event lists	298
Creating views	298
Adding and removing columns	299
Configuring column titles and width	300
Changing the column order	300
Locking columns	301
Changing the sorting precedence	302
Changing the sort order in a column	303
Copying global views	303

Chapter 6. Visualizing high-level event information 305

Visualizing event information on Event Dashboards	305
Event Dashboard overview.	305
Adding monitor boxes to Event Dashboard portlets	306
Deleting monitor boxes from Event Dashboards	307
Customizing monitor box information	308
Changing the event information displayed on monitor boxes	308
Customizing the monitor boxes on Event Dashboards	309
Freezing and unfreezing Event Dashboards	310
Customizing Active Event List actions on Event Dashboards	311
Creating a page with an Event Dashboard and Active Event List	312
Visualizing event information on maps.	313
About maps.	313
Previewing maps	315
Creating maps	315
Customizing maps	317
Publishing maps	334
Previewing map resources	337
Deleting map resources	337
Visualizing high-level event information on charts	337
Tivoli Netcool/OMNIbus Web GUI charts.	337
Creating a chart definition file.	343
Publishing charts	344
Chart configuration reference	345
Visualizing event information on gauges	361
Gauges and metrics	361
Using the gauge page	364
Publishing Gauges pages to mobile devices	365
Creating and managing metrics for gauges	365
Setting the thresholds for a gauge	368

Chapter 7. Monitoring events in the Web GUI 371

Monitoring events in the AEL	371
Event management in Active Event Lists	371
Accessing Active Event Lists	372
Acknowledging and deacknowledging events	372
Assigning ownership to events	373
Changing the event information displayed	373
Copying data from the event list	374

Deleting events	374
Displaying event information in full.	374
Freezing the event data	375
Maintaining a journal for an event	376
Prioritizing events.	377
Refreshing the event data	377
Running predictive eventing tools in the Web GUI	378
Monitoring TADDM events in the Web GUI	379
Searching for event list data	380
Selecting and deselecting events	381
Sorting columns	382
Using tools to manage events	383
Monitoring events in the LEL	383
Viewing events in the Lightweight Event List	384
Displaying full event information in the Lightweight Event List	384
Monitoring events in the Table View	385

Appendix A. Accessibility features for the Web GUI 387

Appendix B. Data source configuration file data reference . . . 389

XML overview	389
DTD reference	389
Element reference	392
Attribute reference	395

Appendix C. Invalid characters in filters, views, and tools 399

Appendix D. SmartPage commands and templates 401

SmartPage commands overview	401
SmartPage code example	401
SmartPage command reference	402
enable:Smartpage command	402
Validate command.	402
insert:AEL command	402
insert:AELMAPVIEW command	405
insert:MAPLET command	406
insert:FILTERPAGE command	407
insert:USERMAPLIST command	408
insert:TABLEVIEW command	408
insert:USERNAME command	409
insert:USERSHOMEURL command	409
Template overview	409
Template example	410
Guidelines for using SmartPage templates.	411
Creating SmartPage templates	411

Appendix E. Web GUI database tables 413

OMNIBUS_WEB_GUI.CONFIG_ITEMS table	413
OMNIBUS_WEB_GUI.NODES.	413
OMNIBUS_WEB_GUI.NODES_CONFIG_ITEMS	414

Appendix F. URLs for opening Web GUI pages. 415

Notices 421

Trademarks	423
----------------------	-----

Index 425

About this publication

The IBM Tivoli Netcool/OMNIBus Web GUI is a Web-based application that processes network events from one or more data sources and presents the event data to users in various graphical formats.

The *IBM Tivoli Netcool/OMNIBus Web GUI Administration and User's Guide* describes how to administer, customize, and use the Tivoli Netcool/OMNIBus Web GUI.

For information on how to use the Web GUI Administration API (WAAPAPI) to administer the Web GUI remotely, refer to the *IBM Tivoli Netcool/OMNIBus Web GUI Administration API (WAAPAPI) User's Guide*.

Intended audience

This publication is intended for administrators and operators who use the Tivoli Netcool/OMNIBus Web GUI. This publication provides information on how to administer the Web GUI, how to create customized event displays, and how to monitor and manage event data.

What this publication contains

This publication contains the following sections:

- Chapter 1, “Administering the Web GUI,” on page 1
Describes the administration tasks that need to be performed as part of system maintenance and to ensure the correct operation of the Web GUI.
- “Administering a load balancing cluster” on page 92
Describes the administration tasks specific to a load balancing cluster.
- Chapter 3, “Setting portlet preferences,” on page 225
Describes how to customize the appearance and behavior of the Web GUI portlets.
- Chapter 4, “Customizing Active Event Lists,” on page 241
Describes how to change the appearance and behavior of the Active Event List (AEL), how to create tools that operators can run against events in the AEL, and how to add tools to AEL menus.
- Chapter 5, “Filtering event information,” on page 283
Describes how to use filters to apply SQL conditions to ObjectServer data, and how to use views to control which columns are displayed in the AEL, and the appearance of the columns.
- Chapter 6, “Visualizing high-level event information,” on page 305
Describes how to use the Event Dashboard portlet, maps, and charts to obtain an overview of the events on your network.
- Chapter 7, “Monitoring events in the Web GUI,” on page 371
Describes how to use the event management functions of Active Event List, Lightweight Event List and the Table View.
- Appendix A, “Accessibility features for the Web GUI,” on page 387
Lists the accessibility features of the Web GUI that help users with a disability, restricted mobility, or limited vision.

- Appendix B, “Data source configuration file data reference,” on page 389
Describes the structure of the `ncwDataSourceDefinitions.xml` data source definitions file.
- Appendix C, “Invalid characters in filters, views, and tools,” on page 399
Describes the characters that cannot be used in the name of filters, views, and tools.
- Appendix D, “SmartPage commands and templates,” on page 401
Describes how to use SmartPage commands to populate Web pages that are served by the Web GUI.
- Appendix E, “Web GUI database tables,” on page 413
Defines the structure of the Web GUI tables in the database of a load balancing cluster.
- Appendix F, “URLs for opening Web GUI pages,” on page 415
Describes how to build URLs that launch the Web GUI.

Publications

This section lists publications in the Tivoli Netcool/OMNIBus library and related documents. The section also describes how to access Tivoli publications online and how to order Tivoli publications.

Your Tivoli Netcool/OMNIBus library

The following documents are available in the Tivoli Netcool/OMNIBus library:

- *IBM Tivoli Netcool/OMNIBus Installation and Deployment Guide*, SC14-7604
Includes installation and upgrade procedures for Tivoli Netcool/OMNIBus, and describes how to configure security and component communications. The publication also includes examples of Tivoli Netcool/OMNIBus architectures and describes how to implement them.
- *IBM Tivoli Netcool/OMNIBus Administration Guide*, SC14-7605
Describes how to perform administrative tasks using the Tivoli Netcool/OMNIBus Administrator GUI, command-line tools, and process control. The publication also contains descriptions and examples of ObjectServer SQL syntax and automations.
- *IBM Tivoli Netcool/OMNIBus Web GUI Administration and User's Guide*, SC14-7606
Describes how to perform administrative and event visualization tasks using the Tivoli Netcool/OMNIBus Web GUI.
- *IBM Tivoli Netcool/OMNIBus User's Guide*, SC14-7607
Provides an overview of the desktop tools and describes the operator tasks related to event management using these tools.
- *IBM Tivoli Netcool/OMNIBus Probe and Gateway Guide*, SC14-7608
Contains introductory and reference information about probes and gateways, including probe rules file syntax and gateway commands.
- *IBM Tivoli Monitoring for Tivoli Netcool/OMNIBus Agent User's Guide*, SC14-7610
Describes how to install the health monitoring agent for Tivoli Netcool/OMNIBus and contains reference information about the agent.
- *IBM Tivoli Netcool/OMNIBus Event Integration Facility Reference*, SC14-7611
Describes how to develop event adapters that are tailored to your network environment and the specific needs of your enterprise. This publication also describes how to filter events at the source.

- *IBM Tivoli Netcool/OMNIBus Error Messages Guide*, SC14-7612
Describes system messages in Tivoli Netcool/OMNIBus and how to respond to those messages.
- *IBM Tivoli Netcool/OMNIBus Web GUI Administration API (WAAPI) User's Guide*, SC22-5403-00
Shows how to administer the Tivoli Netcool/OMNIBus Web GUI using the XML application programming interface named WAAPI.

Accessing terminology online

The *Tivoli Software Glossary* includes definitions for many of the technical terms related to Tivoli software. The *Tivoli Software Glossary* is available at the following Tivoli software library Web site:

<http://publib.boulder.ibm.com/tividd/glossary/tivliglossarymst.htm>

The IBM Terminology Web site consolidates the terminology from IBM product libraries in one convenient location. You can access the Terminology Web site at the following Web address:

<http://www.ibm.com/software/globalization/terminology>

Accessing publications online

IBM posts publications for this and all other Tivoli products, as they become available and whenever they are updated, to the Tivoli Information Center Web site at:

<http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp>

Note: If you print PDF documents on other than letter-sized paper, set the option in the **File > Print** window that allows Adobe Reader to print letter-sized pages on your local paper.

Ordering publications

You can order many Tivoli publications online at the following Web site:

<http://www.elink.ibm.link.ibm.com/publications/servlet/pbi.wss>

You can also order by telephone by calling one of these numbers:

- In the United States: 800-879-2755
- In Canada: 800-426-4968

In other countries, contact your software account representative to order Tivoli publications. To locate the telephone number of your local representative, perform the following steps:

1. Go to the following Web site:
<http://www.elink.ibm.link.ibm.com/publications/servlet/pbi.wss>
2. Select your country from the list and click **Go**. The Welcome to the IBM Publications Center page is displayed for your country.
3. On the left side of the page, click **About this site** to see an information page that includes the telephone number of your local representative.

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully.

With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate most features of the graphical user interface.

For additional information, see the Accessibility Appendix in Appendix A, “Accessibility features for the Web GUI,” on page 387.

Tivoli technical training

For Tivoli technical training information, refer to the following IBM Tivoli Education Web site:

<http://www.ibm.com/software/tivoli/education>

Support information

If you have a problem with your IBM software, you want to resolve it quickly. IBM provides the following ways for you to obtain the support you need:

Online

Go to the IBM Software Support site at <http://www.ibm.com/software/support/probsub.html> and follow the instructions.

IBM Support Assistant

The IBM Support Assistant (ISA) is a free local software serviceability workbench that helps you resolve questions and problems with IBM software products. The ISA provides quick access to support-related information and serviceability tools for problem determination. To install the ISA software, go to <http://www.ibm.com/software/support/isa>

Conventions used in this publication

This publication uses several conventions for special terms and actions and operating system-dependent commands and paths.

Typeface conventions

This publication uses the following typeface conventions:

Bold

- Lowercase commands and mixed case commands that are otherwise difficult to distinguish from surrounding text
- Interface controls (check boxes, push buttons, radio buttons, spin buttons, fields, folders, icons, list boxes, items inside list boxes, multicolumn lists, containers, menu choices, menu names, tabs, property sheets), labels (such as **Tip:** and **Operating system considerations:**)
- Keywords and parameters in text

Italic

- Citations (examples: titles of publications, diskettes, and CDs)

- Words defined in text (example: a nonswitched line is called a *point-to-point* line)
- Emphasis of words and letters (words as words example: "Use the word *that* to introduce a restrictive clause."; letters as letters example: "The LUN address must start with the letter *L*.")
- New terms in text (except in a definition list): a *view* is a frame in a workspace that contains data
- Variables and values you must provide: ... where *myname* represents....

Monospace

- Examples and code examples
- File names, programming keywords, and other elements that are difficult to distinguish from surrounding text
- Message text and prompts addressed to the user
- Text that the user must type
- Values for arguments or command options

Operating system-dependent variables and paths

This publication uses the UNIX convention for specifying environment variables and for directory notation.

When using the Windows command line, replace *\$variable* with *%variable%* for environment variables, and replace each forward slash (/) with a backslash (\) in directory paths. For example, on UNIX systems, the *\$NCHOME* environment variable specifies the path of the Netcool® home directory. On Windows systems, the *%NCHOME%* environment variable specifies the path of the Netcool home directory. The names of environment variables are not always the same in the Windows and UNIX environments. For example, *%TEMP%* in Windows environments is equivalent to *\$TMPDIR* in UNIX environments.

If you are using the bash shell on a Windows system, you can use the UNIX conventions.

Home directories for the Web GUI and Tivoli Integrated Portal

The Web GUI and the Tivoli Integrated Portal use separate directory structures within the main installation directory. References to those directories use the following conventions:

install_dir

Refers to the directory where the Web GUI and the Tivoli Integrated Portal are installed.

Examples:

/opt/IBM/tivoli on UNIX environments.

C:\IBM\tivoli on Windows systems.

webgui_home_dir

Refers to the directory where the Web GUI is installed. This directory is known as the Web GUI home directory.

Examples:

/opt/IBM/tivoli/netcool/omnibus_webgui on UNIX environments.

C:\IBM\tivoli\netcool\omnibus_webgui on Windows systems.

tip_home_dir

Refers to the directory where the Tivoli Integrated Portal is installed. This directory is known as the Tivoli Integrated Portal home directory.

Examples:

/opt/IBM/tivoli/tipv2 on UNIX environments.

C:\IBM\tivoli\tipv2 on Windows systems.

Chapter 1. Administering the Web GUI

You need to perform administration tasks as part of system maintenance and to ensure the correct operation of the Web GUI.

Administering the Web GUI server

Perform these tasks to administer the Web GUI server.

Stopping and starting the application server

The Tivoli Integrated Portal Server starts automatically after it has been installed and whenever the computer is started. You can manually stop the server before beginning certain configuration tasks or as needed.

1. In the *tip_home_dir*/profiles/TIPProfile/bin directory, depending on your operating system, enter one of the following commands:

- **Windows** stopServer.bat server1
- **UNIX** **Linux** stopServer.sh server1

Note: On UNIX and Linux systems, you are prompted to provide an administrator username and password.

2. In the *tip_home_dir*/profiles/TIPProfile/bin directory, depending on your operating system, enter one of the following commands:

- **Windows** startServer.bat server1
- **UNIX** **Linux** startServer.sh server1

Related tasks

“Setting a trace” on page 205

Restarting the server

After customization and configuration activities you might need to restart the Web GUI server.

Restart the server after or while carrying out any of the following actions on your Web GUI server:

- Modifications to any of the following files:
 - server.init
 - ncwDataSourceDefinitions.xml
 - virtualhosts.xml
 - deployment.xml
 - security.xml
 - winconfig.xml
 - Any properties file in the *tip_home_dir*/tip/properties directory
- Setting up a load balancing cluster
- Adding a node to a load balancing cluster
- Adding or changing user registries
- Backing up and restoring the Web GUI

- Copying configurations from another Web GUI server
- Configuring encryption
- Configuring single sign-on
- Configuring LDAP or Active Directory and their connections

If you do not use the timed tasks facility in the `server.init` file, you also need to restart the server after changing any files in the following directories in `webgui_home_dir/etc`:

- `configstore`
- `cgi-bin`
- `charts`
- `charts/definitions`
- `templates` and all the directories it holds

To restart the server:

1. On the command-line interface, change to the `tip_home_dir/profiles/TIPProfile/bin`.
2. Stop the server:
 - **Linux** **UNIX** `stopServer.sh server1`
 - Attention:** Linux and Unix systems prompt you to supply the user name and password of the administrative user.
 - **Windows** `stopServer.bat server1`

Wait a moment for the server to completely shut down.
3. Start the server:
 - **Linux** **UNIX** `startServer.sh server1`
 - **Windows** `startServer.bat server1`

Related tasks

“Administering timed tasks” on page 7

Viewing the application server profile

Open the application server profile to review the port number assignments and other information.

The profile of the application server is available as a text file on the computer where it is installed.

1. Locate the `tip_home_dir/profiles/TIPProfile/logs` directory.
2. Open `AboutThisProfile.txt` in a text editor.

Example

This is the profile for an installation on in a Windows environment as it appears in `tip_home_dir\profiles\TIPProfile\logs>AboutThisProfile.txt`:

```
Application server environment to create: Application server
Location: C:\IBM\tivoli\tip\profiles\TIPProfile
Disk space required: 200 MB
Profile name: TIPProfile
Make this profile the default: True
Node name: TIPNode Host name: tivoliadmin.usca.ibm.com
Enable administrative security (recommended): True
Administrative consoleport: 16315
```


Administrative console secure port: 16316
HTTP transport port: 16310
HTTPS transport port: 16311
Bootstrap port: 16312
SOAP connector port: 16313
Run application server as a service: False
Create a Web server definition: False

What to do next

If you want to see the complete list of defined ports on the application server, you can open *tip_home_dir*/properties/TIPPortDef.properties in a text editor:

```
#Create the required WAS port properties for TIP
#Mon Oct 06 09:26:30 PDT 2008
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS=16323
WC_adminhost=16315
DCS_UNICAST_ADDRESS=16318
BOOTSTRAP_ADDRESS=16312
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS=16321
SOAP_CONNECTOR_ADDRESS=16313
ORB_LISTENER_ADDRESS=16320
WC_defaulthost_secure=16311
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS=16322
WC_defaulthost=16310
WC_adminhost_secure=16316
```

Related tasks

“Viewing TIPProfile logs for login errors” on page 203

Port assignments

The application server requires a set of sequentially numbered ports.

The sequence of ports is supplied during installation in the response file. The installer checks that the number of required ports (starting with the initial port value) are available before assigning them. If one of the ports in the sequence is already in use, the installer automatically terminates the installation process and you must specify a different range of ports in the response file.

Changing the timeout setting

You can change the period of time after which users are automatically logged out of the Web GUI.

The default timeout period is 30 minutes. After you have changed the timeout setting, you must restart the server. In a load balancing cluster, carry out this task on all nodes.

To change the timeout setting:

1. Open the following file: *tip_home_dir*/profiles/TIPProfile/config/cells/TIPCell/applications/isc.ear/deployments/isc/deployment.xml
2. In this file, locate the following section:

```
<tuningParams xmi:id="TuningParams_1226438889945"
usingMultiRowSchema="false"
maxInMemorySessionCount="1000" allowOverflow="true" schedule
Invalidation="false" writeFrequency="TIME_BASED_WRITE" writeInterval="10"
writeContents="ONLY_UPDATED_ATTRIBUTES" invalidationTimeout="30">
<invalidationSchedule xmi:id="InvalidationSchedule_1226438889946"
firstHour="14" secondHour="2"/>
```

3. Change the value of the **invalidationTimeout** attribute to the required timeout period, in minutes. For example, to change the period to 15 minutes, set the value as follows:

```
invalidationTimeout="15"
```

4. Save and close the file.
5. Open the following file: *tip_home_dir/profiles/TIPProfile/config/cells/TIPCell/security.xml*.

6. In this file locate the following section:

```
<authMechanisms xmi:type="security:LTPA" xmi:id="LTPA_1"
OID="oid:1.3.18.0.2.30.2"
authContextImplClass="com.ibm.ISecurityLocalObjectTokenBaseImpl.
WSSecurityContextLTPAImpl" authConfig="system.LTPA"
simpleAuthConfig="system.LTPA"
authValidationConfig="system.LTPA" timeout="1440"
keySetGroup="KeySetGroup_TIPNode_1">
```

7. Change the value of the **timeout** attribute to the required timeout period, in minutes.

8. Save and close the file.

9. Open the following file: *tip_home_dir/profiles/TIPProfile/config/cells/TIPCell/nodes/TIPNode/servers/server1/server.xml*.

10. In this file, locate the following section:

```
<tuningParams xmi:id="TuningParams_1183077764084"
usingMultiRowSchema="false"
maxInMemorySessionCount="1000" allowOverflow="true"
scheduleInvalidation="false"
writeFrequency="TIME_BASED_WRITE" writeInterval="10"
writeContents="ONLY_UPDATED_ATTRIBUTES"
invalidationTimeout="30">
```

11. Change the value of the **invalidationTimeout** attribute to the required timeout period, in minutes.

12. Save and close the file.

13. Restart the server.

Related concepts

"The Web GUI in a load balancing environment" on page 93

Related tasks

"Restarting the server" on page 1

Switching off the session timeout

You can set the session timeout to a large value to prevent sessions from timing out. This is useful where you have facilities such as wall displays that need to be available all the time.

You cannot switch off the timeout setting completely. Instead, you can set it to a large value that effectively means it is switched off. However, take note of the following:

- There are security implications if unattended sessions do not timeout. Consider these implications carefully before implementing this procedure.
- There may be a build up of resource usage should users close their browsers without logging off beforehand. Restarting the server reclaims those resources.

In a load balancing cluster, carry out this task on all nodes.

To set the timeout so that sessions do not time out:

Use the procedure in [<xref href="#web_adm_settimeout"></xref>](#) to set the timeout values to the following values:

Table 1. Attribute values to switch off the session timeout

File	Attribute	Value
deployment.xml	invalidationTimeout	-1
security.xml	timeout	2147483647
server.xml	invalidationTimeout	-1

Related concepts

“The Web GUI in a load balancing environment” on page 93

Related tasks

“Restarting the server” on page 1

Adding ObjectServers as user registries

You can configure additional ObjectServers to be user registries for the Web GUI.

Note: Setting an ObjectServer as a user registry is different from setting an ObjectServer as a data source for the event feed into the Web GUI. To set an ObjectServer as a data source, you must edit the data source configuration file.

After you have set an additional ObjectServer user registry, you must restart the server.

To set additional ObjectServers as Web GUI user registries:

1. Change to the *tip_home_dir/bin* directory.
2. Run the following command for your operating system:

```
UNIX      Linux      confvmm4ncos.sh user password host port [failoverhost failoverport]
```

```
Windows   confvmm4ncos.bat user password host port [failoverhost failoverport]
```

Where the parameters are as follows:

user

A user with administrative rights on the required ObjectServer.

password

The password of the user specified in *user*.

host

The IP address of the ObjectServer host.

port

The port number that is used by the ObjectServer.

failoverhost

Optional: If a failover ObjectServer is required, the IP address of the failover ObjectServer host.

failoverport

Optional: If a failover ObjectServer is required, the port number that is used by the failover ObjectServer host.

3. Restart the server.

What to do next

Now add the ObjectServer to the federated repository.

For more information about adding an ObjectServer user registry to the federated repository, see the *IBM Tivoli Netcool/OMNIBus Installation and Deployment Guide*.

Related tasks

“Restarting the server” on page 1

Changing the password for the connection to the ObjectServer

If the password of the user specified for the connection between the ObjectServer and the Web GUI is changed, the new password must be set on the Web GUI server.

The password of the ObjectServer user must be maintained in the Web GUI for the event data feed from the ObjectServer, and if the ObjectServer is configured as a user registry through the Virtual Member Manager (VMM) adapter.

The ObjectServer connection details for VMM are stored in the following file: *tip_home_dir/profiles/TIPProfile/config/cells/TIPCell/wim/config/wimconfig.xml*. The connection details for the ObjectServer event data feed are stored in the following file: *ncwDataSourceDefinitions.xml*

After you have updated the password on the Web GUI server, you must restart the server.

To change the password:

1. To change the ObjectServer password for VMM, enter the following command:

```
tip_home_dir/bin/confvmm4ncos.sh user newpassword host port  
configfile=../profiles/TIPProfile/config/cells/TIPCell/wim/config/  
wimconfig.xml
```

Where the parameters are as follows:

user

The ObjectServer user that is used for the connection.

newpassword

The new password for the ObjectServer user.

host

The fully-qualified host name on which the ObjectServer is installed.

port

The port on which the ObjectServer is installed.

2. To change the ObjectServer password for the event data feed:
 - a. Edit the *ncwDataSourceDefinitions.xml* file.
 - b. In the `<ncwDataSourceCredentials>` element, change the value of the **password** attribute to the new password.
 - c. Save and close the file.
3. Restart the server.

Related tasks

“Restarting the server” on page 1

Administering timed tasks

Timed tasks are the key part to a Web GUI server automatically loading changes in configuration data, without the need to restart the server. You manage timed tasks through properties in the Web GUI initialization file (`server.init`).

To administer the timed tasks facility:

1. Open `webgui_home_dir/etc/server.init` in a text editor.
2. Carry out any combination of the following tasks, as required:
 - Control the use of timed tasks
 - Set the refresh properties
3. Save the file.
4. If you enabled timed tasks in step 2, restart the Tivoli Integrated Portal server.

Related concepts

“Overview of timed tasks” on page 8

Related tasks

“Restarting the server” on page 1

Related reference

“Load balancing best practices” on page 98

Controlling the use of timed tasks

1. Locate the `timedtasks.enabled` property.
2. Set the value of the property as required:

Value	Meaning
false	Switches off the timed tasks facility.
true	Enables the timed tasks facility. This is the setting required for automatic processing of changes to configuration data.

Setting the refresh properties

In most cases, the supplied values are adequate, however you can use this procedure to change them if necessary.

1. Decide on values (in seconds) for:
 - The start delay
 - The run period
2. Locate the `timedtasks.default.startdelay` property and change its value to the start delay you require.
3. Locate the `timedtasks.default.runperiod` property and change its value to the run period you require.

Overview of timed tasks

Timed tasks simplify the administration of a Web GUI server or of a cluster of Web GUI servers.

Timed tasks enable a Web GUI server to detect and load changes in the configuration data. The tasks implement changes in the configuration without having to restart the server. This is especially important in a load-balancing cluster where maintenance of service to users must continue uninterrupted.

What are timed tasks?

Timed tasks are a set of batch processes that periodically look for changes in the configuration files. If any of those files has changed since the previous run, the processes load the new configuration data into the Web GUI server.

Characteristics of a timed task

A timed task has the following characteristics:

- A start delay
The start delay determines how long (in seconds) after the server starts that the task first looks for changes in the associated configuration data.
- A run period
The run period determines the length of time (in seconds) between each subsequent run of the task.

The start delay and run period together make up the timed task's schedule. In a load-balanced cluster, you are recommended to use the same schedule on all cluster members.

The definition of timed tasks

The Web GUI's configuration file holds the definition of timed tasks. This path of this file is *webgui_home_dir/etc/server.init*.

The definition of the timed tasks looks like this:

```
timedtasks.default.startdelay: 120  
timedtasks.default.runperiod: 120
```

This defines a start delay of two minutes and a run period, also of two minutes.

Querying the event database

You can run the full range of ObjectServer SQL commands and perform queries against ObjectServer data.

Before you begin

To perform this task, the following prerequisites must be met:

- The *ncw_admin* role must be assigned to your Web GUI user profile.
- You need to be a user in the ObjectServer; your user must be a member of the following groups: ISQL and ISQLWrite.

Note: If LDAP is used for authentication, and the **users.credentials.sync** property of the *server.init* file is enabled, LDAP users are automatically

synchronized with the ObjectServer. If not, a user must be created in the ObjectServer and assigned to the groups.

- In the ObjectServer, the `webtop_compatibility` automation, which is a temporal trigger, and must have run. By default, this trigger is enabled

Tip: You can reduce the frequency with which the trigger runs from the default of 60 minutes. For more information about configuring temporal triggers, see the *IBM Tivoli Netcool/OMNIBus Administration Guide*.

- The Web GUI configuration cache must be synchronized with the ObjectServer. The cache can be synchronized by running the `webgui_home_dir/bin/webtop_osresync` script.

The Web GUI event database query is analogous in behavior to the Tivoli Netcool/OMNIBus SQL Interactive Interface, and can be used to perform tasks such as creating a new database table or stopping the ObjectServer. You can connect to ObjectServers and use SQL commands to interact with and control the ObjectServer.

Attention: Use the Event Database Query with care. Any command entered is run against the ObjectServer. Incorrect use of SQL commands can result in the irreparable corruption of your database.

To send SQL instructions to the ObjectServer:

1. Click **Administration > Event Management Tools > Event Database Query**.
2. From the **Select data source** list, select a data source against which you want to run the command.
3. Type an SQL query in the text editor provided and click **Submit**.
The result of the SQL query is displayed in the **Results displayed below** area.

Sample SQL query

The following example result shows an excerpt of the data returned after submitting the following query:

```
select Node, Summary from alerts.status where Severity=4;
```

```
-----
select Node, Summary from alerts.status where Severity=4
-----
>Executed 'select Node, Summary from alerts.status where Severity=4;'
Tokyo Diskspace alert
sl-server2.ibm.com A process running on sl-server2.ibm.com has connected as
username bertha
sl-server2.ibm.com A process running on sl-server2.ibm.com has connected as
username bertha
London Machine has gone offline
sl-server2.ibm.com A process running on sl-server2.ibm.com has connected as
username herbert
sl-server2.ibm.com A process running on sl-server2.ibm.com has connected as
username bertha
sl-server2.ibm.com A process running on sl-server2.ibm.com has connected as
username filter
sl-server2.ibm.com A process running on sl-server2.ibm.com has connected as
username root
Shanghai Link Down on port
sl-server2.ibm.com A process running on sl-server2.ibm.com has connected as
username bertha
```

See the information about ObjectServer SQL syntax in the chapter that explains ObjectServer SQL in the *IBM Tivoli Netcool/OMNIBus Administration Guide*.

For more information about ObjectServer SQL syntax, go to the IBM® Tivoli® Network Management Information Center at <http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/index.jsp>, and search for *ObjectServer SQL*.

Copying data between Web GUI servers

The Web GUI has a utility to copy selected data from one server to another.

Overview

Occasionally you need to copy data from one Web GUI V7.3.1 server to another. One example is when copying data from a test server to a production server. The Web GUI contains a utility for such situations.

The server you are copying data from is the *source server*; the server you are copying data to is the *target server*. Either server can be a stand-alone system or part of a load balancing cluster.

The copying process

The process for copying data between servers is:

1. Define the data to export from the source server.
2. Export the data from the source server.

The utility writes the selected data to a .zip file.

3. Copy the file to the target server.
4. Set up the utility to import data from the .zip file.

In this step you can define whether to import all the data from the file or selected items.

5. Import the data from the .zip file.
6. Restart the target server, if necessary.

Tip: In many cases, you do not need to restart the server if your system uses timed tasks.

Items that you can copy between servers

The items that you can copy between servers are:

- Filters
- Maps
- Menus
- Prompts
- Tools
- Views
- Any file or directory in *webgui_home_dir* or in *tip_home_dir*/profiles/TIPProfile/installedApps/TIPCell/isc.ear/OMNIBusWebGUI.war

You can also specify files and directories not to copy between servers. For example the Web GUI configuration file (*server.init*).

Some items, such as maps, often have dependent items that define their behavior. When exporting such an item the utility includes all the dependent items. This makes it easier to move complex items from one Web GUI server to another.

Cloning data between servers

On occasions you may need to copy all Web GUI and Tivoli Integrated Portal data from one server to another. This process is known as *cloning* and is useful when transferring a set up from a test to a production environment. The process for cloning is similar to the main copying procedure except that you do not have to define the items to export and import. Instead, the utility provides a file that defines all the items required. You specify this file when exporting and importing data. This simplifies the process of cloning and reduces the possibility of any key data not being copied.

Note: The cloning process excludes the `ncwDataSourceDefinitions.xml` file.

Exporting data from a Web GUI server

First export the required data from the source server.

Defining the items to export:

Define the items to export and the location for the .zip file.

1. Make sure you are logged in as an administrative user.
2. Navigate to the directory containing the utility's properties file:
`cd webgui_home_dir/integration/importexport_tool/etc`
3. Edit the file `OMNIBusWebGUI_settings.properties` to define the items you want to export from the source server:

Many of the property definitions are a list of items. In these lists, put a comma between each item. In addition for properties that define file paths, use two backslash characters as the path separator on Windows systems; for example:
`C:\\IBM\\tivoli\\netcool\\omnibus_webgui`.

Set the following properties as required. To set the property, remove the comment marker at the beginning of its line and the modify the value as required. To omit a particular item from the export, leave the property commented out. In some cases the utility uses a default value for omitted properties.

Table 2. Setting properties to define the data to export

Property	Value
product.home	The installation directory of the Web GUI. Set this property if did not install the product into the default location. The default installation location is one of the following: <div><div>UNIX</div><div>Linux</div> ibm/tivoli/netcool/ omnibus_webgui <div>Windows</div> C:\\IBM\\tivoli\\netcool\\ omnibus_webgui</div>
output.dir	The full path of the directory to receive the exported data. The default value is <code>tip_home_dir/profiles/TIPProfile/output</code> .

Table 2. Setting properties to define the data to export (continued)

Property	Value
ExportWebGUIPlugin.input	<p>To export data from the OMNIBusWebGUI.war file remove the comment marker from the beginning of the definition of the ExportWebGUIPlugin.input property and add a comment marker to the element for the Web GUI folder.</p> <p>Add any further files and directories to export to the definition of the remaining property. Specify each directory and file path as relative to <i>webgui_home_dir</i>.</p>
ExportWebGUIPlugin.ignore	Specify the files and directories to exclude from the exported data. Put a comma between each file or directory path. Specify each directory and file path as relative to <i>webgui_home_dir</i> .
ExportPromptPlugin.input	A list of the names for prompts (without their extensions) to export.
ExportToolPlugin.input	A list of names for tools (without their extensions) to export.
ExportMenuPlugin.input	A list of names for menus (without their extensions) to export.
ExportMenuConfigPlugin.input	A list of names for menu configurations (without their extensions) to export.
ExportViewPlugin.global	A list of names for global views (without their extensions) to export. To export all global views use the forward slash (/) character as the value for this property.
ExportViewPlugin.system	A list of names for system views (without their extensions) to export. To export all system views use the forward slash character (/) as the value for this property.
ExportViewPlugin.user	<p>A list of names for user-defined views (without their extensions) to export. For each view, specify the user ID and view name separated by a comma.</p> <p>For example, to specify View1 and View2 for user ID User1 and View3 for User ID User 2, use a definition like this:</p> <p>ExportViewPlugin.user=User1,View1, User1,View2, User2,View3</p> <p>To export all user views, use the forward slash character (/) as the value for this property.</p>
ExportFilterPlugin.global	A list of names for global filters (without their extensions) to export. To export all global filters, use the forward slash character (/) as the value for this property.
ExportFilterPlugin.system	A list of names for system filters (without their extensions) to export. To export all system filters, use the forward slash character (/) as the value for this property.

Table 2. Setting properties to define the data to export (continued)

Property	Value
ExportFilterPlugin.user	<p>A list of names for user-defined filters (without their extensions) to export. For each filter, specify the User ID and filter name separated by a comma.</p> <p>For example, to specify Filter1 and Filter2 for user ID User1 and filter3 for User ID User 2, use a definition like this:</p> <p>ExportFilterPlugin.user=User1,Filter1, User1,Filter2, User2,Filter3</p> <p>To export all user filters, use the forward slash character (/) as the value for this property.</p>
ExportFilterCollectionPlugin.input	<p>A list of names for filter collections (without their extensions) to export. The utility includes all the related objects for each collection such as: filters, dependency filters, and views.</p>
ExportMapPlugin.input	<p>A list of names for maps (without their extensions) to export. The utility includes all the related objects for each map such as: map resources, filters, dependency filters, and views.</p>

Exporting the data:

Export the defined data to a .zip file on the source server.

1. Make sure you are logged in to the source server as an administrative user and that the Tivoli Integrated Portal is running.
2. As supplied, the utility writes information, warning, and error messages to the log file. Change the level of logging, if required.
3. Navigate to the directory containing the utility, *tip_home_dir/profiles/TIPProfile/bin*.
4. Enter one of the following commands to export the data:

```

UNIX      Linux      ./tipcli.sh Export --username tipadmin --password
tippass --excludePlugins ExportPagePlugin,ChartExportPlugin
--settingFile webgui_home_dir/integration/importexport_tool/etc/
OMNIBusWebGUI_settings.properties

```

```

Windows   tipcli.bat Export --username tipadmin --password tippass
--excludePlugins ExportPagePlugin,ChartExportPlugin --settingFile
webgui_home_dir\integration\importexport_tool\etc\
OMNIBusWebGUI_settings.properties

```

Replace:

tipadmin

with the name of the Tivoli Integrated Portal administrative user.

tippass

with the password for the Tivoli Integrated Portal administrative user.

To exclude any plug-ins from the export operation, add their names to the **--excludePlugins** qualifier, putting a comma between each plug in name. For example, to exclude the prompt plug-in, use:

```
--excludePlugins ExportPagePlugin,ChartExportPlugin,ExportPromptPlugin
```

Always include ExportPagePlugin and ChartExportPlugin in the list.

Results

The utility creates:

- The file of data in `data.zip` within the directory specified by the **output.dir** property of `OMNIBusWebGUI_settings.properties`. If that property is not set, the file is in `tip_home_dir/profiles/TIPProfile/output`.
- A log file in `tip_home_dir/profiles/TIPProfile/logs/tipcli.log`.

Related tasks

“Setting the logging level for the utility” on page 20

Verifying the export is complete:

Before copying the `.zip` file to the target server, make sure that it contains all required directories, files, and Web GUI objects.

1. Open the log file in `tip_home_dir/profiles/TIPProfile/logs/tipcli.log` and check that all stages of the export operation completed successfully.
Resolve any errors and repeat the export activity before continuing. For example, correct any misspelled names and paths in the `OMNIBusWebGUI_settings.properties` file, repeat the export operation, and then verify that the errors are resolved.
2. Navigate to the output directory and open the `data.zip` file using a suitable file compression tool.
3. Check that the file contains all the files, directories, and Web GUI objects that you wanted to export.
If any items are missing, edit `OMNIBusWebGUI_settings.properties` and add the missing items. Then repeat the export operation and verify that the items are now present.
4. Close the `data.zip` file.

Importing data to a Web GUI server

Secondly import the data to the target server.

Defining the items to import:

Copy the exported file to the target server, install the utility (if necessary), and define the items to import from the file.

1. Copy the `thedata.zip` file from the source server to a suitable directory on the target server.
2. Make sure you are logged in to the target server as an administrative user and that the Tivoli Integrated Portal server is running.
3. Navigate to the directory containing the utility's properties file:
`cd webgui_home_dir/integration/importexport_tool/etc`
4. Edit the file `OMNIBusWebGUI_settings.properties` to define the items you want to import from the `thedata.zip` file.

Many of the property definitions are a list of items. In these lists, put a comma between each item. In addition for properties that define file paths, use two backslash characters as the path separator on Windows systems; for example:
`C:\\IBM\\tivoli\\netcool\\omnibus_webgui.`

Set the following properties as required. To set a property, remove the leading comment marker and then provide a suitable value.

Table 3. Setting properties to define the data to import

Property	Value
product.home	<p>The installation directory of the Web GUI. Set this property if did not install the product into the default location. The default installation location is one of the following:</p> <div> <div>UNIX</div> <div>Linux</div> <div>ibm/tivoli/netcool/omnibus_webgui</div> </div> <div> <div>Windows</div> <div>C:\IBM\tivoli\netcool\omnibus_webgui</div> </div>
output.dir	<p>The full path of the directory where you placed data.zip.</p> <p>If you do not define this property, the utility expects to find the file in <i>tip_home_dir/profiles/TIPProfile/output</i></p>
import.backupDir	<p>The full path of the directory the utility uses to store backup copies of files it imports. Specify a directory other than output.dir, although you can specify a subdirectory in output.dir.</p> <p>If you do not define this property, the utility uses <i>tip_home_dir/profiles/TIPProfile/backups</i>.</p>

Related tasks

“Administering timed tasks” on page 7

Importing the data:

Import the data from the .zip file to the target server.

1. As supplied, the utility writes information, warning, and error messages to the log file. Change the level of logging, if required.
2. Navigate to the directory containing the utility, *tip_home_dir/profiles/TIPProfile/bin*.
3. Enter one of the following commands to import the data:

```


UNIX



Linux


./tipcli.sh Import --username tipadmin --password tippass --excludePlugins ImportPagePlugin,ChartImportPlugin --settingFile webgui_home_dir/integration/importexport_tool/etc/OMNIBusWebGUI_settings.properties

```

```


Windows


tipcli.bat Import -username tipadmin --password tippass --excludePlugins ImportPagePlugin,ChartImportPlugin --settingFile webgui_home_dir\integration\importexport_tool\etc\OMNIBusWebGUI_settings.properties

```

Replace:

tipadmin

with the name of the Tivoli Integrated Portal administrative user.

tippass

with the password for the Tivoli Integrated Portal administrative user.

To exclude any plug-ins from the import operation, add their names to the **--excludePlugins** qualifier, putting a comma between each plug in name. For example, to exclude the prompt plug-in, use:

```
--excludePlugins ImportPagePlugin,ChartImportPlugin,ImportPromptPlugin
```

Always include the **ImportPagePlugin** and **ChartImportPlugin** in the list.

4. Restart the Tivoli Integrated Portal server, if necessary.

You need to restart the server if your site does not use timed tasks or you have imported a key system file such as `server.init` or `ncwDataSourceDefinitions.xml`.

Related tasks

“Setting the logging level for the utility” on page 20

“Restarting the server” on page 1

Verifying the import is complete:

Check the log file to make sure the import completed successfully, that all the required files and directories are present, and that the imported facilities are available in the Web GUI.

1. Open the log file in `tip_home_dir/profiles/TIPProfile/logs/tipcli.log` and check that all stages of the import operation completed successfully.
Resolve any errors and repeat the import activity before continuing. For example, correct any misspelled names and paths in the `OMNIBusWebGUI_settings.properties` file, repeat the import operation, and then verify that the errors are resolved.
2. Verify that all required files, directories, and Web GUI objects are present on the target server.
3. Verify that all the facilities are available in the Web GUI.
For example, verify that all the imported filters are available.
4. In a load balancing cluster, verify that all the files, directories, and Web GUI objects have propagated to all nodes in the cluster.

Cloning data between servers

Use this procedure to copy all Web GUI with or without Tivoli Integrated Portal data, except `ncwDataSourceDefinitions.xml`, between servers running Web GUI V7.3.1.

Exporting the data:

Export all the Web GUI with or without Tivoli Integrated Portal data to a .zip file on the source server.

Related tasks

“Setting the logging level for the utility” on page 20

Export Web GUI and Tivoli Integrated Portal data:

1. Make sure you are logged in to the source server as an administrative user and that the Tivoli Integrated Portal server is running.
2. As supplied, the utility writes information, warning, and error messages to the log file. Change the logging level, if required.
3. If your installation of the Web GUI does not use the default location:
 - a. Navigate to the directory `webgui_home_dir/integration/plugins`.
 - b. Edit the file `OMNIBusWebGUI.properties`.
 - c. Remove the comment marker (#) for the following property and set its value to the installation directory of the Web GUI:
`product.home=webgui_home_dir`

Replace *webgui_home_dir* with the actual installation directory of the Web GUI.

Note: The default installation directory of the Web GUI is one of the following:

UNIX **Linux** `ibm/tivoli/netcool/omnibus_webgui`

Windows `C:\IBM\tivoli\netcool\omnibus_webgui`

4. Navigate to the directory containing the utility, *tip_home_dir*/profiles/TIPProfile/bin.
5. Enter one of the following commands to export the data:

UNIX **Linux** `./tipcli.sh Export --username tipadmin --password tippass --settingFile webgui_home_dir/integration/plugins/OMNIBusWebGUI.properties`

Windows `tipcli.bat Export --username tipadmin --password tippass --settingFile webgui_home_dir\integration\plugins\OMNIBusWebGUI.properties`

Replace:

tipadmin

with the name of the Tivoli Integrated Portal administrative user.

tippass

with the password for the Tivoli Integrated Portal administrative user.

Results

The utility creates the following files:

- The file of data in *data.zip* within the directory *tip_home_dir*/profiles/TIPProfile/output.
- A log file in *tip_home_dir*/profiles/TIPProfile/logs/tipcli.log.

Exporting Web GUI data only:

1. Make sure you are logged in to the source server as an administrative user and that the Tivoli Integrated Portal server is running.
2. As supplied, the utility writes information, warning, and error messages to the log file. Change the logging level, if required.
3. If your installation of the Web GUI does not use the default location:
 - a. Navigate to the directory *webgui_home_dir*/integration/plugins.
 - b. Edit the file *OMNIBusWebGUI_clone_settings.properties*.
 - c. Remove the comment marker (#) for the following property and set its value to the installation directory of the Web GUI:
`product.home=webgui_home_dir`
Replace *webgui_home_dir* with the actual installation directory of the Web GUI.

Note: The default installation directory of the Web GUI is one of the following:

UNIX **Linux** `ibm/tivoli/netcool/omnibus_webgui`

Windows `C:\IBM\tivoli\netcool\omnibus_webgui`

4. Navigate to the directory containing the utility, *tip_home_dir*/profiles/TIPProfile/bin.
5. Enter one of the following commands to export the data:

```

UNIX Linux ./tipcli.sh Export --username tipadmin --password
tippass --excludePlugins ExportPagePlugin,ChartExportPlugin
--settingFile webgui_home_dir/integration/plugins/
OMNIBusWebGUI_clone_settings.properties

```

```

Windows tipcli.bat Export --username tipadmin --password tippass
--excludePlugins ExportpagePlugin,ChartExportPlugin --settingFile
webgui_home_dir\integration\plugins\
OMNIBusWebGUI_clone_settings.properties

```

Replace:

tipadmin

with the name of the Tivoli Integrated Portal administrative user.

tippass

with the password for the Tivoli Integrated Portal administrative user.

Results

The utility creates the following files:

- The file of data in *data.zip* within the directory *tip_home_dir/profiles/TIPProfile/output*.
- A log file in *tip_home_dir/profiles/TIPProfile/logs/tipcli.log*.

Importing the data:

Import the data from the *.zip* file to the target server.

Related tasks

“Setting the logging level for the utility” on page 20

“Restarting the server” on page 1

Import Web GUI and Tivoli Integrated Portal data:

1. Copy *data.zip* from the source server to the directory *tip_home_dir/profiles/TIPProfile/output* on the target server.
2. On the target server make sure that you are logged in as an administrative user and that the Tivoli Integrated Portal server is running.
3. As supplied, the utility writes information, warning, and error messages to the log file. Change the level of logging, if required.
4. If your installation of the Web GUI does not use the default location:
 - a. Navigate to the directory *webgui_home_dir/integration/plugins*.
 - b. Edit the file *OMNIBusWebGUI.properties*.
 - c. Remove the comment marker from the following property and set its value to the installation directory of the Web GUI:


```
product.home=webgui_home_dir
```

 Replace *webgui_home_dir* with the actual installation directory of the Web GUI.

Note: The default installation directory of the Web GUI is one of the following:

```

UNIX Linux ibm/tivoli/netcool/omnibus_webgui

```

```

Windows C:\IBM\tivoli\netcool\omnibus_webgui

```

5. Navigate to the directory containing the utility, *tip_home_dir/profiles/TIPProfile/bin*.

6. Enter one of the following commands to import the data:

```
UNIX      Linux      ./tipcli.sh Import --username tipadmin --password  
tippass --settingFile webgui_home_dir/integration/plugins/  
OMNIBusWebGUI.properties
```

```
Windows   tipcli.bat Import --username tipadmin --password tippass  
--settingFile webgui_home_dir\integration\plugins\  
OMNIBusWebGUI.properties
```

Replace:

tipadmin

with the name of the Tivoli Integrated Portal administrative user.

tippass

with the password for the Tivoli Integrated Portal administrative user.

7. Verify that the utility has added or updated files as required:
 - a. Check the log file, *tip_home_dir*/profiles/TIPProfile/logs/tipcli.log, and ensure there were no errors.
 - b. Verify that backup copies of the original files on the target server are in a .zip file in *tip_home_dir*/profiles/TIPProfile/backups.
8. Optional: Edit the *ncwDataSourceDefinitions.xml* file and apply any settings you want to copy over from the existing source server.
9. Restart the Tivoli Integrated Portal server.

Note: If target server is part of a load balancing cluster, wait until the next timed task schedule completes before restarting the server. This ensures that the imported data is replicated to other nodes in the cluster and the data base.

Import Web GUI data only:

1. Copy *data.zip* from the source server to the directory *tip_home_dir*/profiles/TIPProfile/output on the target server.
2. On the target server make sure that you are logged in as an administrative user and that the Tivoli Integrated Portal server is running.
3. As supplied, the utility writes information, warning, and error messages to the log file. Change the level of logging, if required.
4. If your installation of the Web GUI does not use the default location:
 - a. Navigate to the directory *webgui_home_dir*/integration/plugins.
 - b. Edit the file *OMNIBusWebGUI_clone_settings.properties*.
 - c. Remove the comment marker from the following property and set its value to the installation directory of the Web GUI:
product.home=webgui_home_dir
Replace *webgui_home_dir* with the actual installation directory of the Web GUI.

Note: The default installation directory of the Web GUI is one of the following:

```
UNIX      Linux      ibm/tivoli/netcool/omnibus_webgui  
Windows   C:\IBM\tivoli\netcool\omnibus_webgui
```

5. Navigate to the directory containing the utility, *tip_home_dir*/profiles/TIPProfile/bin.
6. Enter one of the following commands to import the data:

```
UNIX Linux ./tipcli.sh Import --username tipadmin --password tippass --settingFile webgui_home_dir/integration/plugins/OMNIBusWebGUI_clone_settings.properties
```

```
Windows tipcli.bat Import --username tipadmin --password tippass --settingFile webgui_home_dir\integration\plugins\OMNIBusWebGUI_clone_settings.properties
```

Replace:

tipadmin

with the name of the Tivoli Integrated Portal administrative user.

tippass

with the password for the Tivoli Integrated Portal administrative user.

7. Verify that the utility has added or updated files as required:
 - a. Check the log file, *tip_home_dir/profiles/TIPProfile/logs/tipcli.log*, and ensure there were no errors.
 - b. Verify that backup copies of the original files on the target server are in a .zip file in *tip_home_dir/profiles/TIPProfile/backups*.
8. Optional: Edit the *ncwDataSourceDefinitions.xml* file and apply any settings you want to copy over from the existing source server.
9. Restart the Tivoli Integrated Portal server.

Note: If target server is part of a load balancing cluster, wait until the next timed task schedule completes before restarting the server. This ensures that the imported data is replicated to other nodes in the cluster and the data base.

Setting the logging level for the utility

As supplied, the export/import utility writes information messages to the log file in addition to warnings and errors. Use this procedure to customize the level of logging during export and import operations.

1. Navigate to the directory containing the logging properties file:

```
cd tip_home_dir/profiles/TIPProfile/etc/
```
2. Open the file *logging.properties* in a text editor.
3. Locate the property **java.util.logging.FileHandler.level** and set its value to the required level. Commonly used values, in increasing severity, are:
 - INFO
 - WARNING
 - SEVERE

Other available values are FINE, FINER, and FINEST. Each of these produces increasing numbers of messages. They can be helpful when diagnosing a problem in an import or export utility. After using one of these values, however, be sure to return to the original value before recommencing normal operations on the server.

For example, setting the property to WARNING, excludes INFO messages from the log file.

4. Save the file and exit from the text editor.

Backing up and restoring data

To prevent the loss of information in the event of a disaster, and for disaster recovery, back up your installation of the Web GUI and the Tivoli Integrated Portal.

Related tasks

“Restarting the server” on page 1

Backing up and restoring the Web GUI

Use the Web GUI export/import utility to back up and restore Web GUI configuration data.

- To back up the Web GUI configuration data, use the export facility of the Web GUI export/import utility. This creates a .zip file that you can copy to a secure place.
- To restore previously backed up Web GUI data, use the import facility of the Web GUI export/import utility. Edit the properties file to specify the files you want to restore and then import the data.

Related tasks

“Copying data between Web GUI servers” on page 10

Backing up and restoring the Tivoli Integrated Portal

Use the Tivoli Integrated Portal export/import facilities to back up and restore the Tivoli Integrated Portal configuration data.

- To back up the Tivoli Integrated Portal configuration data, use the Tivoli Integrated Portal export facility. This creates a .zip file that you can copy to a secure place.
- To restore previously backed up Tivoli Integrated Portal data, use the Tivoli Integrated Portal import facility.

Related concepts

“Exporting and importing Tivoli Integrated Portal data” on page 47

Restore data from previous releases of the Web GUI and the Tivoli Integrated Portal

Use the System Cloning Solution (SCS) to restore data from a previous version of the Web GUI or Tivoli Integrated Portal.

Previous releases of the Web GUI and the Tivoli Integrated Portal used the System Cloning Solution (SCS) to back up and restore configuration data. So, use SCS to restore data from a backup of a previous version.

Running SCS to export data:

Use the System Cloning Solution (SCS) to export instances of the Tivoli Integrated Portal Server. Exported settings can be later applied to another server instance at the same version level with the same products deployed.

To export settings for a Tivoli Integrated Portal Server instance:

1. On the command-line interface, change to the *tip_home_dir*/profiles/TIPProfile/bin directory. The *tip_home_dir* directory defaults to C:\IBM\tivoli\tipv2 on Windows and /opt/IBM/tivoli/tipv2 on UNIX/Linux
2. Run the following command:

```
ws_ant.bat|sh -f tipExportImport.xml export -DarchiveDir=dir  
-DtipAdmin=tipadmin -DtipPassword=tipass
```

The export argument results in the script copying all required data from the TIPProfile profile into the directory specified by *dir* in the *archiveDir* option.

Note: To avoid the accidental loss of existing user data, the export script fails if the specified archive directory exists. Please specify a nonexistent directory for the *archiveDir* option.

Replace *tipadmin* with the Tivoli Integrated Portal administrator ID and *tippass* with the Tivoli Integrated Portal administrator password.

Run the command with the export argument on the source Tivoli Integrated Portal Server server.

Running SCS to import data:

Use the System Cloning Solution (SCS) to import settings to a target Tivoli Integrated Portal Server instance. The target server instance must have the same configuration as the server instance from which the settings were sourced.

Before you begin

The Tivoli Integrated Portal cloning procedure does not automatically perform a backup of the target system in a cloning import operation. It is recommended that you export the target system as a backup operation.

This is accomplished by running the System Cloning Solution export option on the target server before running the import of the data exported from the source system. If the import fails, the backup archive can be imported to restore the system to its original state.

Important: The target server instance should not be configured for load balancing. The cloning process imports data for a local server instance only.

To import settings for a Tivoli Integrated Portal Server instance:

1. On the command-line interface, change to the *tip_home_dir*/profiles/TIPProfile/bin directory. The *tip_home_dir* directory defaults to C:\IBM\tivoli\tipv2 on Windows and /opt/IBM/tivoli/tipv2 on UNIX/Linux
2. Run the following command:

```
ws_ant.bat|sh -f tipExportImport.xml import -DarchiveDir=dir  
-DtipAdmin=tipadmin -DtipPassword=tippass -DexcludesFile=TBSM_HOME/etc/  
cloneExcludesFile
```

The import argument is used to import data from an existing archive directory, specified by replacing *dir* in the *archiveDir* option, which overwrites the Tivoli Integrated Portal Server instance to complete the cloning. Run the command with the import argument on the target Tivoli Integrated Portal Server instance.

Replace *tipadmin* with the Tivoli Integrated Portal administrator ID and *tippass* with the Tivoli Integrated Portal administrator password. They must have the same values as the source Tivoli Integrated Portal Server instance.

The excludesFile option must be provided and must point to the file specified above. This file is provided with TBSM 4.2.1 Fix Pack 1 and is located in TBSM_HOME/etc. Replace TBSM_HOME with the TBSM install directory for your server. The default for Windows is C:\IBM\tivoli\tbsm and /opt/IBM/Tivoli/tbsm for UNIX and Linux operating systems. This file gives TBSM the flexibility to exclude some configuration files from being imported by the utility.

Backing up and restoring the Deployment Engine

Use the Deployment Engine (DE) backup script before installing additional components or other products that are based on the Tivoli Integrated Portal platform. If you need to recover the original configuration after a failure, you can then run the Deployment Engine restore script.

The Deployment Engine performs the installation of new and upgraded products. It keeps track of the installed components and skips installing a given component if it is already present on the system. Perform the following steps to back up or restore the DE database.

1. From the command line, change to the `acsi` directory:

- **Windows** `cd C:\Program Files\IBM\Common\acsi`
- **Linux** **UNIX** For Linux and UNIX-based systems, the path to the `acsi` directory varies depending on whether you are installing as root or as a non-root user, as follows:
 - Installing as a non-root user, the path is relative to the user's home directory:
`<non-root user home directory>/.acsi<user_name>`
 - Installing as root, the path is as follows:
`/var/ibm/common/acsi`

2. Initialize the Deployment Engine environment from the command line:

- **Windows** `setenv.bat`
- **Linux** **UNIX** `. setenv.sh`

3. Change to the `bin` directory:

- **Windows** Change to the `bin` child directory, that is:
`C:\Program Files\IBM\Common\acsi\bin`
- **Linux** **UNIX** For Linux and UNIX-based systems, the path to the `bin` directory varies depending on whether you are installing as root or as a non-root user, as follows:
 - For a non-root user, change to the `bin` child directory, that is:
`<non-root user home directory>/.acsi<user_name>/bin`
 - For root, the path is as follows:
`/usr/ibm/common/acsi/bin`

4. Run the backup script to back up the Deployment Engine database, as follows:

- **Windows** `de_backupdb.cmd`
- **Linux** **UNIX** `de_backupdb`

5. If you need to restore the Deployment Engine database, from the `bin` directory run the restore script:

- **Windows** `de_restoredb.cmd`
- **Linux** **UNIX** `de_restoredb`

What to do next

If you backed up the Deployment Engine database, you can run the installer now to add additional components or products. If you restored the Deployment Engine database, you can resume using the original installed environment.

Administering the GUI framework

You can use the functions of Tivoli Integrated Portal to administer the setup of your Web GUI installation.

Tip: If you do not find the information that you require in the *IBM Tivoli Netcool/OMNIBus* information center, see the *IBM Websphere Application Server* information center at the following Web address:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/topic/com.ibm.websphere.zseries.doc/info/welcome_nd.html

Tip: If you do not find the information that you require in this publication, see the *IBM Websphere Application Server* information center at the following Web address:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/topic/com.ibm.websphere.zseries.doc/info/welcome_nd.html

Tivoli Integrated Portal layout

The layout of the console user interface has these major elements.

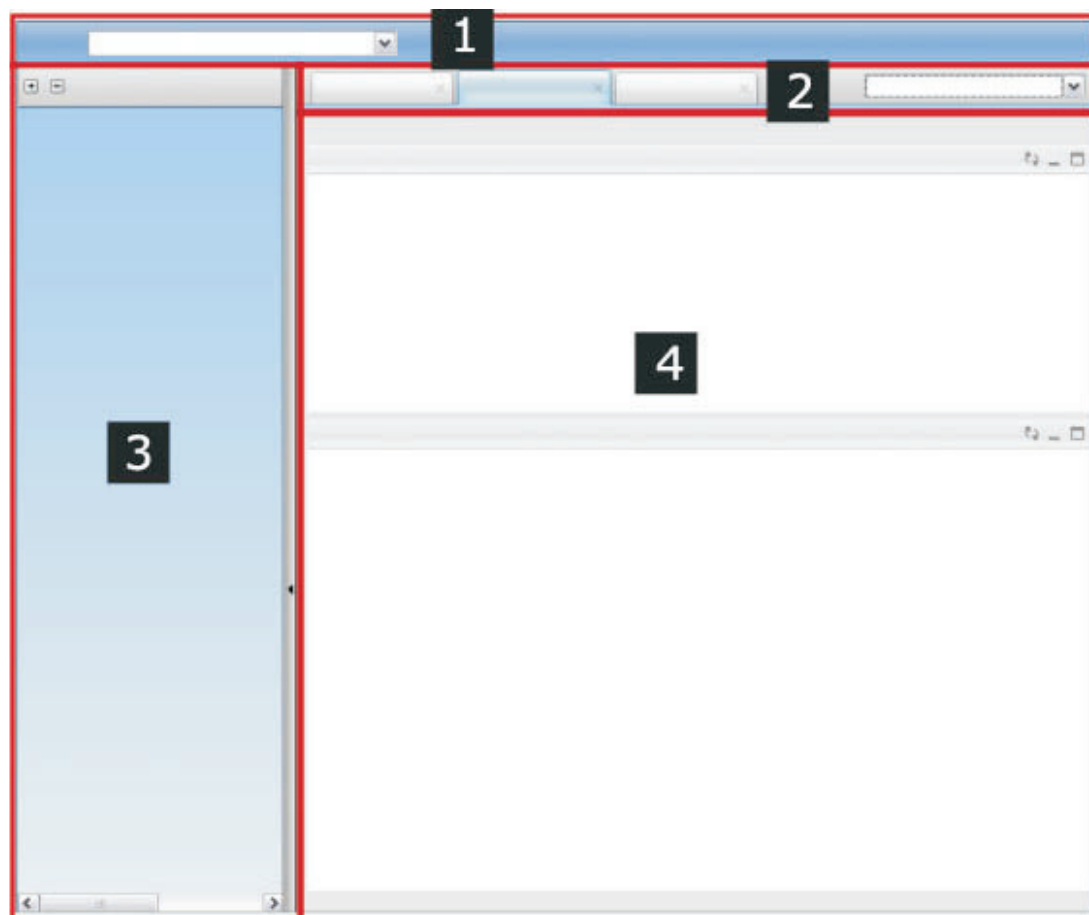


Figure 1. Console layout

1 Banner

Displays a common image across all console installations. The banner

includes a greeting to the user as well as links to log out of the console and to open console help. The **View** selection list in the banner controls which nodes are displayed in the navigation as well as pages that are opened when the view is selected.

2 Page bar

Displays tabs to select between open pages. The page bar allows you to work on different pages without closing the page or losing unsaved data. For example, if you are working on an application on Page A, you can open an application on another page to gather information about a resource that you need to finish the form on Page A without losing any unsaved data you have already entered. Multiple pages can be opened at one time, but only one of the open pages is in focus (*current page*). The page bar also contains a **Select Action** drop-down list for performing actions on the current page.

3 Navigation pane

Displays a set of navigation nodes used for accessing content. The nodes shown in the navigation pane are only those to which you have access.

4 Work area

Displays the current page that you are working on. The page contains one or more Web applications or *portlets*, each in its own portlet window with a title bar.

How to customize a Tivoli Integrated Portal setup

Use this information to learn about the resources provided in Tivoli Integrated Portal, so that you can administer your installation in line with your requirements.

To get started setting up the console, you should already be familiar with the concepts and characteristics of the console layout. You should take time navigating through the console to become familiar with the portlets, pages, views, roles, and preference profiles that are provided. As you work with the console, you will create some of these resources to suit your organization's needs.

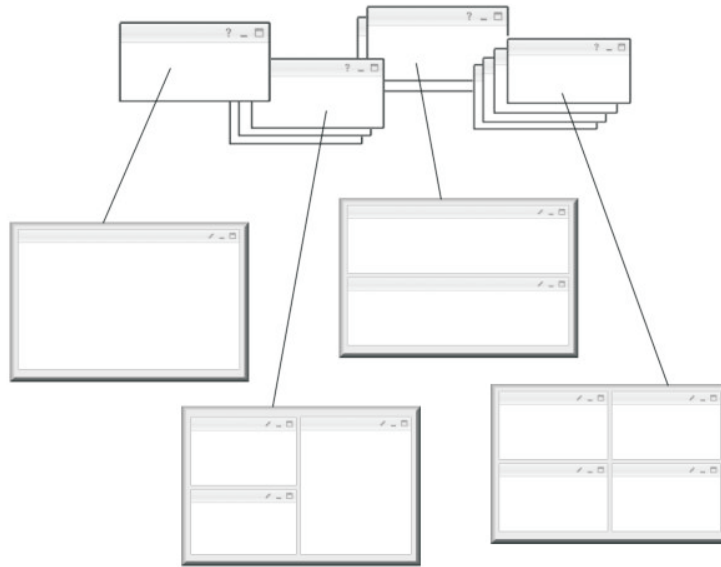
Understanding the structure of the console

Access to each level in the console organization is assigned based on the users' roles. Keep each role in mind when planning how to structure the console.

Content in the console is composed of portlets. The following figure shows how portlets are arranged on a page using a row and column layout. Access to each page, and to each portlet on each page, is assigned to users based on their defined role.

Portlets

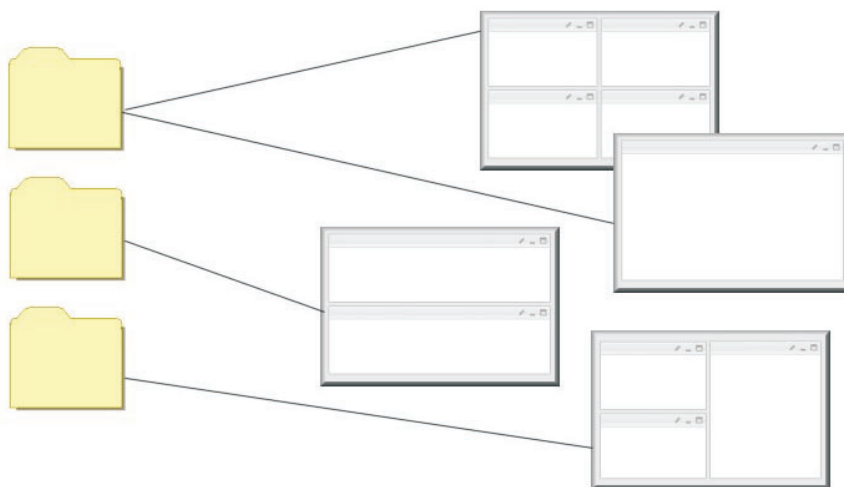
Pages



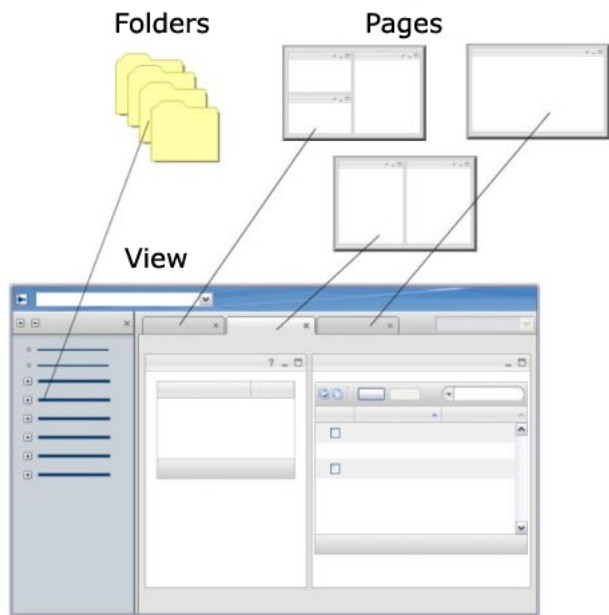
Each page is accessed from the console navigation, either from the console root or they can be grouped into folders. The hierarchical structure of the navigation affects how quickly users can find a page and work with the portlets on that page.

Folders

Pages



Folders and pages can be assembled into views that the user can select from the View drop-down list in the banner. Each view can include pages that are initially launched when the view is selected.



Finally, you can define a set of preferences, called a *preference profile*, that determines what views are available to each role, and whether the navigation should be displayed.

Related information

“Administering roles” on page 80

“Administering pages” on page 28

“Administering views” on page 36

“Administering portlets” on page 40

“Administering console preference profiles” on page 45

How to customize a Tivoli Integrated Portal setup

Use this information to learn about the resources provided in Tivoli Integrated Portal, so that you can administer your installation in line with your requirements.

1. Define your console users and what tasks they perform. Console users are assigned to roles, which are used to determine what tasks they can perform in the console. As you assess the users' tasks, think about how these roles will be defined. Consider how the community of console users will be assigned to different roles and whether there are any existing roles that you can use, or if you need to create new roles.
Roles can be created without assigning access to any resources. This step can be performed later.
2. Review the content. Users' tasks are performed using portlets on console pages. You need to understand what portlets are available and how they will be used to perform these tasks. For each portlet, determine which roles should have access and which roles should be restricted.
3. Create a navigation structure of pages and folders. Determine which pages are currently used to access the portlets. Are these pages sufficient for the roles that you have defined, or do you need to create new pages? For existing pages, do

you need to add or remove any portlets or change the way they are arranged on the page? Consider that multiple roles can access a page with different access to the portlets on that page.

Review the folders in the navigation and the pages that are contained in these folders. Do these folders help the users find their content? Do you need to edit existing folders or create new folders? Should you move any pages between folders? What folders or pages should be hidden for each role?

4. Organize the content and navigation into views. Determine which navigation folders and pages have a related purpose for each role. You can define one or more views for each role, and even make a single view appear differently between roles based on access control. Each view can also include one or more pages that are launched when the view is selected. Each of these options is provided to help remove other content and pages that can distract users.
5. Define the presentation for each role Determine which views should be available to users in a role. For some roles, you can remove the navigation bar and just provide a set of startup pages. You can assign exactly one preference profile per role.
6. Test the console for each role. Create a test user for each role. Log into the console as each user and verify the use cases.
 - The navigation is shown or not, depending on the setting in the console preference profile.
 - The view selection list shows only the views to which the role has access and as defined by the preference profile.
 - Each view shows only the navigation nodes and startup pages allowed for that role.
 - Each folder shows only the pages allowed for that role.
 - Each page launched in the navigation shows only the portlets allowed for that role.
 - If the role has Editor access to a page, the **Edit Page** option is available in the **Page Actions** selection list. This option is not showing if the user's role does not have Editor access.
 - Each page shows only the portlets allowed for that role.
 - The portlet title bar provides an **Edit Options** icon that provides access to two options, a **Personalize** option, and an **Edit Shared Settings** option. The **Personalize** option is available, if the user's role has Privileged User access. The **Edit Shared Settings** option is available if the user's role has Editor access. Otherwise, neither of these options are available.

Go back and make corrections as indicated by the results of your testing.

7. Move the console to production use. Assign roles to actual users and notify the user community that the console server is ready for use.

Administering pages

Console content is composed of pages, folders, and external URLs. Each of these resources is represented in the navigation pane as a node. Click **Settings > Page Management** to create, edit, and delete pages and folders for the console navigation. You can also edit external URLs that are launched from the navigation pane. You cannot create URLs in the console. Instead, URLs are created when an application is deployed to the console that includes the URL node in its descriptors.

Field descriptions

This section describes the fields and controls in the main panel of Page Management.

Select all icon

Selects all items displayed in the table for deletion. If you are displaying only a filtered set of items, only those items are selected. You can deselect specific items before actually deleting.

Deselect all icon

Deselects all items displayed in the table.

New Page

Opens a panel for creating a new page.

New Folder

Opens a panel for creating a new folder.

Delete Immediately deletes all selected items in the list. Only Custom resource types can be deleted.

Filter Type in this field to quickly find an item in the table. This field is useful when there are a large number of items to look through.

Select Selects or deselects a single item in the table.

Name Displays the title of the page as it is shown in the navigation.

Type Displays the type of page.

Unique Name

Displays the string used by the system to uniquely identify the page or folder.

Related tasks

“Creating startup pages” on page 35

Related information

“Administering portlets” on page 40

“Administering roles” on page 80

Creating pages

To create a page, you must first select content for the page and specify the layout of the portlet window. You must then set the properties of the page, including the page name and its location in the navigation pane. All pages that are created in the console have a resource type of “Custom”.

To create a page for testing purposes:

1. To display the Create New Work Page portlet:
 - In the taskbar, click the **Create page** tab.
 - In the navigation pane, click **Settings > Page Management** and in the Page Management portlet, click **New Page**.

A Page Settings page is displayed.

2. In the **Page name** field, provide a descriptive name for the page and in the **Page location** field indicate where you want the page to be displayed in the navigation pane. Consider the content on the page and how users will find that content by looking for the page name in the navigation pane.
3. Optional: Click the **Optional Setting** label and associate one or more roles with the new page and set the level of access for each role.

4. Click **Save**. The taskbar tab is updated with the name of the new page and a Choose a Portlet window is displayed.
5. To add a portlet, scroll through the list and select a portlet or use the **Filter** field to find the portlet you want to add.
6. Optional: Use the **Horizontal split** icon or the **Vertical split** icon to add more portlet containers to the page and select a portlet for each section.
7. Click **OK** in the Choose a Portlet window. The selected portlet is displayed.
8. Click **Save** to commit your changes.

Results

The new page is displayed. Users with “editor” access to the page can add more content, arrange the content using horizontal and vertical layouts, and replace and remove content.

What to do next

Make sure that the roles with access to this page also have access to the portlets that are on the page. You can also edit the new page to customize its page persistence settings and text direction settings.

Related tasks

“Publishing charts” on page 344

“Creating startup pages” on page 35

Editing page content and layout

Pages are an arrangement of one or more portlets in the work area and contain the portlets needed to complete tasks. Users whose roles have “Editor” access to a page can edit a page's layout and content using the **Edit Page** option in the page action list. After saving changes to the layout and content, you can change a page's properties, including its name and location in the navigation.

Note: User's with “Privileged User” access can change the size of portlet windows on the page.

1. Locate the page you want to edit in the navigation pane and open it.
2. In the page bar, select **Edit Page** from the page actions selection list. The page is changed to show buttons at the top. Each portlet title bar displays new icons for creating horizontal and vertical layouts and replacing and removing portlet content.
3. Optional: To add more portlets to the page, follow these steps.
 - a. Create a window for the new portlet by splitting one of the windows displayed.
 - Use the **Horizontal split** icon to create a window below an existing window.
 - Use the **Vertical split** icon to create a window to the right of an existing window.

The Portlet Picker is displayed within the new portlet window for selecting the portlet content.

- b. Scroll through the list or use the **Filter** field to find the portlet you want to add.
 - c. Click **OK**. The portlet is added to the window.
 4. Optional: To replace a portlet in a window, follow these steps.

- a. Click the **Replace content** icon in the title bar where you want to replace the portlet content. The Portlet Picker is displayed within the new window.
- b. Scroll through the list or use the **Filter** field to find the portlet you want to add.
- c. Click **Add Portlet**. The portlet is added to the window.
5. Optional: To remove a portlet and its window, click the **Delete** icon in the title bar. The content is removed immediately without a warning prompt.
6. Optional: To create wires between portlets so they can share information and updates, click **Show Wires**. Before working with wires, make sure that you have enough information about the events that a portlet supports.
7. Click **Page settings**. The page settings are displayed.
8. Optional: Make changes to the page's settings as required.
 - a. Click the **General** tab.
 - b. In the **Page name** field, provide a descriptive name for the page and in the **Page location** field indicate where you want the page to be displayed in the navigation pane. Consider the content on the page and how users will find that content by looking for the page name in the navigation pane.
 - c. Use the **Navigation visibility** list to indicate whether or not you want the page to be listed in the navigation pane.
 - d. From the **Page persistence** list, make one of the following selections:
 - Client side (default setting) - This setting preserves any changes that the user makes on the page when the user navigates away from the page. Changes include not only form data, but any state changes to portlets, for example, opening edit mode, switching to another panel in the portlet, or minimizing a portlet. Page data and page state are maintained on the client side until the user closes the page or logs out of the console.
 - None
 - Server side - This setting maintains unsubmitted or unsaved form data from a page when the user navigates away from the page. The data is saved on the server and fetched when the user returns to the page. Unsaved data is saved until the user closes the page or logs out of the console.

Note: The Server side setting only applies to forms on a page. Any user interaction outside of a form is not maintained.

- e. Use the **Page tasking** radio buttons to indicate whether multiple instances of the page can be launched.
- f. In the **Component direction** drop-down list, you can accept the **Default** setting to allow the component direction to be governed at console level or select one of the other settings to indicate whether you want to display page components from left-to-right or from right-to-left. If you select a setting other than **Default**, it will override any component direction setting that may be set at console or browser level.
- g. In the **Text direction** drop-down list, you can accept the **Default** setting to allow the text direction to be governed at console level or select Left-to-Right or from Right-to-Left to indicate the direction that you want the page text to display. You can also select **Contextual Input** so that for pages that include text entry fields, the direction of text is dependent on the language used to enter data. If you select a setting other than **Default**, it will override any text direction setting that may be set at console or browser level.

9. Optional: Click the **Roles** tab to update the list of roles with permissions to the page and their access level. A list of all roles with access to the page is displayed.

Option	Description
To remove access for a role	Select a role and click Remove . The role is removed immediately from the access list without a warning prompt.
To add access for a role	Click Add . Select one or more of the roles displayed and click OK . The roles you added are included to the list.
To change the access level for a role	Select one of the options under Access Level for the role.

Attention: Make sure that the roles with access to a page also have access to the portlets that are on the page.

10. Optional: Click the **View Membership** tab to update the list of views that include this page.

Option	Description
To add this page to a view	Click Add and select one or more views.
To remove this page from a view	Select one or more views in the list and click Remove .

11. If you accessed the Page settings window and made changes, click **Save** to commit your changes and return to the main edit page window.
12. When you are satisfied with your updates, click **Save** to commit your changes.

Results

You are returned to the page with your changes displayed.

Arranging portlets using the drag-and-drop feature:

When editing a page, you can drag portlets to any window on the page. The portlet must already be placed in a window on the page, and the target window must already exist.

The target window can be an empty window or it can already contain a portlet.

- If you drag a portlet into an empty window, the original window becomes empty after the portlet has been moved.
 - If you drag a portlet into a window that already contains another portlet, the two portlets exchange windows.
1. Locate the mouse over the portlet title in the title bar. You cannot drop a portlet into another window by dragging from any other location in the portlet window or title bar. The portlet must be dragged using the title.
 2. Drop the portlet in the target window when the target window displays a blue, dotted outline around the frame. The outline is the only indication that the portlet can be dropped into this location.

Creating folders

Folders are used to group nodes in the console navigation. All folders that are created in the console have a resource type of “Custom”.

1. Click **Settings > Page Management** in the navigation pane. A Page Settings page is displayed.
2. Click **New Folder**. The properties panel for the new folder is displayed.
3. Complete the fields in the properties panel.
4. Click **Save** to save your changes and return to Page Management.

Results

The new folder is displayed in the summary table. The folder is also displayed in the navigation pane once you have added page content to it. Add other nodes to the folder by editing their location properties.

Editing the properties of a page, folder, or external URL

You can edit the properties of custom and system navigation nodes, which include pages, folders, and external URLs. Properties of a node include its display name and its location in the navigation. You can also indicate whether multiple or only single instances of a page node can be launched in the console.

When changes are made to a system node, the updated system node is saved as System - Customized. You cannot delete a system node. Instead, you can restore the system node, which deletes the custom copy of it.

You can perform the following tasks when you edit a node's properties.

- Define who can access a page or external URL and the level of access
- Determine which view should include the node. When the view is selected, the page, folder, or URL is included in the navigation pane for that view.
- Change the name that is displayed in the navigation pane for a node.
- Change the location of a node in the navigation pane. For example, you can group pages into folders.

Attention: You cannot create URLs in the console. Instead, URLs are created when an application is deployed to the console that includes the URL node in its descriptors.

1. Click **Settings > Page Management** in the navigation pane. Page Management is opened displaying console navigation nodes in a summary table.
2. Locate the node that you want to edit in the table provided. Use the filter in the table to type in the node name and quickly display it.
3. Click the link for the node provided in the **Name** column. The properties panel for the node is displayed.
4. Make your changes to the node's Page, Folder, and External URL properties.
5. Click **Save** when you have finished.

Results

The changes you made are reflected in the navigation pane.

Deleting custom pages and folders

You can delete only pages with the resource type of Custom. These are nodes created using the console.

System nodes that have been customized can be restored.

Attention: Before deleting a page or folder, consider whether any users are actively using the resource and any impacts this might have on services. If necessary, notify users in advance of any plans for changes that could affect their work.

1. Click **Settings > Page Management** in the console navigation. Page Management is opened displaying console navigation nodes in a summary table.
2. Locate the node that you want to delete in the table provided. Use the filter in the table to type in the node name and quickly display it.
3. Check the box in the **Select** column for the node. You can select more than one custom page or folder for deletion.
4. Click **Delete**. A message is displayed at the top prompting you to confirm the deletion.
5. Click **OK**.

Results

The page or folder is deleted and removed from the navigation pane.

Restoring system pages, folders, and external URLs

System nodes are always preserved with their original settings. After making changes to a system node, the changes are saved in a customized copy of the page, folder, or URL. When you restore a system node, the customized copy is deleted and the original system node is restored in its place.

To delete the customized copy and restore the system node, follow these steps.

1. Click **Settings > Page Management** in the console navigation. Page Management is opened displaying console navigation nodes in a summary table.
2. Locate the node that you want to edit in the table provided. Use the filter in the table to type in the node name and quickly display it.
3. Click the link for the node provided in the **Name** column. The properties panel for the node is displayed.
4. Scroll to the bottom of the panel and click **Restore**.
5. Click **OK** to save your changes.

Results

You are returned to the main panel of Page Management. The resource type of the node is displayed as System.

Creating startup pages

You can create startup pages, which are displayed after a user logs in, and assign them to users or user groups based on their role. You can also hide the links to other portlets and pages from the navigation.

Before you begin

Make sure that the user account you want to use has the `isc_admins` role assigned.

To create startup pages:

1. Optional: To create a new role, click **Users and Groups > Role Management** and follows the steps under “Creating roles” on page 83.
2. Optional: To create a new page:
 - a. Click **Settings > Page Management**
 - b. Follow the steps under “Creating pages” on page 29.
Add the necessary content to the page. For example, you can add AEL portlets to an IFrame portlet along with other content for the intended users.
 - c. To specify the role or roles required to access the page, click **Roles with access to this page**. Click **Add** and select the required roles.
3. Add the startup page to a view:
 - a. Click **Settings > View Management** and click **New**.
 - b. In the **View name** field, type a name for the view.
 - c. Select **Hide any open pages in the work area that are not part of this view**.
 - d. Click **Roles with Access to This View**.
 - e. On the Available Roles page, select the required roles and click **Add**.

Tip: To give Web GUI administrators access to the view, select the `ncw_admin` role.
 - f. Select the level of access for each assigned role.

Tip: Set editor-level access for the `ncw_admin` role.
 - g. Click **Pages in This View** and click **Add**.
 - h. On the Available Pages page, select the required pages and click **Add**.
 - i. Select **Select and Set all pages in this view to launch**.
 - j. Click **Save**.
4. Link the view and the role by creating a console preference profile:
 - a. Click **Settings > Console Preference Profiles** and click **New**.
 - b. In the **Preference profile name** field, type a name for the profile.
 - c. Select **Show navigation tree**.
 - d. To restrict the navigation options to only the pages specified in the view, click **Required view** and clear **All tasks**.
 - e. Clear **Core views**.
 - f. Click **Roles using this preference profile** and click **Add**.
 - g. On the Available Roles page, select the required roles and click **Add**.
 - h. Under **Default console view** select the view that you created in step 3.
 - i. Click **Save**.
5. To assign roles to users:
 - a. Click **Users & Groups > User Roles**.

- b. Complete any combination of the search fields to help locate the users.
 - c. Select how many users to display and click **Search**. A list of matching users appears in the grid.
 - d. Click the user ID of the user you want to assign roles to.
 - e. From the **Role(s)** list, select the roles to assign the user.
 - f. Click **Save**.
6. To assign roles to user groups:
 - a. Click **Users & Groups > Group Roles**.
 - b. Complete any combination of the search fields to help locate the groups.
 - c. Select how many groups to display and click **Search**. A list of groups appears in the grid.
 - d. Click the name of the group you want to assign roles to.
 - e. From the **Role(s)** list, select the roles to assign the user group.
 - f. Click **Save**.

Results

When a user belonging to a group that has the assigned role logs in, the selected startup page is automatically loaded. In the **View** list above the navigation, the user can switch between navigation displays: **All tasks** displays all the navigation options assigned to the role, and *viewname* displays all the options assigned to the view. If a user is assigned to multiple roles that have startup pages, all pages are automatically loaded after login.

Related tasks

“Administering users” on page 73

“Administering groups” on page 84

“Creating roles” on page 83

“Creating pages” on page 29

“Creating views” on page 38

“Creating preference profiles” on page 45

“Creating users” on page 75

“Creating groups” on page 87

Related information

“Administering pages” on page 28

“Administering console preference profiles” on page 45

“Administering views”

Administering views

Views are a defined set of tasks that are displayed in the console navigation pane. Views also can include one or more pages that are launched when the view is selected.

For example, if you find a set of tasks related to obtaining sales and cost reports from retail stores throughout a region, you could create a view called “Reports” that includes all of the pages associated with those tasks in the navigation. Each page, along with the folders that include them, would be added to the view. You could then set some of the most important pages to launch when the view is

selected. In this way, views can make your experience with the console more productive than sorting through all of the navigation tasks that are displayed by default.

If you have sufficient access, you can create your own custom views. You can only edit system views.

To access View Management in the console, click **Settings > View Management** in the navigation.

Field descriptions

This section describes the fields and controls in the main panel of View Management.

Select all icon

Selects all items displayed in the table for deletion. If you are displaying only a filtered set of items, only those items are selected. You can deselect specific items before actually deleting.

Deselect all icon

Deselects all items displayed in the table.

New Opens a panel for creating a new view.

Delete Immediately deletes all selected items in the list. Only Custom resource types can be deleted.

Filter Type in this field to quickly find an item in the table. This field is useful when there are a large number of items to look through.

Select Selects or deselects a single item in the table.

View Name

Displays the name of the view as it is shown in the **View** selection list in the banner. Click the name to edit the view.

Type Displays the type of view. The actions you can perform on a view depend upon its type.

Role Count

Displays the number of roles that have access to this view..

Page Count

Displays the number of pages that are available in the console when the view is selected.

Related tasks

“Creating startup pages” on page 35

Related information

“Administering roles” on page 80

Creating views

Views determine what pages are listed in the navigation pane as well as which pages are launched when the view is selected. All views that are created in the console have a resource type of Custom. This procedure walks you through the task of creating a view for testing purposes. After completing these steps, you can remove or edit this view for production use.

Before you begin

You should understand the Console layout before starting this task.

1. Click **Settings > View Management** in the navigation pane. The View Management page is displayed with the list of system and custom views in the console.
2. Click **New**. The properties panel for the new view is displayed.
3. Enter a descriptive name for the view. This name is displayed in the **View** selection list in the banner.
4. Expand the **Roles with Access to This View** section and click **Add**. The **Add Roles** panel is displayed with a list of available roles. For this task, add a role that can be used to test the view before adding access for other roles.
Attention: Granting access to the view does not grant access to the pages within the view.
5. Select your role in the table. You can use the filter to quickly find your role if the list of roles is very large.
6. Click **Add** after making your selection. You are returned to the view properties. The next step is to determine the pages that make up the view.
7. Expand the **Pages in This View** section and click **Add**. The **Add Pages** panel is displayed with a list of available pages.
8. Select several folders or pages in the list. Selecting a folder also selects all of the pages contained in that folder. You can individually deselect pages in a folder if necessary.
9. Click **Add** after making your selections. You are returned to the view properties.
10. Select the **Launch** option for two or three of the pages and select one of the launch pages as the default.
11. Click **Save** to save the new view and return to View Management.

Results

Select the new view from the **View** drop down list located above the navigation pane. Verify that all pages and folder that you selected are displayed in the navigation, that the pages selected to launch are available in the page bar, and that the default selection has focus in the work area.

Editing views

Views provide a limited set of nodes in the console navigation and optional set of startup pages to help users focus on their tasks. If you have sufficient authorization in the console, you can change the view name, navigation content, and access permissions for system and custom views. You can delete only custom views. Changes you make to a system view are saved as *System Customized*.

1. In the navigation pane, click **Settings > Views**. The View Management page is displayed with the list of system and custom views in the console.
2. Click the view name in the list displayed in View Management. This displays the view's properties.
3. Optional: Expand **Roles with Access to This View** to update the list of roles with permissions to the view and their access level. A list of all roles with access to the view is displayed.

Option	Description
To remove access for a role	Select a role and click Remove . The role is removed immediately from the access list without a warning prompt.
To add access for a role	Click Add . Select one or more of the roles displayed and click OK . The roles you added are included to the list.
To change the access level for a role	Select one of the options under Access Level for the role.

Note: Granting access to the view does not grant access to the pages within the view.

4. Optional: Expand **Pages in This View** to change which pages are displayed in the navigation when the view is selected.

Option	Description
Add a page to the view	Click Add to add a page to the view.
Remove a page from the view	Select the page in the Select column and click Remove . You can select multiple pages to remove.
Change the launch options for a page	Select Launch for each page that should be opened when the view is selected. Only one page can be in focus (current) when the view is selected. When multiple pages are set to launch, set the current page in the Default column.

5. Click **Save** to save your changes and return to the main view panel.

Results

For customized versions of a system view, you can retrieve the system view settings by editing the system customized view and clicking **Restore**. The “system customized” version of the view is removed and replaced by the original system view.

Deleting custom views

You can delete only views with the resource type of Custom. These are views created using the console.

Customized versions of system views can be restored. Restoring a System Customized view deletes the custom copy and replaces it with the original system view.

Attention: Before deleting a view, consider whether any users are actively using the view and any impacts this might have on services. If necessary, notify users in advance of any plans for changes that could affect their work.

Follow these steps to delete a custom view.

1. Click **Settings > View Management** in the console navigation. The View Management page is displayed with the list of system and custom views in the console.
2. Select the custom view that you want to delete. You can select more than one custom view.
3. Click **Delete**. A message is displayed at the top prompting you to confirm the deletion.
4. Click **OK**.

Results

The custom view is removed from the view list.

Administering portlets

Portlets are web applications that display information or provide a service in a console page. You can only work with portlets that have been deployed to the console. Use Portlet Management to create, edit, and delete portlet from a page.

To access Portlet Management in the console, click **Settings > Portlet Management** in the navigation. The main panel displays a list of all of the portlets in the console. Within the list, the portlets are grouped into the pages and folders as they are located in the console navigation. The group Uncategorized portlets indicates portlets that are not placed on a page within the navigation. To place a portlet on a page, you have to edit the page.

A delete icon to the right of a portlet indicates that it is a copy.

Field descriptions

This section describes the fields and controls in the main panel of Portlet Management.

Select all icon

Selects all items displayed in the table for deletion. If you are displaying only a filtered set of items, only those items are selected. You can deselect specific items before actually deleting.

Deselect all icon

Deselects all items displayed in the table.

Copy Creates a copy of all selected portlets and updates the table. Copied portlets are placed in the Uncategorized portlets folder with the name *Copy of Original Portlet Name*.

Filter Type in this field to quickly find an item in the table. This field is useful when there are a large number of items to look through.

Select Selects or deselects a single item in the table.

Portlet Name

Displays the title of the portlet as it is shown on the page.

Unique Name

Displays the name used by the console to uniquely identify this portlet.

Delete For portlet copies only, shows the **Delete** icon to remove the portlet from the system. Clicking this icon removes the portlet copy immediately without a warning prompt.

Related information

“Administering pages” on page 28

“Administering roles” on page 80

Creating portlets

You can create a new copy of an existing portlet. You can create many different portlet copies, or *portlet entities*, of a single portlet, each entity with a different name. The portlet must already be installed to the console for you to create a copy of it.

Note: If you are creating a new copy of a portlet, add your own role to the portlet access list so that you can view the portlet when it is placed on a page.

Follow these steps to create a copy of a single portlet.

1. Click **Settings > Portlet Management** in the navigation pane. A list of all of the console portlets is displayed in a scrollable table.
2. Browse through the list or use the **Filter** field to locate the portlet you want to copy. To use the filter field, start typing the portlet name. The list is reduced to portlets whose names match the characters you enter.
3. Check the box next to the portlet and click **Copy**. The portlet properties panel is opened. The **Portlet Entity Title** field displays the original portlet's title prefixed with Copy of.
4. Click the new portlet name.
5. Enter a new, descriptive name for **Portlet Entity Title**. In most circumstances, the entity title becomes the display name.
6. Click **Roles with Access to This Portlet**. The list of roles with access is based on the access list of the original portlet.
7. Use the **Add** and **Remove** buttons to update the list. For each role, verify their access level is set correctly.
Attention: You must add your own role to this list to access the portlet copy on any page it is placed.
8. Click **Save** to save your changes and return to Portlet Management.

Results

Now that you have finished creating your portlet copy, use Page Management to place the portlet on a page.

To create copies of multiple portlets, follow these steps, but select multiple portlets from the list before you click **Copy**. This adds each new portlet copy to the list under Uncategorized portlets. Click each portlet in the list to change the title.

Editing portlets

Portlets provide content on a console page, for example, viewing system information or submitting reports. If you have sufficient authorization in the console, you can change access permissions to a portlet. For copies of portlets that have been created, you can also change the display name of the portlet. If you want to place a portlet on a page, you have to edit the page and select the portlet from the displayed list.

1. Click **Settings > Portlet Management** in the console navigation. A list of all of the console portlets is displayed in a scrollable table.
2. Browse through the list or use the **Filter** field to locate the portlet you want to edit. To use the **Filter** field, start typing the portlet name. The list is reduced to portlets whose names match the characters you enter.
3. Click the name of the portlet that you want to edit. The portlet properties are displayed.
4. Optional: Enter a descriptive name for the portlet. The portlet name can be changed only if this is a copy of a portlet.
5. Optional: Expand **Roles with Access to This Portlet** to update the list of roles with permissions to the portlet and their access level. A list of all roles with access to the portlet is displayed.

Option	Description
To remove access for a role	Select a role and click Remove . The role is removed immediately from the access list without a warning prompt.
To add access for a role	Click Add . Select one or more of the roles displayed and click OK . The roles you added are included to the list.
To change the access level for a role	Select one of the options under Access Level for the role.

6. Optional: Use the **Component direction** and **Text direction** fields to set the direction to display portlet content and text. For both portlet content and text, the **Default** option allows the portlet to inherit the display direction that is set at page level. You can set the text and content direction at portlet level to either left-to-right or right-to-left. Additionally, in the **Text direction** list, you can also select **Contextual Input** so that for portlets that include text entry fields, the direction of text is dependent on the language used to enter data.
7. Click **Save** to save your changes and return to the main portlet panel.

Editing portlet shared settings

Some portlets include an *Edit Shared Settings* mode that allows users with “Editor” access level to configure common settings for other users of the portlet. Once shared settings are configured, users with “Privileged User” level of access can change these values for their own personal use of the portlet. Default settings cannot be changed by users with “User” level of access. Follow these steps to set the shared settings for a portlet.

Before you begin

You must have “Editor” access to the portlet to perform this task.

1. Navigate to the page where the portlet is located.
2. Click the **Edit options** icon in the portlet title bar. Two options are displayed: **Personalize** and **Edit Shared Settings**.

Attention: If this icon is not available in the portlet title bar, then either the portlet does not support *Edit Shared Settings* mode, or you do not have “Editor” access for the portlet.

3. Select **Edit Shared Settings**. The portlet displays shared settings that can be changed.
4. Make any changes to the settings and submit them when you are finished. The portlet might provide a **Save**, **OK**, or **Submit** button. Once you have submitted your changes, you should be returned to the main panel for the portlet. If not, click the **Back** icon in the title bar.

Results

The shared settings for using this portlet are saved. If the portlet is located on more than one page, the updated settings will be observed on the other pages as well.

What to do next

The updated settings configuration only affect settings that have not been personalized by users. To verify that the a user's preferences have been preserved, log in with a test user name and verify that the shared settings are set as intended.

Deleting portlets

You can use the console to delete only a copy of a portlet. To remove the original portlet, the console administrator must undeploy the console module application to which the portlet belongs.

1. Click **Settings > Portlets** in the console navigation. A list of all of the console portlets is displayed in a scrollable table.
2. Browse through the list or use the Filter field to locate the portlet you want to remove. To use the filter field, start typing the portlet name. The list is reduced to portlets whose names match the characters you entered.
3. Click the **Delete** icon.

Results

The portlet is removed immediately without a warning prompt.

Creating and configuring a Web widget

Administrators can configure the Tivoli Integrated PortalIntegrated Solutions Console web widget portlet to display a discrete Web page.

To create a Web widget portlet to display a discrete Web page:

1. Create a new page and in the Choose a Portlet panel, select **Web widget** and click **OK** to display the Web widget page.
2. To apply general settings for the Web widget, click the the **Edit options** icon. A Web widget settings page is displayed.
3. In the **Widget Title** field provide a brief descriptive name. When selecting a title, consider the purpose of the Web widget, its home page, and that the title is used in the navigation pane to access the Web widget.
4. In the **Home URL** field provide a valid Web address. This the Web address of the page that displays by default when a user accesses the Web widget from the navigation pane.

5. Optional: In the **Help URL** field, provide a relative or absolute URL to a custom help page HTML topic to replace the default help topic that ships with the Web widget.
6. Optional: In the **HTML iFrame name** field, provide a unique HTML iFrame name. The entry in this field serves to uniquely identify the Web widget to allow its content to be dynamically updated.
7. Check the **Show a browser control toolbar** to provide users with a Web navigation toolbar, that is, standard Web navigation buttons and a Web address entry field.
8. To allow users to personalize their Web widget settings , check the relevant check box. By default, the following check boxes are cleared:
 - **Widget title**
 - **Home page**
 - **Help page**
 - **Browser control bar**
9. Click **Save** to commit your changes, or **Restore Default Settings** to reset the form.

Editing shared settings for a Web widget:

Administrators can set shared settings to provide a common experience for users of a Web widget.

If you have not configured any shared settings for a Web widget, users can by default enter a complete Web address in the field provided and browse Web pages. To edit shared settings:

1. In the title bar of the Web widget, click the **Edit options** icon to display an edit shared settings page.
2. Optional: In the **Widget Title** field provide a brief descriptive name. When selecting a title, consider the purpose of the Web widget, its home page, and that the title is used in the navigation pane to access the Web widget.
3. Optional: In the **Home URL** field provide a valid Web address. This the Web address of the page that displays by default when a user accesses the Web widget from the navigation pane.
4. Optional: In the **Help URL** field, provide a relative or absolute URL to a custom help page HTML topic to replace the help topic that is currently associated with the Web widget.
5. Optional: In the **HTML iFrame name** field, provide a unique HTML iFrame name. The entry in this field serves to uniquely identify the Web widget to allow its content to be dynamically updated.
6. Check the **Show a browser control toolbar** to provide users with a Web navigation toolbar, that is, standard Web navigation buttons and a Web address entry field.
7. To allow users to personalize their Web widget settings , check the relevant check box. By default, the following check boxes are cleared:
 - **Widget title**
 - **Home page**
 - **Help page**
 - **Browser control bar**
8. Click **Save** to commit your changes, or **Restore Default Settings** to reset the form.

Administering console preference profiles

Preference profiles are a collection of console behavior preferences for using the console that are created by the console administrator. These preferences include the visibility of the navigation tree, contents of the view selection list, and the default view. Assign preference profiles to roles to manage how the navigation area and view selections are displayed to the users in the role.

Attention: Each role is limited to one preference profile.

Field descriptions

This section describes the fields and controls in the main panel of Console Preference Profiles.

Select all icon

Selects all items displayed in the table for deletion. If you are displaying only a filtered set of items, only those items are selected. You can deselect specific items before actually deleting.

Deselect all icon

Deselects all items displayed in the table.

New Opens a panel for creating a new preference profile.

Delete Immediately deletes all selected items in the list. Only Custom resource types can be deleted.

Filter Type in this field to quickly find an item in the table. This field is useful when there are a large number of items to look through.

Select Selects or deselects a single item in the table.

Profile Name

Indicates the name of the profile. You can sort the list of names by clicking the column heading.

Role Count

Indicates the number of roles assigned to a preference profile. Each role is limited to one preference profile. However, multiple roles can be assigned to any single preference profile.

Related tasks

“Creating startup pages” on page 35

Creating preference profiles

Preference profiles are a collection of console behavior preferences for using the console that are created by the console administrator. Follow these steps to create a preference profile and assign it to a role.

1. Click **Settings > Console Preference Profiles** in the console navigation. The Console Preference Profiles page is displayed with the list of preference profiles that have already been created in the console.
2. Click **New**. The properties panel for the new preference profile is displayed.
3. Enter a descriptive name for the preference profile. Consider how the name reflects the roles that have been assigned to it or the console settings that are defined.
4. Indicate whether the navigation tree should be hidden. This might be preferable when the user has few pages to access and display space in the console is better reserved for page content.

5. Optional: Use the Console Bidirection Options to set the direction to display console content and text. The default option lets the browser dictate the text and content direction, for example, for Arabic and Hebrew, the text is displayed right-to-left, whereas for other languages it is displayed left-to-right. Alternatively, you can decide to set the text and content direction to either left-to-right or right-to-left. In the **Text direction** list, you can also select **Contextual Input** so that for portlets that include text entry fields, the direction of text is dependent on the language used to enter data.
6. Select which view options should be available for users in the role.
7. Expand the section **Roles Using this Preference Profile**.
8. Click **Add** and select one or more roles to use this preference profile. When assigning roles, you might notice some roles missing from the list. This means they are assigned to another preference profile. The role must be removed from the other profile before it can be assigned to this one.
9. Select the default console view for this preference profile. The default view is the one that is selected when users in this role log in to the console. This field is enabled when at least one role has been added for this preference profile.
10. Click **Save** to save your changes and return to Console Preference Profiles.

Results

The new preference profile is created and listed on the main panel for Console Preference Profiles.

Editing console preference profiles

Preference profiles are a collection of console behavior preferences for using the console that are created by the console administrator. Follow these steps to change the properties or roles assigned to a preference profile.

1. In the navigation pane, click **Settings > Console Preference Profiles**. The Console Preference Profiles page is displayed with the list of preference profiles that have already been created in the console.
2. Click the name of the preference profile that you want to edit. The properties panel for the preference profile is displayed.
3. Optional: Enter a descriptive name for the preference profile. Consider how the name reflects the roles that have been assigned to it or the console settings that are defined.
4. Optional: Indicate whether the navigation tree should be hidden. This might be preferable when the user has few pages to access and display space in the console is better reserved for page content.
5. Optional: Use the Console Bidirection Options to set the direction to display console content and text. The default option lets the browser dictate the text and content direction, for example, for Arabic and Hebrew, the text is displayed right-to-left, whereas for other languages it is displayed left-to-right. Alternatively, you can decide to set the text and content direction to either left-to-right or right-to-left. In the **Text direction** list, you can also select **Contextual Input** so that for portlets that include text entry fields, the direction of text is dependent on the language used to enter data.
6. Optional: Select which view options should be available for users in the role.
7. Expand the section **Roles Using this Preference Profile**.

Option	Description
To add roles	Click Add and select one or more roles to add to the list. Click OK when you have made all of your selections. Note: If a role is not listed, it likely means that it has been assigned to another preference profile.
To remove roles	Select one of more roles in the list and click Remove . Be certain of your selections. When you delete, there is no warning prompt and the action cannot be undone.
To assign a default view	Select from the Default console view section to the side of the role list.

8. Click **Save** to save your changes.

Results

The preference profile is updated and you are returned to the main panel for Console Preference Profiles.

Deleting console preference profiles

Preference profiles are a collection of console behavior preferences for using the console that are created by the console administrator. Follow these steps to delete a preference profile.

1. Click **Settings > Console Preference Profiles** in the navigation pane. The Console Preference Profiles page is displayed with the list of preference profiles that have already been created in the console.
2. Locate the preference profile that you want to delete in the table provided. You can use the filter in the table to type in the preference profile name and quickly display it.
3. In the **Select** column select one or more preference profiles.
4. Click **Delete**. A message is displayed at the top prompting you to confirm the deletion.
5. Click **OK**.

Results

The preference profile is removed.

Exporting and importing Tivoli Integrated Portal data

You can export customized configuration data from an existing Tivoli Netcool/OMNIBus Web GUI installation to another by exporting the data and subsequently importing the exported data.

Exporting and importing customized settings can be done at the command line through the `tipcli.bat|.sh Export` and `tipcli.bat|.sh Import` commands.

Note: The `tipcli.bat|.sh Export` and `tipcli.bat|.sh Import` commands are case sensitive. Also, if you make a typing error, that is, if you type a parameter incorrectly, or use the incorrect case, then the commands runs as if no parameters were specified and no warning message is displayed.

You can export and import the following elements:

- Custom pages and customized system page elements, with the exception of core and system pages, including:
 - Page name and layout.
 - Portlet entities.

Note: Copies of a portlet entity are not exported; either through the console Export Wizard or through the `tipcli.bat|.sh Export` command.

- View profiles.
 - Events and wires.
 - Access permissions.
 - Navigation structure.
- Custom views (or customized system views).

Note: You can also export pages associated with a view if the `exportpageinview` parameter is set to `true`.

- Custom roles, including:
 - Role name, creation date, and update date.
 - Role mapping information in relation to users and groups.
 - Associated role preference, that is, the relevant console preference profile.
- Console properties and customization properties, including:
 - Transformations.
 - Themes and images.
 - Bundles.

In a load balanced environment the import operation migrates imported elements across all the computers in the pool, with following conditions:

- All the required applications (WAR files) must be deployed on all computers in the pool.
- The load balanced pool configuration must be locked during the import operation.
- The import operation must be ran on one of the nodes in the pool.
- You must provide the load balancing manager an updated file list to update the load balancing scope. The migration tool plugin provides the file list.
- The load balanced pool configuration, can then be unlocked.
- The import of transformations in a load balanced environment is not supported. Transformations must be imported to each node independently.

The `haSupport` command controls this aspect of the import operation:

- If it is set to `True`, then only load balancing information is imported, that is, no transformation data.
- If it is set to `False`, then only transformation data is imported, that is, no load balancing data.
- If it is set to `Both`, then transformation data and load balancing data is imported.

Basic export commands

You can export pages, views and profile preferences using the basic export commands.

Exporting pages in simplified mode:

By using the `ExportPage` command you can export specific pages without having to provide additional qualifying parameters.

Before you begin

Ensure that the Tivoli Integrated Portal Server is running.

To export specific pages in simplified mode for an instance of Tivoli Integrated Portal:

1. At the command line change to: `tip_home_dir/profiles/TIPProfile/bin`.
2. To return a list of customized pages that can be exported, run the following command:

- **Windows** `tip_home_dir\profiles\TIPProfile\bin\tipcli.bat ListPages --customizePages true`
- **UNIX** **Linux** `tip_home_dir/profiles/TIPProfile/bin/tipcli.sh ListPages --customizePages true`

Note: The page ID is the last element of the returned records, for example, the page ID for the following record is BIXRjLkKYngNsRavnu0fYpx1279539744250:

```
com.ibm.isclite.global.custom.module-SPSVS-  
com.ibm.isclite.admin.PortletPicker.navigationElement.pagelayoutA  
.modified.BIXRjLkKYngNsRavnu0fYpx1279539744250
```

3. Review the list of returned page records and take note of the page IDs for the pages that you want to export.
4. To export specific pages, run the following command:

- **Windows** `tip_home_dir\profiles\TIPProfile\bin\tipcli.bat ExportPage --uniqueName pageID_1,pageID_2,pageID_3`
- **UNIX** **Linux** `tip_home_dir/profiles/TIPProfile/bin/tipcli.sh ExportPage --uniqueName pageID_1,pageID_2,pageID_3`

Note: The file `portletEntities.xml` is always exported, even if you specify `NONE` as an argument to the `uniqueName` parameter.

Results

When the command completes, a `Data.zip` file is created in `tip_home_dir/profiles/TIPProfile/output/`.

What to do next

Locate `tip_home_dir/profiles/TIPProfile/output/Data.zip` and copy it to the computer where you intend to apply the exported customization data.

Exporting views in simplified mode:

By using the ExportView command you can export specific views without having to provide additional qualifying parameters.

Before you begin

Ensure that the Tivoli Integrated Portal Server is running.

To export specific views in simplified mode for an instance of Tivoli Integrated Portal:

1. At the command line change to: *tip_home_dir*/profiles/TIPProfile/bin.
2. Optional: To return a list of customized views that can be exported, run the following command:
 - **Windows** `tip_home_dir\profiles\TIPProfile\bin\tipcli.bat ListViews`
 - **UNIX** **Linux** `tip_home_dir/profiles/TIPProfile/bin/tipcli.sh ListViews`
3. Review the list of returned view records and take note of the view IDs for the views that you want to export.
4. To export specific views, run the following command:
 - **Windows** `tip_home_dir\profiles\TIPProfile\bin\tipcli.bat ExportView --uniqueName viewID_1, viewID_2, viewID_3`
 - **UNIX** **Linux** `tip_home_dir/profiles/TIPProfile/bin/tipcli.sh ExportView --uniqueName viewID_1, viewID_2, viewID_3`

Note: The file portletEntities.xml is always exported, even if you specify NONE as an argument to the uniqueName parameter.

Results

When the command completes, a Data.zip file is created in *tip_home_dir*/profiles/TIPProfile/output/.

What to do next

Locate *tip_home_dir*/profiles/TIPProfile/output/Data.zip and copy it to the computer where you intend to apply the exported customization data.

Exporting console preference profiles in simplified mode:

By using the ExportProfile command you can export console preference profiles without having to provide additional qualifying parameters.

Before you begin

Ensure that the Tivoli Integrated Portal Server is running.

To export console preference profiles in simplified mode:

1. At the command line change to: *tip_home_dir*/profiles/TIPProfile/bin.
2. Optional: To return a list of console preference profiles that can be exported:
 - **Windows** `tip_home_dir\profiles\TIPProfile\bin\tipcli.bat ListPreferenceProfiles`

- **UNIX** **Linux** `tip_home_dir/profiles/TIPProfile/bin/tipcli.sh`
ListPreferenceProfiles
- 3. Review the list of returned records and take note of the unique names for the console preference profiles that you want to export.
- 4. To export specific console preference profiles, run the following command:
 - **Windows** `tip_home_dir\profiles\TIPProfile\bin\tipcli.bat`
ExportProfile --uniqueName *profile_ID1,profile_ID2,profile_ID3*
 - **UNIX** **Linux** `tip_home_dir/profiles/TIPProfile/bin/tipcli.sh`
ExportProfile --uniqueName *profile_ID1,profile_ID2,profile_ID3*

Note: The file `portletEntities.xml` is always exported, even if you specify `NONE` as an argument to the `uniqueName` parameter.

Results

When the command completes, a `Data.zip` file is created in `tip_home_dir/profiles/TIPProfile/output/`.

What to do next

Locate `tip_home_dir/profiles/TIPProfile/output/Data.zip` and copy it to the computer where you intend to apply the exported customization data.

Advanced export commands

You can use the more advanced `tipcli` Export commands and apply a number of parameters to define which items you want to include and exclude in relation to the export operation.

Exporting all customization data:

You can export all customization data for an instance of Tivoli Integrated Portal in one command.

Before you begin

Ensure that the Tivoli Integrated Portal Server is running.

To export all customization data for an instance of Tivoli Integrated Portal:

1. At the command line change to: `tip_home_dir/profiles/TIPProfile/bin`.
2. Optional: To return a list of plugins that will be run during the export operation, run the following command:
 - **Windows** `tip_home_dir\profiles\TIPProfile\bin\tipcli.bat`
ListExportPlugins
 - **UNIX** **Linux** `tip_home_dir/profiles/TIPProfile/bin/tipcli.bat`
ListExportPlugins
3. To export all customization data, run the following command:
 - **Windows** `tip_home_dir\profiles\TIPProfile\bin\tipcli.bat` Export
--username *tipadmin_user_name* --password *tipadmin_password*
 - **UNIX** **Linux** `tip_home_dir/profiles/TIPProfile/bin/tipcli.sh`
Export --username *tipadmin_user_name* --password *tipadmin_password*

Results

When the Export command completes, a Data.zip file is created in *tip_home_dir/profiles/TIPProfile/output/*.

Note:

Refer to the links at the end of the page to view details of customs parameters that can be applied to the Export command.

What to do next

Locate *tip_home_dir/profiles/TIPProfile/output/Data.zip* and copy it to the computer where you intend to apply the exported customization data.

Related reference

“Export tipcli commands” on page 58

Exporting using a properties file:

You can specify your export requirements in properties file instead of specifying your requirements using separate parameters at the command line.

Before you begin

By default, the tipcli command uses the *tip_home_dir/TIPProfile/etc/tipcli.properties* file unless this behavior is overridden by the specifying a discrete settings file using the *settingFile* parameter.

Ensure that the Tivoli Integrated Portal Server is running.

To export customization data using a properties file:

1. Create a properties file that specifies the data that you want to export and save it as *export-settings.properties* in a known location.

Below is example content for an export properties file:

```
import.includePlugins=ImportPagePlugin
export.includePlugins=ExportPagePlugin
import.backupDir=c:/tmp/bkups
export.exportFile=c:/tmp/extest.zip
import.importFile=c:/tmp/extest.zip
username=tip_admin_user
password=tip_admin_password
import.hasSupport=true
```

Note: Some parameters are import or export specific. Import specific parameters should be prefixed by *import.* and export specific parameters should be prefixed by *export.* For example, *import.backupDir=c:/tmp/bkups*.

2. At the command line change to: *tip_home_dir/profiles/TIPProfile/bin*.
3. To export customization data based on the contents of a specific properties file, run the following command:

- **Windows** `tip_home_dir\profiles\TIPProfile\bin\tipcli.bat Export --username tipadmin_user_name --password tipadmin_password --settingFile export_properties_file`
- **UNIX** **Linux** `tip_home_dir/profiles/TIPProfile/bin/tipcli.sh Export --username tipadmin_user_name --password tipadmin_password --settingFile export_properties_file`

Where:

export_properties_file

An argument to the `settingFile` parameter that provides the location and name of the export properties file, for example, `C:\\tmp\\export.properties`.

Note: Windows On systems running Windows you must use double backslashes characters (\\) when specifying the path to your settings file.

Note:

If there is a conflict between settings specified in the properties file and parameters provided at the command line, then the command line parameters take precedence.

Results

When the Export command completes, a `extest.zip` file is created in the root temporary directory, for example on Windows systems the file is saved in `c:\\tmp`.

What to do next

Locate `extest.zip` and copy it to the computer where you intend to apply the exported customization data.

Exporting specific pages:

When exporting Tivoli Integrated Portal data, you can specify that you want to export particular pages.

Before you begin

Ensure that the Tivoli Integrated Portal Server is running.

To export specific pages for an instance of Tivoli Integrated Portal:

1. At the command line change to: `tip_home_dir/profiles/TIPProfile/bin`.
2. To return a list of customized pages that can be exported, run the following command:

- Windows `tip_home_dir\\profiles\\TIPProfile\\bin\\tipcli.bat ListPages --customizePages true`
- UNIX Linux `tip_home_dir/profiles/TIPProfile/bin/tipcli.sh ListPages --customizePages true`

Note: The page ID is the last element of the returned records, for example, the page ID for the following record is `BIXRjLkKYngNsRavnu0fYpx1279539744250`:

```
com.ibm.isclite.global.custom.module-SPSVS-  
com.ibm.isclite.admin.PortletPicker.navigationElement.pagelayoutA  
.modified.BIXRjLkKYngNsRavnu0fYpx1279539744250
```

3. Review the list of returned page records and take note of the page IDs for the pages that you want to export.
4. To export specified pages, run the following command:

- **Windows** `tip_home_dir\profiles\TIPProfile\bin\tipcli.bat Export --username tipadmin_user_name --password tipadmin_password --pages pageID_1, pageID_2, pageID_3`
- **UNIX** **Linux** `tip_home_dir/profiles/TIPProfile/bin/tipcli.sh Export --username tipadmin_user_name --password tipadmin_password --pages pageID_1, pageID_2, pageID_3`

Results

When the command completes, a Data.zip file is created in `tip_home_dir/profiles/TIPProfile/output/`.

What to do next

Locate `tip_home_dir/profiles/TIPProfile/output/Data.zip` and copy it to the computer where you intend to apply the exported customization data.

Exporting specific views:

When exporting Tivoli Integrated Portal data, you can specify that you want to export particular views.

Before you begin

Ensure that the Tivoli Integrated Portal Server is running.

To export specific views for an instance of Tivoli Integrated Portal:

1. At the command line change to: `tip_home_dir/profiles/TIPProfile/bin`.
2. Optional: To return a list of customized views that can be exported, run the following command:
 - **Windows** `tip_home_dir\profiles\TIPProfile\bin\tipcli.bat ListViews`
 - **UNIX** **Linux** `tip_home_dir/profiles/TIPProfile/bin/tipcli.sh ListViews`
3. Review the list of returned view records and take note of the view IDs for the views that you want to export.
4. To export specific views, run the following command:
 - **Windows** `tip_home_dir\profiles\TIPProfile\bin\tipcli.bat Export --username tipadmin_user_name --password tipadmin_password --views viewID_1,viewID_2,viewID_3 --exportpageinviews [true|false]`
 - **UNIX** **Linux** `tip_home_dir/profiles/TIPProfile/bin/tipcli.sh Export --username tipadmin_user_name --password tipadmin_password --views viewID_1,viewID_2,viewID_3 --exportpageinviews [true|false]`

Where:

exportpageinviews

An optional parameter, when set to true ensures that you also export pages associated with the views that you have specified.

Note: Whether the optional parameter `exportpageinviews` is set to true or false, if a view has a default node in the navigation pane associated with it, then the page associated with the node is always exported. This is also true, even if you specify `NONE` as the argument to the `--pages` parameter.

Results

When the command completes, a `Data.zip` file is created in `tip_home_dir/profiles/TIPProfile/output/`.

What to do next

Locate `tip_home_dir/profiles/TIPProfile/output/Data.zip` and copy it to the computer where you intend to apply the exported customization data.

Rules for exporting:

When exporting customized configuration data, it is important to know the rules governing the export function and the options available to you.

The following rules apply when exporting customized configuration data from a Tivoli Integrated Portal environment:

Rules and options for pages

Rule

1. You can export a particular page by page ID or choose to export all pages.
2. You can export pages associated with a particular view.
3. You can export pages that are associated with a particular portlet from a particular WAR.
4. If a page contains multiple portlets, but only some from a specified WAR, then all elements of the page are exported.
5. Pages that are targets of a wire for a specified page are exported.
6. The default export scope is `All` if you do not define pages to be exported under rule 2 and rule 3.
7. The default export scope is `NONE` if you define pages to be exported under rule 2 and rule 3.

Rules and options for views

1. You can export a particular view by view ID or choose to export all views.
2. You can optionally export all views that contains a specified page.
3. The default export scope is `All`.
4. You can optionally export all pages associated with the views that you want to export.
5. If an view has a default node in the navigation pane associated with it, then that page is automatically exported with the view.

Rules and options for custom roles and role preferences (console preference profiles)

1. You can export a particular role by role ID or choose to export all roles.
2. You can export a custom role and role preference that is associated with a specified page or view.
3. The default export scope is set to `All`, unless the `includeEntitiesFromApps` parameter has been specified for a page or view, whereby it is then set to `REQUIRED`.
4. If a console preference profile has a custom view as its default view, then that view is automatically exported. If the exported view has a

default node in the navigation pane, then the associated page is automatically exported with the view.

Rules and options for user preferences

1. You can export user preferences by user ID or choose to export preferences for all users.
2. The default export scope is set to All, unless the `includeEntitiesFromApps` parameter has been specified for a page or view, whereby it is then set to REQUIRED.

Rules and options for console properties and customization properties

All of console properties and customization properties are exported.

Rules and options for transformations

All transformations are exported.

Import commands

You can use the `tipcli` Import commands and apply a number of parameters to define which items you want to include and exclude in relation to the import operation.

Importing previously exported data:

You can import data that was exported from another instance of Tivoli Integrated Portal.

Before you begin

Ensure that the Tivoli Integrated Portal Server is running.

Ensure that you have run the export operation on an originating instance of the Tivoli Integrated Portal Server and that you have copy the output file (`data.zip`) to the following directory on the other instance:

`tip_home_dir/profiles/TIPProfile/output`

To import data from a `data.zip` that was exported from another instance Tivoli Integrated Portal Server:

1. At the command line change to: `tip_home_dir/profiles/TIPProfile/bin`.
2. Optional: To return a list of plugins that will be run during the import operation, run the following command:
 - **Windows** `tip_home_dir\profiles\TIPProfile\bin\tipcli.bat ListImportPlugins`
 - **UNIX** **Linux** `tip_home_dir/profiles/TIPProfile/bin/tipcli.bat ListImportPlugins`
3. To import the customization data, run the following command:
 - **Windows** `tip_home_dir\profiles\TIPProfile\bin\tipcli.bat Import --username tipadmin_user_name --password tipadmin_password`
 - **UNIX** **Linux** `tip_home_dir/profiles/TIPProfile/bin/tipcli.sh Import --username tipadmin_user_name --password tipadmin_password`

Results

When the Import command completes, the imported data is merged with the existing Tivoli Integrated Portal environment.

Related reference

“Import tipcli commands” on page 61

Rolling back imports:

After you import data you can rollback your configuration to the pre-import state provided you have made no changes to the environment.

Before you begin

If you have performed multiple imports, you can also consecutively rollback individual imports. In all cases, you must have not had made changes to the environment.

Ensure that the Tivoli Integrated Portal Server is running.

To roll back imports for a Tivoli Integrated Portal environment:

1. At the command line change to: *tip_home_dir/profiles/TIPProfile/bin*.
2. To rollback an import, run the following command:

- **Windows** `tip_home_dir\profiles\TIPProfile\bin\tipcli.bat Import --rollback ALL`
- **UNIX** **Linux** `tip_home_dir/profiles/TIPProfile/bin/tipcli.sh Import --rollback ALL`

Once the command completes successfully, the Tivoli Integrated Portal environment is restored to the state that prevailed before the latest import operation was performed.

3. Optional: If you performed multiple imports and you want to roll back more than the most recent import operation, you can re-run the `tipcli.bat Import --rollback ALL` command. You can re-run the rollback command multiple times to consecutively roll back a number of import operations.

When you re-run the rollback command a second or subsequent time, the Tivoli Integrated Portal environment is restored to the state that prevailed prior the settings for that particular import operation being applied.

Rules for importing:

When importing customized configuration data, it is important to know the rules governing the import function and the options available to you.

The following rules apply when importing customized configuration data for a Tivoli Integrated Portal environment:

Rules and options for pages

Rule

1. You can import all pages included in an exported package.
2. You can exclude system customized pages that do not exist in the new environment.
3. You can exclude pages associated with a WAR that is not deployed in the new environment and thereby avoid introducing empty pages.
4. If a page contains multiple portlets and some of portlets are associated with a WAR that is not deployed in the new environment, the page is not imported.

Rules and options for views

All views included in an exported package are imported.

Rules and options for custom roles and role preferences (console preference profiles)

All roles included in an exported package are imported.

Rules and options for user preferences

All user preferences included in an exported package are imported.

Rules and options for console properties and customization properties

All console properties and customization properties included in an exported package are imported.

Rules and options for transformations

All transformations included in an exported package are imported, if the haSupport parameter is set to Both or False.

Table 1 provides details how various elements are processed during import:

Table 4. Rules for overwriting and merging during import

Element	Action	Comments
Pages	Overwritten	In relation to pages, roles are merged, view memberships remain unchanged, and positions are modified.
Views	Overwritten	In relation to views, existing page memberships are merged with imported pages
Roles	Skipped	In relation to roles, user and group mappings are merged.
Console preference profiles	Skipped	
Credential data	Merged	
Property files	Merged	
Transformations	Skipped	
Charts	Overwritten	

tipcli command reference

Export tipcli commands:

tipcli commands for exporting Tivoli Integrated Portal data.

Note: If you specify additional parameters for the tipcli.bat|.sh Export and make a typing error, that is, if you type a parameter incorrectly, or use the incorrect case, then the commands runs as if no parameters were specified and no warning message is displayed.

ListExportPlugins

Use the ListExportPlugins command to list all plugins that can be exported.

Export [--includePlugins|--excludePlugins *plugin1,plugin2*] [--settingFile *setting_file*] --username *tip_username* --password *tip_user_password*

Use the Export command to export customization data for an instance of

Tivoli Integrated Portal. If you provide no parameters to the Export command, all custom data is exported by default.

Table 5. Export command arguments

Parameter and arguments	Description
<code>[--includePlugins --excludePlugins plugin1,plugin2]</code>	Optional parameter. You can choose to include or exclude a list of plugins when you run the Export command.
<code>[--settingFile setting_file]</code>	Optional parameter. You can specify your export requirements in properties file instead of specifying your requirements using separate parameters at the command line. Provide a path to the settings file as the argument to the settingFile parameter. On systems running Windows you must use double backslashes characters (\\) when specifying the path to your settings file, for example, C:\\tmp\\export.properties. Command line parameters take precedence over entries in the settings file.
<code>--username tip_username</code>	Mandatory parameter. The user name for a user with the iscadmin role.
<code>--password tip_user_password</code>	Mandatory parameter. The password for the specified user name.

Export `[--exportFile export_file]` `[--pages ALL|NONE|page1,page2]` `[--views ALL|NONE|view1,view2]` `[--roles ALL|NONE|REQUIRED|role1,role2]` `[--exportPagesInViews true|false]` `[--userPreferences ALL|NONE|REQUIRED|user_ID1,user_ID2]` `[--consolePreferenceProfiles ALL|NONE|pref_ID1,pref_ID2]` `[--includeEntitiesFromApp war1,war2]` `[--includeCustomData true|false]` `[--includeCredentialData true|false]` `[--includeMytasks true|false]` `[--includeMyStartupPages true|false]` `[--includeTransformations true|false]` `--username tip_username` `--password tip_user_password`

Table 6. ExportPagePlugin command arguments

Parameter and arguments	Description
<code>[--exportFile export_file]</code>	Optional parameter. Specifies the path and file name for the exported data, for example, c:/tmp/extest.zip.
<code>[--pages ALL NONE page1,page2]</code>	Optional parameter. If you do not use the pages parameter, the default setting is ALL unless either exportPagesInViews or includeEntitiesFromApp is defined, then the default setting is NONE. You can also provide a list of pages that you want to export.

Table 6. ExportPagePlugin command arguments (continued)

Parameter and arguments	Description
<code>--views ALL NONE view1,view2]</code> <code>--exportpageinviews [true false]</code>	Optional parameter. If you do not use the views parameter, the default setting is ALL. You can also provide a list of views that you want to export and optionally specify that you want to export all pages associated with the specified views. Note: Whether the optional parameter <code>exportpageinviews</code> is set to true or false, if a view has a default node in the navigation pane associated with it, then the page associated with the node is always exported. This is also true, even if you specify NONE as the argument to the <code>--pages</code> parameter.
<code>--roles ALL NONE REQUIRED role1,role2]</code>	Optional parameter. You can export no roles, all roles, or a specific list of roles. The default setting is ALL unless the pages parameter or the <code>includeEntitiesFromApp</code> parameter is specified. Then, the default setting is set to REQUIRED.
<code>--exportPagesInViews true false]</code>	Optional parameter. Use this parameter, set to true, to export the pages associated with an exported view. The default value is false.
<code>--userPreferences ALL NONE REQUIRED user_ID1,user_ID2]</code>	Optional parameter. You can export preferences for all users, no users, or for a specified list of users by user ID. The default setting is ALL. This parameter overrides the <code>includeMytasks</code> and <code>includeMyStartupPages</code> parameters.
<code>--consolePreferenceProfiles ALL NONE pref_ID1,pref_ID2]</code>	Optional parameter. You can export no preference profile data, all preference profile data, or data for a specific list of preference profiles. The default setting is ALL. Note: If a console preference profile has a custom view as its default view, then that view is automatically exported. If the exported view has a default node in the navigation pane, then the associated page is automatically exported with the view.
<code>--includeEntitiesFromApp war1,war2]</code>	Optional parameter. You can provide a list of WARs to export pages that contain portlets associated with the listed WARs.
<code>--includeCustomData true false]</code>	Optional parameter. The default value is true. If is set to false, no customization data is exported.
<code>--includeCredentialData true false]</code>	Optional parameter. The default value is true. If is set to false, no credential data is exported.
<code>--includeMytasks true false]</code>	Optional parameter. The default setting is true. This parameter only applies when the <code>includeEntitiesFromApp</code> parameter is also specified.

Table 6. *ExportPagePlugin* command arguments (continued)

Parameter and arguments	Description
<code>[--includeMyStartupPages true false]</code>	Optional parameter. The default setting is true. This parameter only applies when the <code>includeEntitiesFromApp</code> parameter is also specified.
<code>[--includeTransformations true false]</code>	Optional parameter. The default setting is true.
<code>--username tip_username</code>	Mandatory parameter. The user name for a user with the <code>iscadmins</code> role.
<code>--password tip_user_password</code>	Mandatory parameter. The password for the specified user name.

Export `[--includeCharts ALL|NONE|page_ID1,page_ID2] --username tip_username --password tip_user_password`

Table 7. *ChartExportPlugin* commands

Parameter and arguments	Description
<code>[--includeCharts ALL NONE page_ID1,page_ID2]</code>	Optional parameter. You can export all charts, no charts, or specify a list of charts to be exported. The default setting is ALL. Note: If you run the Export command using the <code>--includeCharts</code> parameter, it must be run by the same user that started the Tivoli Integrated Portal Server.
<code>--username tip_username</code>	Mandatory parameter. The user name for a user with the <code>chartAdministrator</code> role.
<code>--password tip_user_password</code>	Mandatory parameter. The password for the specified user name.

Import tipcli commands:

`tipcli` commands for importing Tivoli Integrated Portal data.

Note: If you specify additional parameters for the `tipcli.bat|.sh` Import and make a typing error, that is, if you type a parameter incorrectly, or use the incorrect case, then the commands runs as if no parameters were specified and no warning message is displayed.

ListImportPlugins

Use the `ListImportPlugins` command to list all plugins that are available to be imported.

Import `[--includePlugins|--excludePlugins plugin1,plugin2] [--settingFile setting_file] [--backupDir backup_dir] --username tip_username --password tip_user_password`

Use the Import command to import customization data into a Tivoli Integrated Portal environment. If you provide no parameters to the Import command, all custom data is imported by default.

Table 8. Import command arguments

Parameter and arguments	Description
<code>[--includePlugins --excludePlugins plugin1,plugin2]</code>	Optional parameter. You can choose to include or exclude a list of plugins when you run the Import command.
<code>[--settingFile setting_file]</code>	Optional parameter. You can specify your import requirements in properties file instead of specifying your requirements using separate parameters at the command line. Provide a path to the settings file as the argument to the settingFile parameter. On systems running Windows you must use double backslashes characters (\\) when specifying the path to your settings file, for example, C:\\tmp\\import.properties. Command line parameters take precedence over entries in the settings file.
<code>[--backupDir backup_dir]</code>	You can specify a directory to save the backup data during an import operation so that if it is required you can subsequently restore settings.
<code>--username tip_username</code>	Mandatory parameter. The user name for a user with the iscadmin role.
<code>--password tip_user_password</code>	Mandatory parameter. The password for the specified user name.

Import `[--importFile import_file] [--rollback ALL] [--haSupport both|true|false] --username tip_username --password tip_user_password`

Table 9. ImportPagePlugin command arguments

Parameter and arguments	Description
<code>[--importFile import_file]</code>	Optional parameter. Specifies the path and file name for the data to be imported, for example, c:/tmp/extest.zip.
<code>[--rollback ALL]</code>	Optional parameter. Use the rollback parameter if you want to restore a Tivoli Integrated Portal environment to its pre-import state. You can only roll back an import if you have made no changes to the environment since you performed the import.
<code>[--haSupport both true false]</code>	Optional parameter. You can set this parameter to both, true, or false. The setting indicates whether to include load balancing data, the default value is both. If you set it to false, only non-load balancing data is imported, that is, transformations. If is set to true, only load balancing base data is imported. When it is set to both, both types of data are imported. This parameter can also be used in non-load balanced environments. If is set to true, only base data is imported. If you set it to false, only non-base data is imported, that is, transformations.

Resource types

You can use the console to create pages, roles, and views. All of these resources that you create using the console are assigned a resource type of Custom. With other resource types, a more limited set of actions are available.

The type of resource is determined by how it was created.

Core This resource type is central to the operation of the console. Core resources cannot be created or deleted in the console, and you cannot edit its properties. However, you can make other changes that do not alter the nature of the resource, for example, including a core page in a custom view.

System

This resource type is created by products and applications that deploy the resource to the console. For example, when an application is installed to the console environment, it can define certain pages, roles, and views needed to administer the application through the console. All of these have a resource type of System. Like core resources, system resources cannot be created or deleted. However, for views, pages, and folders, you can create copies of system resources, which are explained under System Customized. And like core resources, you can perform actions on a system resource, like changing access to the resource, without modifying its properties.

System Customized

This is a copy of a system resource with properties, such as the name of the resource, that have been changed in the console. The original system resource is always maintained, but the system customized version of the resource is used until the original is restored. When the system resource is restored, the system customized copy is deleted.

You can create system customized pages, folders, and views, but not roles, wires, or external URLs.

Custom

These are resources that you create using the console. Custom resources can be created, edited, and deleted by any user whose role has access to the **Page Management**, **View Management**, **Portlet Management**, and **Role Management** portlets under the **Settings** folder in the navigation.

Manage Global Refresh

Console administrators use Manage Global Refresh to configure portlet refresh settings for all users of the console. Portlet refresh is used to refresh the content of a single console module without reloading the entire console page. As a result, your experience with the console interface is quicker and more interactive. Use these settings to fine tune how each portlet refreshes its content individually on the page.

Using Manage Global Refresh

Use this module for the following tasks:

- Giving permission to console users to edit their own portlet refresh options.
- Configuring default refresh settings for console modules. Administrators can set values for refresh mode, refresh interval, and show timer settings. These settings become the default values for Configure Portlet Refresh.

- Setting the minimum refresh interval for each console module. Use this setting to prevent the performance impacts of too many calls to the server to refresh content.

Portlet refresh settings

Restore Default Configuration

Changes all of the displayed field values to the values that were last saved. At least one portlet must be selected to enable this button. To save the changes displayed by this button, select the portlets that you want to restore to the default settings and click **Apply** or **OK**.

Select all Select all icon

Selects all of the portlets displayed. A maximum of 10 refreshable portlets can be displayed and selected at a time.

Deselect all Deselect all icon

Deselects all of the portlets displayed.

Select Use the checkbox to select individual portlets that you want to restore to the default settings.

Portlet

Indicates the name of the portlet or console module which can be refreshed.

Refresh Mode

Select one of the following options:

- **No Refresh**

Indicates that the portlet content will not be refreshed automatically. The refresh timer is not displayed in the portlet title bar, but the portlet can still be refreshed manually.

- **Timed Refresh**

Indicates that the portlet content is refreshed automatically based on the value of the refresh interval.

- **Smart Refresh**

Indicates that after the refresh interval has timed out, the client should query the portlet on the server to determine if it should refresh the content. If the portlet has updates to provide, then the content is updated on the client. Otherwise, no change is made and the timer is started again.

- **Unregister**

Disables portlet refresh capabilities for this portlet. The portlet still displays in Manage Global Refresh. Portlet refresh can be subsequently restored by setting this value to one of the other settings.

Refresh Interval

Indicate a value in seconds after which the portlet's content can be refreshed from the server without reloading the entire console page. This value must be greater than or equal to the minimum refresh interval.

Minimum Refresh Interval

Indicates the minimum value for the refresh interval. This value is determined by the administrator.

User Configurable

Indicates whether users can change refresh setting in Configure Portlet Refresh.

Show Timer

Indicates whether to display a timer in the portlet title bar showing the number of seconds remaining until the next refresh can take place.

Administering users, roles, and groups

You can create different kinds of Web GUI users, assign them roles and add them to groups to determine their ability to perform tasks. You can also modify the preferences of a Web GUI user.

Web GUI users, roles, and groups

The users of all products that are installed into the Tivoli Integrated Portal framework, including the Web GUI, are centrally administered. Users are associated with roles and groups.

Groups and roles can be administered only by a Tivoli Integrated Portal administrator, that is, a user with the iscadmins role.

Web GUI users

Web GUI users can be classified in different ways. Users can be classified according to the roles that are assigned to them. This determines their ability to access features and administer content. Within the Web GUI, administrators and users are defined by the roles they are assigned.

The following table describes the different types of Web GUI users that can be defined using roles.

Table 10. Web GUI user types

Type of user	Roles necessary	Description
Administrator	Web GUI administrator role (ncw_admin) Web GUI user role (ncw_user)	The administrator can access the administrative functions and all event management functions of the Web GUI.
Read-write user	Web GUI user role (ncw_user) Web GUI read-write role (netcool_rw) Optional: Web GUI editor role for the portlet preferences of the Event Dashboard (ncw_dashboard_editor)	A read-write user can access event management functions, run AEL tools, and change the filter or view applied to an AEL or monitor box. Read-write users who are also ObjectServer users can modify ObjectServer data. Read-write users who are also assigned the ncw_dashboard_editor role can modify the portlet preferences of the event dashboard; these preferences control the layout of the monitor boxes, and the actions that can be executed from the monitor boxes.
Read-only user	Web GUI user role (ncw_user) Web GUI read-only role (netcool_ro)	A read-only user can access event management functions, but cannot run AEL tools or modify ObjectServer data.

Related tasks

“Creating the Web GUI administrative user” on page 69

“Administering users” on page 73

Web GUI roles

Roles must be assigned to users so that the users can view data and execute functions.

The user roles for the Web GUI are as follows:

ncw_user

This role gives a user access to the event management functions of the Web GUI. In addition to this role, the user must also have the ncw_admin role, netcool_rw role or the netcool_ro role; these roles control whether the user has administrative access, read-write access or read-only access to these functions.

ncw_admin

This role gives a user access to the administrative functions of the Web GUI. A user with this role also requires the ncw_user role.

netcool_rw

This role gives a user read-write access to Web GUI event management functions. Users with this role have access to AEL tools and can change event data. A user with this role also requires the ncw_user role.

netcool_ro

This role gives a user read-only access to Web GUI event management functions. Users with this role can access the AEL and view events, but cannot run AEL tools or change event data. A user with this role also requires the ncw_user role.

ncw_dashboard_editor

This role gives a user access to the portlet preferences of the Event Dashboard portlet. Read-only users can also have this role.

ncw_gauges_viewer

This role gives a user access to the Gauges page. Users without the ncw_user role require this role to view the Gauges page in a Web browser, or to view an HTML page generated from the Gauges page on a supported mobile device. Use this role in preference to ncw_user when you want to restrict a user to viewing gauges only and not other parts of the Web GUI, for example the Active Event List. Users with the ncw_admin role also require this role to view and edit a Gauges page.

ncw_gauges_editor

This role gives a user access to the portlet preferences of the Gauges page. Users with this role also require the ncw_gauges_viewer role. Users with the netcool_ro role can edit the portlet preferences of the Gauges page if they are assigned the ncw_gauges_editor and ncw_gauges_viewer roles.

Related information

“Administering roles” on page 80

User groups in the Web GUI

Groups can be used to logically categorize users into units with common functional goals.

Related tasks

“Administering groups” on page 84

Supplied users and groups

As supplied, the Web GUI has two users and two groups. These enable you to begin using the product as soon as installation is complete.

The Web GUI is supplied with two users (named ncouser and ncoadmin) and two groups (named Netcool_OMNIBus_User and Netcool_OMNIBus_Admin). Each user is a member of at least one of these groups, as follows:

Table 11. Group membership for the supplied users

User ID	Groups
ncouser	Netcool_OMNIBus_User
ncoadmin	Netcool_OMNIBus_Admin Netcool_OMNIBus_User

Group roles

The groups have the following roles:

Table 12. Roles assigned to the supplied groups

Group name	Roles assigned
Netcool_OMNIBus_Admin	ncw_admin ncw_dashboard_editor ncw_gauges_editor ncw_user netcool_rw
Netcool_OMNIBus_User	ncw_user netcool_ro

These assigned roles mean that ncouser is a read-only user and ncoadmin is an administrator.

Using the users and groups

The supplied users enable you to access the product as soon as the product is ready for use. They also provide a convenient means of accessing the product for temporary or demonstration purposes.

The groups provide a convenient way of allocating user or administrative privileges to any user you create. For example, as you create each read-only user you assign them to the Netcool_OMNIBus_User group.

Web GUI user administration

User administration in the Web GUI is divided between the Tivoli Integrated Portal administrators and the Web GUI administrators.

By default, the authorization for performing user administration tasks is as follows:

- Generic user administration tasks, such as creating users and assigning roles and groups, are performed by the Tivoli Integrated Portal administrators, that is, users with the `iscadmins` role.
- Modifying the preferences of a Web GUI user is performed by the Web GUI administrators, that is, users with the `ncw_admin` role.

Note: The Web GUI does not close sessions that are incorrectly logged-out and counts these as active sessions.

The following table lists the different user administration tasks, and shows what kind of user can perform them.

Table 13. User administration tasks

Task	Performed by
Creating users	Tivoli Integrated Portal administrator
Deleting users	Tivoli Integrated Portal administrator
Changing another user's password	Tivoli Integrated Portal administrator
Assigning roles to a user	Tivoli Integrated Portal administrator
Assigning groups to a user	Tivoli Integrated Portal administrator
Changing a user's display name	Tivoli Integrated Portal administrator
Changing your own password	Any user
Setting ObjectServer SQL filters for a user	Web GUI administrator
Setting the Web GUI home page for a user	Web GUI administrator
Setting event list preferences for a user	Web GUI administrator

Related tasks

“Creating the Web GUI administrative user” on page 69

“Administering users” on page 73

“Creating users” on page 75

“Deleting users” on page 79

“Changing information about a user” on page 77

“Changing passwords” on page 69

“Modifying the preferences of a Web GUI user” on page 70

“Assigning roles to users and groups” on page 92

Changing passwords

You can use the **Change Your Password** portlet to change your password from the default provided by the administrator.

When you log in to the portal, you can change your own password using the **Change Your Password** portlet. Administrators can change passwords for other users using the **Manage Users** portlet.

Attention: If you are an administrator and you want to change the password for the tipadmin administrator and the Tivoli Netcool/OMNIBus ObjectServer root user, you must use the **Settings > Change Your Password** portlet to change their password. Do not use the **Users and Groups > Manage Users** portlet.

Tip: For security reasons, change the password of the Tivoli Netcool/OMNIBus ObjectServer root user after installation.

To change passwords:

- To change your own password, follow these steps:
 1. Log in to the portal using the user ID whose password you would like to change.
 2. In the navigation pane, click **Settings > Change Your Password**.
 3. Enter your new password in the relevant fields and click **Set Password**.
- As an administrator, to change the password for a user, follow these steps:
 1. In the navigation pane, click **Users and Groups > Manage Users** and click the user's name from the **User ID** column. A **User Properties** page is displayed.
 2. In the **General** tab, enter the new password in the relevant fields and click **OK**.

Attention:

If you authenticate to a Microsoft Active Directory server, it must be configured for SSL before you can use the **Change Your Password** portlet. If SSL is not enabled, you will receive an error when attempting to change the password for any user who is registered on the Active Directory Server.

TIPCP0005E Could not set the password via the underlying security system. This could be because a password rule was not met, you do not have access to change the password, or another reason.

Related concepts

“Web GUI user administration” on page 68

Creating the Web GUI administrative user

The Web GUI is supplied with one administrative user, ncoadmin. It is good practice for the Tivoli Integrated Portal administrator to create one or more additional Web GUI administrative users to have permissions to modify Web GUI settings.

The administrator does not have to be an ObjectServer superuser.

If your configuration uses an ObjectServer as the user authentication source, note that user names cannot exceed 30 characters.

To create a Web GUI administrative user:

1. Log in as the Tivoli Integrated Portal administrator, that is a user with the `isc_admins` role.
2. Optional: Create a new user.
3. Click **Users and Groups > User Roles**.
4. Complete any combination of the search fields to help find the user.
5. Select the number of users to display and click **Search**.
6. Click the user ID of the user in the grid.
7. Set the check boxes for the following roles:
 - `ncw_admin` (Web GUI administrator)
 - `ncw_user` (Web GUI user)
8. Click **Save**.
9. Log in as the Web GUI administrative user and check that you have access to administrative pages.

The Web GUI can view the **Administration** entry in the navigation.

Related concepts

“Web GUI users” on page 65

“Web GUI user administration” on page 68

Related tasks

“Administering users” on page 73

Modifying the preferences of a Web GUI user

Edit the user profile settings and event list options for Web GUI users.

You can modify preferences for Web GUI users. All other user account administration tasks are handled by the Tivoli Integrated Portal administrator.

To modify the preferences and the event list options of a Web GUI user:

1. Click **Settings > User Preferences for Tivoli Netcool/OMNIBus Web GUI**.
2. Select a user and click **Modify**. The preferences and event list configuration settings are displayed for the selected user.
3. To modify the user profile and set the event list options for the user, set the following parameters:

User filter

Type ObjectServer SQL commands to filter alert data from the ObjectServer for the individual user. This filter is optional, and is the highest level of data filtering applied to a user session. The following example shows alerts only if they occur more than 100 times and have a severity of 4 or higher:

```
Tally > 100 AND Severity >=4
```

Note: If a user is ObjectServer-authenticated, the Restriction Filters defined in the ObjectServer override any user filters.

For more information about ObjectServer SQL syntax, see the *IBM Tivoli Netcool/OMNIBus Administration Guide*.

For more information about ObjectServer SQL syntax, go to the IBM Tivoli Network Management Information Center at <http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/index.jsp>, and search for *ObjectServer SQL*.

User's home-page

Type a default page for the user. This page is displayed when the user logs in to the Web GUI using the following URL:

`http://server:port/ibm/console/webtop`

To define the home page, you add the full URL or type a path to the page, for example: `/ibm/console/webtop/mypage.html`. The `mypage.html` file needs to be defined in the `tip_home_dir/profiles/TIPProfile/installedApps/TIPCell/isc.ear/OMNIBusWebGUI.war` directory. You can also save the files for user home pages to separate directories within the `OMNIBusWebGUI.war` location. In this case, remember to set the correct path. For example, if the `mypage.html` file is saved to the `.../OMNIBusWebGUI.war/pages` directory, then the path must be set to `.../pages/mypage.html`.

For more information about setting home pages for users, see the *IBM Tivoli Netcool/OMNIBus Web GUI Administration and User's Guide*.

Allow filter and view selection

Select this check box if you want the user to be able to select predefined filter and view settings from the drop-down menus for filters and views in the AEL toolbar.

Allow filter builder access

Select this check box if you want the user to be able to use the Filter Builder component of the AEL.

Allow view builder access

Select this check box if you want the user to be able to use the View Builder component of the AEL.

Allow preference configuration

Select this check box if you want the user to be able to change user preferences in the AEL. If you do not select this check box the **Preferences** option is not displayed to the user in the **Edit** menu of the AEL.

Allow refresh rate configuration

Select this check box if you want the user to be able to set the refresh rate of the AEL. The user must also have access to the **Preferences** window to be able to do this.

Refresh rate (seconds)

Type the value in seconds to set the default refresh rate for the AEL.

Minimum refresh rate (seconds)

Type the value in seconds to set the minimum refresh rate for the AEL.

Allow event selection

Select this check box if you want the user to be able to select alerts in the AEL.

Show basic event information

Select this check box if you want the user to have access to the Information window in the **Alerts** menu of the AEL. If selected, the user can view the **Fields** tab in the Information window. You can provide full access to the Information window, or limit what the user can see in this window by selecting or clearing the following check boxes.

Show event details

Select this check box if you want the user to be able to view the **Details** tab in the Information window.

Show journals

Select this check box if you want the user to be able to view the **Journal** tab in the Information window. To view the **Journal** tab, the user must also exist in the ObjectServer that the event is associated with.

Edit journals (read write role)

Select this check box if you want the user to be able to add journal entries.

Restriction: Only users who are also ObjectServer users and have the Web GUI read-write user role (netcool_rw) assigned can add journal entries.

Calendar Type

Choose the type of calendar to use for dates and times in the Web GUI. You can choose any of the following calendars:

Gregorian

Hebrew

Hijri

Hijri Civilian

Enable Bidi support

Select this check box if you want to enable support for bi-directional text in Web GUI portlets. When you set this you can specify further characteristics of the system's behavior:

Plain text alignment in editable fields and cursor position in empty editable fields

You can set either of two behaviors for fields on Web GUI pages that you can edit:

The text direction matches that set for the Web GUI itself.

The text direction follows the text direction set the portlet.

Text alignment in complex expression fields

You can set either of two behaviors for pages that contain complex expressions such as a SQL query, a file path, or a URL:

The text direction matches that set for the Web GUI itself.

The text direction is always left to right.

4. Click **Save** to save and activate the settings.

Related concepts

"Web GUI user administration" on page 68

"The Web GUI in a load balancing environment" on page 93

"Filters" on page 283

"Views" on page 285

Administering users

Use the Web console to create users and manage user profiles.

What to do next

From the left navigation pane, click **Manage Users**. Before you can perform some tasks for users, you must first search for existing users that match the search criteria that you specify. After the search completes, a table displays the users that match your search criteria. To manage users, you can perform these tasks:

Related concepts

“Web GUI users” on page 65

Related tasks

“Creating startup pages” on page 35

Searching for users

You can search for existing users that match the search criteria that you specify.

1. From the navigation pane, click **Manage Users**.
2. In the **Search by** field, select the attribute from the list that you want to use to search for one or more users. For example, select **User ID**.
3. In the **Search for** field, either type the string that you want to search for to limit the set of users, or use the wildcard character (*) to search for all users. Whether the search is case sensitive or case insensitive depends on the user registry that you are using.
4. In the **Maximum results** field, specify the maximum number of search results that you want to display.
5. Click **Search**. After the search completes, a table displays the users that match your search criteria.

What to do next

Creating users

Deleting users

Duplicating group assignments for a user


Customizing search filters for users

Changing the display options for the list of users

Customizing search filters for users:

You can create a filtered list of users by specifying the type of filter and the text to be used as part of the search criteria. The filtered list of users is limited to displaying only the users that meet the filter requirements. You can filter users by the letters that the user ID contains or by the letters that the user ID starts or ends with.

1. From the navigation pane, click **Manage Users**.
2. In the **Search by** field, select the attribute from the list that you want to use to search for one or more users. For example, select **User ID**.
3. In the **Search for** field, either type the string that you want to search for to limit the set of users, or use the wildcard character (*) to search for all users. Whether the search is case sensitive or case insensitive depends on the user registry that you are using.
4. In the **Maximum results** field, specify the maximum number of search results that you want to display.

5. Click **Search**. After the search completes, a table displays the users that match your search criteria.
6. Click the  filter icon.
7. Do one of the following:
 - To create a new filter for an attribute if none exists, click the **[No Filter]** link.
 - To select an existing filter, click on the filter name.
8. If you are creating a new filter, select a filter type from the list. The types of filters are Contains, Starts with, or Ends with.
9. Type the letters that you want to search for in the **Text** field. The wildcard character is not permitted in this field.
10. Click **Apply**.

What to do next

After you click **Apply**, the name of the filter is listed in the column under the attribute. The filter name matches the letters that were typed in the **Text** field. If no filter has been used for an attribute, the text **[No Filter]** displays.

The list of users is refreshed, and a filtered list of user IDs is displayed. The filter name and the total number of filtered user IDs are also shown.


Click the  hide filter icon, or click **Close**, when you have finished working with filters.


Searching for users

Changing the display options for the list of users

Changing the display options for the list of users:

You can change how the search results are displayed when viewing the list of users. For example, you can change the number of users to be viewed per page or you can display additional details about the users.

1. From the navigation pane, click **Manage Users**.
2. In the **Search by** field, select the attribute from the list that you want to use to search for one or more users. For example, select **User ID**.
3. In the **Search for** field, either type the string that you want to search for to limit the set of users, or use the wildcard character (*) to search for all users. Whether the search is case sensitive or case insensitive depends on the user registry that you are using.
4. In the **Maximum results** field, specify the maximum number of search results that you want to display.
5. Click **Search**. After the search completes, a table displays the users that match your search criteria.
6. Click the  options icon.
7. If you want to view more or fewer entries on a page, change the number in the **Entries Per Page** field.
8. If you want to show more details about a user, select one or more check boxes next to the attributes that you want to display additional columns for.

9. If you want to see the user IDs on a different page, type the page number in the field at the bottom of the list, and click **Go** to jump to that page.
10. Click **Apply**.
11. Click the  hide options icon, or click **Close**, after changing the display options for the list of users.

What to do next

Searching for users

Customizing search filters for users

Creating users

You can create one or more users. The users are added to the registry and a login account for each new user is automatically created. When creating the new user, you can also add the user as a member of one or more groups.

If your configuration uses an ObjectServer as the user authentication source, note that user names cannot exceed 30 characters.

1. Click **Manage Users** on the navigation pane.
2. Click **Create** to create a new user.
3. In the **User ID** field, type a unique name to identify the user. This user ID will be added to the user registry and also will be used as the login account name. For example, you might type d1ucas
4. Optional: Click **Group Membership** and then follow the steps in “Changing group membership for a user” on page 76 to add the user as a member of one or more groups.
5. In the **First name** field, type the given or first name of the user. For example, you might type Diana
6. In the **Last name** field, type the family or last name of the user. For example, you might type Lucas
7. Optional: In the **E-mail** field, type an e-mail address for the user. For example, you might type d1ucas@tivoli.com
8. In the **Password** field, type a unique password. For example, you might type d4lucas.
9. In the **Confirm password** field, type the same password again.
10. Click **Create**. If successful, a message will display that indicates that the user has been created. Also, the user ID and other user information will be added to the user registry, and a new login account will be created for the user.
11. To create another user, click **Create Like**.
12. Repeat the procedure until all the new users have been created.

What to do next

Related concepts

“Web GUI user administration” on page 68

Related tasks

“Assigning roles to users and groups” on page 92

Deleting users

Duplicating group assignments for a user

Changing group membership for a user

You can search for and list the existing groups that match the search criteria. When creating a new user, you can choose the groups from the search results list in which you want the user to be a member.

1. During the process of “Creating users” on page 75, click **Group Membership**.
2. In the **Search by** field, select the attribute from the list that you want to use to search for one or more users. For example, select **Group name**.
3. In the **Search for** field, either type the string that you want to search for to limit the set of groups, or use the wildcard character (*) to search for all groups. Whether the search is case sensitive or case insensitive depends on the user registry that you are using.
4. In the **Maximum results** field, specify the maximum number of search results that you want to display.
5. Click **Search**. After the search completes, the results are displayed in two lists: one list is for groups that matched the search criteria and one list, named **Current Groups**, is for groups that the user is already a member.
6. To add the user to one or more groups, highlight the groups from the matching groups list to select them. For example, you might highlight `ibmaustin01` and `ibmaustin02` and then click **< Add**.
7. Optional: To undo or remove the user as a member, highlight the groups from the **Current Groups** list and then click **Remove >**.
8. Return to the process of “Creating users” on page 75 to complete the steps.

What to do next

- Adding a user to other groups
- Changing information about a user
- Removing a user from other groups
- Viewing information about a user
- Viewing the groups the group is a member of

Viewing information about a user

You can view information about a specific user.

1. From the navigation pane, click **Manage Users**.
2. In the **Search by** field, select the attribute from the list that you want to use to search for one or more users. For example, select **User ID**.
3. In the **Search for** field, either type the string that you want to search for to limit the set of users, or use the wildcard character (*) to search for all users. Whether the search is case sensitive or case insensitive depends on the user registry that you are using.
4. In the **Maximum results** field, specify the maximum number of search results that you want to display.
5. Click **Search**. After the search completes, the users that match your search criteria are displayed as hypertext links.
6. Click on one of the user links to view information about the selected user. You can only view the information, you cannot change it.
7. Click **Cancel** after viewing to return to the previous window.

What to do next

- Changing information about a user
- Changing group membership for a user

Changing information about a user

You can change information about a specific user, such as the e-mail address. You can update the e-mail address, change the first or last name information, or set a new password.

1. From the navigation pane, click **Manage Users**.
2. Complete the steps in “Searching for users” on page 73 to find the user.
3. Click on one of the user links to change information about the user, as needed.
4. In the **First name** and **Last name** fields, enter the new information, if needed.
5. Optional: In the **E-mail** field, enter the new information, if needed.
6. Optional: In the **Password** and **Confirm password** fields, enter the new password, if needed, and confirm the new password.
7. To save the changes, either click **OK** to save and return to the previous window, or click **Apply** to save but remain on the same window.

What to do next

Related concepts

“Web GUI user administration” on page 68

Changing group membership for a user

Viewing information about a user

Viewing the groups the user is a member of

Viewing the groups the user is a member of

You can view a list of existing groups that the specified user is currently a member of.

1. From the navigation pane, click **Manage Users**.
2. Complete the steps in “Searching for users” on page 73 to find the user.
3. Click the user name link to see the user properties.
4. Click the **Groups** tab to see the list of groups, in ascending order, that the user is currently a member of.

What to do next

Adding a user to other groups

Changing group membership for a user

Changing information about a user

Removing a user from other groups

Viewing information about a user

Adding a user to groups

You can add a user as a member to selected groups.

1. Complete the steps in “Viewing the groups the user is a member of”
2. Confirm that the user in the **User ID** field is the user that you want to add to more groups.
3. Click **Add**.
4. In the **Search by** field, select the attribute from the list that you want to use to search for one or more groups. For example, select **Group name**.
5. In the **Search for** field, either type the string that you want to search for to limit the set of users, or use the wildcard character (*) to search for all users. Whether the search is case sensitive or case insensitive depends on the user registry that you are using.

6. In the **Maximum results** field, specify the maximum number of search results that you want to display.
7. Click **Search**. After the search completes, a list is displayed of the groups that match your search criteria.
8. Highlight one or more groups to select them, and click **Add**.
9. Repeat steps 3 to 7 for any other groups you want to add the user to.
10. Click **Close**.

What to do next

Removing a user from other groups
 Changing group membership for a user
 Changing information about a user
 Viewing information about a user
 Viewing the groups the user is a member of

Removing a user from groups

After searching for the groups in which the user is currently a member, you can remove the user from membership in groups that you select.

1. Complete the steps in “Viewing the groups the user is a member of” on page 77.
2. Confirm that the user in the **User ID** field is the user that you want to remove from other groups.
3. Select the check boxes next to one or more groups, and then click **Remove**.
4. Click **Remove** when asked to confirm the removal.

What to do next

Adding a user to other groups
 Changing group membership for a user
 Changing information about a user
 Viewing information about a user
 Viewing the groups the group is a member of

Adding users as members of a group

You can add more users as members of a group.

1. After completing the steps in “Viewing a list of members of a group” on page 89, click **Add Users**.
2. Confirm that the **Group name** field displays the name of the group to which you want to add more users as members.
3. In the **Search by** field, select the attribute from the list that you want to use to search by. For example, select **User ID**.
4. In the **Search for** field, either type the string that you want to search for to limit the set of users or use the wildcard character (*) to search for all users.
5. In the **Maximum results** field, specify the maximum number of search results that you want to display.
6. Click **Search**. After the search is complete, a list displays of the users that matched your search criteria.
7. Highlight one or more users in the list to select them, and then click **Add**.
8. Click **Close**.

What to do next

The users are immediately added to the list of members.

Adding more groups as members of a group

Removing members from a group

Duplicating group assignments for a user

You can search for users that match your search criteria. After selecting one or more users, these users can be added as members of the same groups that another existing user is already a member of. For example, if all the members of a department need to belong to the same groups as the department manager, you can duplicate the groups that the manager belongs to for all the other users that you choose.

1. Complete the steps in “Searching for users” on page 73.
2. Select the check boxes next to one or more users that you want to assign the same membership as another group.
3. Choose the **Duplicate Group Assignments** action.
4. In the **Search by** field, select the attribute from the list that you want to use to search for one or more users. For example, select **User ID**.
5. In the **Search for** field, either type the string that you want to search for to limit the set of users, or use the wildcard character (*) to search for all users. Whether the search is case sensitive or case insensitive depends on the user registry that you are using.
6. In the **Maximum results** field, specify the maximum number of search results that you want to display.
7. Click **Search**. After the search has been completed, a table displays the users that match your search criteria.
8. Highlight the name of the user whose group membership you want to duplicate for the previously selected users, and click **OK**. The group membership of the user is duplicated for the previously selected users.

What to do next

Creating users

Deleting users

Deleting users

You can search for and list the existing users that match your search criteria. After selecting one or more users, you can delete them and remove their user IDs from the user registry.

1. From the navigation pane, click **Manage Users**.
2. In the **Search by** field, select the attribute from the list that you want to use to search for one or more users. For example, select **User ID**.
3. In the **Search for** field, either type the string that you want to search for to limit the set of users, or use the wildcard character (*) to search for all users. Whether the search is case sensitive or case insensitive depends on the user registry that you are using.
4. In the **Maximum results** field, specify the maximum number of search results that you want to display.
5. Click **Search**. After the search completes, a table displays the users that match your search criteria.
6. Select the check boxes next to the users that you want to delete.

7. Click **Delete**.
8. Click **Delete** again when asked to confirm the deletion. The users are immediately deleted and removed from the user registry. The table that lists the users is refreshed, and the selected users are no longer displayed in the list.

What to do next

Related concepts

“Web GUI user administration” on page 68

Creating users

Duplicating group assignments for a user

Administering roles

Console users are granted access to resources based on the role to which they have been assigned. In the console navigation, click **Users and Groups > Role Management** to add and remove roles and to assign access to portlets, pages, and views.

To manage users and groups and assign them to roles, click **Users and Groups**.

After the console is installed, there are some roles already defined to the server.

Attention: The “suppressmonitor” role is used to hide the tasks associated with the application server, including the tasks in the Security, Troubleshooting, and Users and Groups folders.

Access levels

The access level that a role has to a resource determines the actions that users within that role can perform on the resource.

Table 14. Access rights to console resources based on access level

Resource	Access Level		
	“User”	“Privileged User”	“Editor”
Portlet	View and interact with the portlet and access portlet help	View and interact with the portlet, edit personal settings, and access portlet help	View and interact with the portlet, edit personal settings, edit global settings, and access portlet help
Page	Launch the node from the navigation		Launch the node from the navigation and edit the content and layout
Folder	Note: Folders are always available in the navigation if the user has access to at least one of its pages.		
External URL	Launch the node from the navigation		
View	Select the view		

For a given resource, if a role does not have one of these access level settings, then the role has no access to the resource.

Only users with “adminsecuritymanager” and “Administrator” role can create, delete or change the properties of a role. If you assign access for any other role to the Role Management portlet, users in that role will only be able to view roles and change access to views and pages.

Note: The access control settings are not observed when using the administrative portlets under the **Settings** node. Users with access to these pages and portlets will be able to create, edit, and delete all custom pages, portlets, and views. For example, if a user has no access to “Page Two”, but has access to Page Management, that user can edit all of the properties of “Page Two” and change access control settings. Keep this in mind when granting access to the **Settings** portlets for a role.

If a user is assigned to multiple roles, the user acquires the highest access level between these roles for a resource. For example, if a user belongs to the manager role with “Privileged User” access to a portlet and also belongs to the communications role with no access to the portlet, then the user has “Privileged User” access to the portlet.

Tasks

You can grant access for multiple roles while creating or editing a resource, such as a page or a portlet. You can also grant access to multiple pages or views while creating or editing a role.

Related concepts

“Web GUI roles” on page 66

Related information

“Administering views” on page 36

“Administering portlets” on page 40

“Administering pages” on page 28

Managing roles for users

Administrators can search for users and manage their roles in the User Roles page.

To search for users and manage their roles:

1. In the navigation pane, click **Users and Groups > User Roles**. The User Roles page is displayed.
2. In the search fields provided, you can enter search criteria by given name, surname, user ID, and e-mail address. If you do not have exact details for a particular item, all of the search fields support using an asterisk (*) as a wildcard character. For example, to return all user records with a given name that starts with “Mich”, enter mich* in the **First name** field.

Tip: You can leave the search fields blank to return all user records.

3. From the **Number of results to display** list, select the number of records that you want returned and click **Search**.

Restriction: Returned records are displayed one page only. If more records are available than the setting you chose from the list, only a partial list is returned. To display all records you need to search again after selecting a larger number from the **Number of results to display** list.

A list of records that match your search criteria are listed in the grid below the search fields.

4. Select a user from the **Unique Name** column. A list of available roles for the selected user is displayed on a new page. Those roles that are currently associated with the selected user are checked.
5. Modify the roles associated with the user as required, that is, check the roles that you want associated with the user and clear those that you do not.
6. Click **Save** to commit your changes, or **Reset** to reset the form to its initial state. Once you click **Save**, the User Roles page is displayed. The entry for the user in the **Roles** column is updated to reflect your changes.

What to do next

You can select another user from the search results and update their role settings, enter new search criteria to manage other user records, or close the User Roles page.

Managing roles for groups

Administrators can search for groups and manage their roles in the Group Roles page.

To search for user groups and manage their roles:

1. In the navigation pane, click **Users and Groups > Group Roles**. The Group Roles page is displayed.
2. In the search fields provided, you can enter search criteria by group ID and description. If you do not have exact details for a particular item, both search fields support using an asterisk (*) as a wildcard character. For example, to return all group records with a group ID that starts with "tes", enter tes* in the **Group ID** field.

Tip: You can leave the search fields blank to return all records.

3. From the **Number of results to display** list, select the number of records that you want returned and click **Search**.

Restriction: Returned records are displayed one page only. If more records are available than the setting you chose from the list, only a partial list is returned. To display all records you need to search again after selecting a larger number from the **Number of results to display** list.

A list of records that match your search criteria are listed in the grid below the search fields.

4. Select a group from the **Unique Name** column. A list of available roles for the selected group is displayed on a new page. Those roles that are currently associated with the selected group are checked.
5. Modify the roles associated with the group as required, that is, check the roles that you want associated with the group and clear those that you do not.
6. Click **Save** to commit your changes, or **Reset** to reset the form to its initial state. Once you click **Save**, the Group Roles page is displayed. The entry for the group in the **Roles** column is updated to reflect your changes.

What to do next

You can select another group from the search results and update its role settings, enter new search criteria to manage other group records, or close the Group Roles page.

Creating roles

Console users are granted access to resources based on the role to which they have been assigned. All roles that are created in the console have a resource type of Custom. This procedure walks you through the task of creating a role for testing purposes. After completing these steps, you can remove or edit this role for production use.

1. Click **Users and Groups > Role Management** in the navigation. A list of all roles in the console is displayed.
2. Click **New**. The properties panel for the new role is displayed.
3. Enter a descriptive name for the role.
4. Expand the **Access to Views** section. Use this section to grant access to one or more custom views for users who are assigned to the new role. If you have already created a custom view, follow these steps.
 - a. Click **Add**. A list of available views is displayed.
 - b. Select one or more views and click **OK**.
 - c. To make sure the role has access to all of the pages within the view, click **Grant to All**.
5. Expand the **Access to Pages** section. A list of pages that the role can access is displayed. However, this list is empty if you did not add a view and grant access to all of the pages within the view.
6. Optional: Click **Add** to grant access to additional pages.
7. For each page that is listed, verify that the **Access Level** is set correctly.
8. Click **Save** to save your changes and return to Role Management.

Results

The new role is created with access to the views and pages that you indicated. To grant access to the portlets on those pages you must edit the portlets.

Related concepts

“Web GUI roles” on page 66

Related tasks

“Assigning roles to users and groups” on page 92

Editing roles

Console users are granted access to resources based on the role to which they have been assigned. If you have sufficient authorization in the console, you can change the name of custom roles. For all roles, you can change access to views and pages and set the access level to pages.

1. In the navigation pane, click **Users and Groups > Role Management**. A list of all roles in the console is displayed.
2. Click the name of the role that you want to edit. The properties panel for the role is displayed. If this is a custom role, the only field you can edit is **Role Name**. For all other resource types, you cannot edit any of the role properties.
3. Expand the **Access to Views** section. Use this section to grant access to one or more custom views for users who are assigned to the new role. If you have already created a custom view, follow these steps.
 - a. Click **Add**. A list of available views is displayed.
 - b. Select one or more views and click **OK**.
 - c. To make sure the role has access to all of the pages within the view, click **Grant to All**.

4. Expand the **Access to Pages** section. A list of pages that the role can access is displayed. However, this list is empty if you did not add a view and grant access to all of the pages within the view.
5. Optional: Click **Add** to grant access to additional pages.
6. For each page that is listed, verify that the **Access Level** is set correctly.
7. Click **OK**.

Results

Your changes are saved and you are returned to the Role Management page.

What to do next

For any pages that you added for the role, you should ensure that the role also has access to the portlets on the page..

Related concepts

“Web GUI roles” on page 66

Related tasks

“Assigning roles to users and groups” on page 92

Deleting custom roles

You can delete only roles with the resource type of Custom. These are roles created using the console.

Attention: Before deleting a role, consider whether any users are actively using the role and any impacts this might have on services. If necessary, notify users in advance of any plans for changes that could affect their work.

Follow these steps to delete a custom role.

1. Click **Users and Groups > Role Management** in the navigation pane. The Role Management page is displayed with the list of roles in the console.
2. Select the custom role that you want to delete. You can select more than one custom role.
3. Click **Delete**. A message is displayed at the top prompting you to confirm the deletion.
4. Click **OK**.

Results

The custom role is removed from the list.

Administering groups

You can perform tasks that help you manage groups.

What to do next

From the left navigation pane, click **Users and Groups > Manage Groups**. Before you can perform some tasks for groups, you must first search for existing groups that match the search criteria that you specify. After the search completes, a table displays the groups that match your search criteria. To manage groups, you can perform these tasks:

Related concepts

“User groups in the Web GUI” on page 67

Related tasks

“Creating startup pages” on page 35

Creating groups

Deleting groups

Duplicating group assignments for a group

Customized search filters for groups

Changing the display options for the list of groups

Searching for groups

You can search for existing groups that match the search criteria that you specify.

1. From the navigation pane, click **Users and Groups > Manage Groups**.
2. In the **Search by** field, select the attribute from the list that you want to use to search for one or more groups. For example, select **Group name**.
3. In the **Search for** field, either type the string that you want to search for to limit the set of groups, or use the wildcard character (*) to search for all groups. Whether the search is case sensitive or case insensitive depends on the user registry that you are using.
4. In the **Maximum results** field, specify the maximum number of search results that you want to display.
5. Click **Search**. After the search completes, a table is displayed that lists the group names that match your search criteria. Descriptions, if any, are also provided.

What to do next

Creating groups

Deleting groups


Duplicating group assignments for a group


Customized search filters for groups

Changing the display options for the list of groups

Customizing search filters for groups:

You can create a filtered list of groups by specifying the type of filter and the text to be used as part of the search criteria. The list of groups will be limited because only the groups that meet the extended search criteria will be displayed.

1. From the navigation pane, click **Users and Groups > Manage Groups**.
2. In the **Search by** field, select the attribute from the list that you want to use to search for one or more groups. For example, select **Group name**.
3. In the **Search for** field, either type the string that you want to search for to limit the set of groups, or use the wildcard character (*) to search for all groups. Whether the search is case sensitive or case insensitive depends on the user registry that you are using.
4. In the **Maximum results** field, specify the maximum number of search results that you want to display.
5. Click **Search**. After the search completes, a table displays the groups that match your search criteria.
6. Click the  filter icon.

7. Do one of the following:
 - To create a new filter for an attribute if none exists, click the **[No Filter]** link.
 - To select an existing filter for an attribute, click on the filter name link.
8. If you are creating a new filter, select a filter type from the list. The types of filters are **Contains**, **Starts with**, or **Ends with**.
9. Type the letters that you want to search for in the **Text** field. The wildcard character is not permitted in this field.
10. Click **Apply**. The list of groups is refreshed and a filtered list of group names displays. The filter name and the total number of filtered group names are also shown.
11. Click the  hide filter icon, or click **Close**, when you have finished working with filters.

What to do next



If no filter has been applied for an attribute, the text **[No Filter]** displays. If you click **Apply**, the name of the filter is listed in the column under the attribute. The filter name matches the letters that were typed in the **Text** field.

Searching for groups

Changing the display options for the list of groups

Changing the display options for the list of groups:

You can change how the search results are displayed when viewing the list of groups. For example, you can change the number of groups to be viewed per page.

1. From the navigation pane, click **Manage Groups**.
2. In the **Search by** field, select the attribute from the list that you want to use to search for one or more groups. For example, select **Group name**.
3. In the **Search for** field, either type the string that you want to search for to limit the set of groups, or use the wildcard character (*) to search for all groups. Whether the search is case sensitive or case insensitive depends on the user registry that you are using.
4. In the **Maximum results** field, specify the maximum number of search results that you want to display.
5. Click **Search**. After the search completes a table displays the groups that match your search criteria.
6. Click the  options icon.
7. If you want to view more or fewer entries on a page, change the number in the **Entries per page** field, and click **Apply**.
8. If you want to see the group names on a different page, type the page number in the field at the bottom of the list, and click **Go** to jump to that page.
9. Click the  hide options icon, or click **Close**, after changing the display options for the list of groups.

What to do next

Searching for groups

Customizing search filters for groups

Creating groups

You can create one or more groups. The group names and descriptions are added to the user registry.

1. From the navigation pane, click **Users and Groups > Manage Groups**.
2. Click **Create** to create a new group.
3. In the **Group name** field, type a name to be used to identify the group. This group name will be added to the user registry. For example, you might type `ibm`
4. Optional: In the **Description** field, type a brief description for the group to distinguish this group from other groups. This description will be added to the user registry. The description must be an alphanumeric string with characters that are part of the local code set. For example, `Users and groups, CNC Company Dept 047`
5. Click **Create** to add the group name and the description, if entered, to the user registry. If successful, a message displays indicating that the group has been created.
6. To create another group, click **Create like**.
7. Repeat the procedure until all the new groups have been created.

Related tasks

“Assigning roles to users and groups” on page 92

Deleting groups

Duplicating group assignments for a group

Viewing information about a group

You can view information about a specific group.

1. Complete the steps in “Searching for groups” on page 85.
2. Click on one of the group name links to view the information about the selected group. You can only view the information, you cannot change it.
3. Click **Cancel** after viewing to return to the previous window.

What to do next

Adding a group to other groups

Changing information about a group

Removing the group from other groups

Viewing the groups the group is a member of

Changing information about a group

You can change the information about a specific group. You can change the name of the group, add a new description for the group if none exists, or change the existing description.

1. From the navigation pane, click **Users and Groups > Manage Groups**.
2. In the **Search by** field, select the attribute from the list that you want to use to search for one or more groups. For example, select **Group name**.
3. In the **Search for** field, either type the string that you want to search for to limit the set of groups, or use the wildcard character (*) to search for all groups. Whether the search is case sensitive or case insensitive depends on the user registry that you are using.
4. In the **Maximum results** field, specify the maximum number of search results that you want to display.

5. Click **Search**. After the search completes, the groups that match your search criteria are displayed in the column as hypertext links.
6. Click on one of the group links to change information about the group, as needed.
7. Optional: In the **Group name** field, enter a different name for the group, if needed.
8. Optional: In the **Description** field, enter a different description than the existing description, or enter a new description if none currently exists, if needed.
9. To save the changes, either click **OK** to save and return to the previous window, or click **Apply** to save but remain on the same window.

What to do next

Viewing information about a group

Viewing the groups the group is a member of

Viewing the groups the group is a member of

You can view a list of existing groups that the specified group is currently a member of.

1. From the navigation pane, click **Users and Groups > Manage Groups**.
2. In the **Search by** field, select the attribute from the list that you want to use to search for one or more groups. For example, select **Group name**.
3. In the **Search for** field, either type the string that you want to search for to limit the set of groups, or use the wildcard character (*) to search for all groups. Whether the search is case sensitive or case insensitive depends on the user registry that you are using.
4. In the **Maximum results** field, specify the maximum number of search results that you want to display.
5. Click **Search**. After the search completes, a table displays a list of groups, as hypertext links, that match your search criteria.
6. Click the group name link to see the group properties.
7. Click the **Groups** tab to see the list of groups, in ascending order, that the group is currently a member of.

What to do next

Adding a group to other groups

Changing information about a group

Removing a group from other groups

Viewing information about a group

Viewing the members of a group

Adding a group to other groups

Before you can add a group to other groups, you must first search for the groups in which you want the group to be a member. Note that when an ObjectServer registry is used to manage users and groups you cannot add groups to other groups.

Restriction: If the ObjectServer acts as the user repository for your system, a group cannot contain other groups.

1. Complete the steps in "Viewing the groups the group is a member of."
2. Confirm that the group in the **Group name** field is the group that you want to add to other groups.



3. Click **Add**.
4. In the **Search by** field, select the attribute from the list that you want to use to search for one or more groups. For example, select **Group name**.
5. In the **Search for** field, either type the string that you want to search if for to limit the set of groups, or use the wildcard character (*) to search for all groups. Whether the search is case sensitive or case insensitive depends on the user registry that you are using.
6. In the **Maximum results** field, specify the maximum number of search results that you want to display.
7. Click **Search** . After the search completes, a list is displayed of the groups that match your search criteria.
8. Highlight one or more groups to select them, and click **Add**.
9. Click **Close**.

Removing the group from other groups

Viewing the members of a group

Viewing a list of members of a group

You can search for a list of users and groups that are existing members of a specific group.

1. From the navigation pane, click **Users and Groups > Manage Groups**.
2. Complete the steps in “Searching for groups” on page 85.
3. Click the name of a group.
4. Click the **Members** tab to view the users and groups that are existing members of the specified group. Icons are used to help distinguish a user  from a group  member.

What to do next

Adding more groups as members of a group

Adding more users as members of a group

Removing members from a group

Removing a user from groups

After searching for the groups in which the user is currently a member, you can remove the user from membership in groups that you select.

1. Complete the steps in “Viewing the groups the user is a member of” on page 77.
2. Confirm that the user in the **User ID** field is the user that you want to remove from other groups.
3. Select the check boxes next to one or more groups, and then click **Remove**.
4. Click **Remove** when asked to confirm the removal.

What to do next

Adding a user to other groups

Changing group membership for a user

Changing information about a user

Viewing information about a user

Viewing the groups the group is a member of

Adding groups as members of a group

After viewing the list of members in a group that you specify, you can add more groups. Note that when an ObjectServer registry is used to manage users and groups you cannot add groups as members of a group.

Restriction: If the ObjectServer acts as the user repository for your system, a group cannot contain other groups.

1. After completing the steps in “Viewing a list of members of a group” on page 89, click **Add Groups**.
2. Confirm that the **Group name** field displays the name of the group to which you want to add more groups as members.
3. In the **Search by** field, select the attribute from the list that you want to use to search by. For example, select **Group name**.
4. In the **Search for** field, either type the string that you want to search for to limit the set of groups, or use the wildcard character (*) to search for all groups.
5. In the **Maximum results** field, specify the maximum number of search results that you want to display.
6. Click **Search**. After the search is complete, a list of the groups that matched your search criteria is displayed.
7. Highlight one or more groups in the list to select them, and then click **Add**.
8. Click **Close**.

What to do next

The groups are immediately added to the list of members.

Adding more groups as members of a group

Removing members from a group

Removing members from a group

After searching for the list of members in a group that you specify, you can remove users and groups as members from the group.

1. Complete the steps in “Viewing a list of members of a group” on page 89.
2. Confirm that the **Group name** field displays the name of the group from which you want to remove members.
3. Select the check boxes next to one or more group members.
4. Click **Remove**.
5. Click **Remove** again when queried to confirm the removal of the selected members from the group. The members are immediately removed and are no longer displayed in the table.

What to do next

Adding more users as members of a group

Adding more groups as members of a group

Duplicating group assignments for a group

You can search for groups that match the search criteria that you specify. After selecting one or more groups, these groups can be members of the same groups as another existing group. For example, if all groups at the Austin site should belong to the same groups as an existing group, you can duplicate the groups that the one group belongs to for all the other groups that you choose.

1. Complete the steps in “Searching for groups” on page 85.
2. Select the check boxes next to one or more groups that you want to assign the same membership as another group.
3. Choose the **Duplicate Group Assignments** action.
4. In the **Search by** field, select the attribute from the list that you want to use to search for one or more groups. For example, select **Group name**.
5. In the **Search for** field, either type the string that you want to search for to limit the set of groups, or use the wildcard character (*) to search for all groups. Whether the search is case sensitive or case insensitive depends on the user registry that you are using.
6. In the **Maximum results** field, specify the maximum number of search results that you want to display.
7. Click **Search**. After the search has been completed, a table displays the groups that match your search criteria.
8. Highlight the name of the group whose group assignment is to be duplicated for the previously selected groups, and click **OK**. The group membership of the group is duplicated for the previously selected groups

What to do next

Creating groups

Deleting groups

Deleting groups

You can search for and list the existing group names that match the search criteria that you specify. After selecting one or more groups, you can delete them and remove the group names from the user registry.

1. Complete the steps in “Searching for groups” on page 85.
2. Select the check boxes next to the groups that you want to delete.
3. Click **Delete**.
4. Click **Delete** again when asked to confirm the deletion. The groups are immediately deleted and removed from the user registry. The table that lists the groups is refreshed, and the selected groups are no longer displayed in the list.

What to do next

Creating groups

Duplicating group assignments for a group

Assigning roles to users and groups

After you have created the required users, groups, and roles, or after the users have been imported from the selected user registry, assign the roles to the users and groups.

To assign roles to users and groups:

1. To assign roles to users:
 - a. Click **Users & Groups > User Roles**.
 - b. Complete any combination of the search fields to help locate the users.
 - c. Select how many users to display and click **Search**. A list of matching users appears in the grid.
 - d. Click the user ID of the user you want to assign roles to.
 - e. From the **Role(s)** list, select the roles to assign the user.
 - f. Click **Save**.
2. To assign roles to user groups:
 - a. Click **Users & Groups > Group Roles**.
 - b. Complete any combination of the search fields to help locate the groups.
 - c. Select how many groups to display and click **Search**. A list of groups appears in the grid.
 - d. Click the name of the group you want to assign roles to.
 - e. From the **Role(s)** list, select the roles to assign the user group.
 - f. Click **Save**.

Results

The changes take effect immediately. Users must log out and back in before they can access the functions specified by the new roles.

Related concepts

"Web GUI user administration" on page 68

Related tasks

"Creating roles" on page 83

"Editing roles" on page 83

"Creating groups" on page 87

"Creating users" on page 75

Administering a load balancing cluster

Tasks for administering the Web GUI in a load balancing environment.

For instructions on how to set up a load balancing cluster, add nodes to a cluster, and remove nodes from a cluster, refer to the *IBM Tivoli Netcool/OMNIBus Installation and Deployment Guide*.

For instructions on how to set up a load balancing cluster, add nodes to a cluster, and remove nodes from a cluster see Setting up and configuring a load balancing environment.

Related concepts

"The Web GUI in a load balancing environment" on page 93

The Web GUI in a load balancing environment

Information on how the Web GUI can operate in a load balancing environment and the implications for administering and using the product.

A load balancing environment consists of a group of Web GUI servers that are linked together and operate as a single server. The name for the group of servers is a *cluster* and each of the servers is known as a *node*.

The primary benefits of a cluster are as follows:

- Load balancing – where the workload of servicing user requests is spread among the nodes. This improves the overall performance of the system.
- Availability – to maintain the availability of network monitoring even if some cluster nodes are unavailable for any reason (for example, they are shut down for maintenance).

The following sections contain more information on clusters, on administering them, and on using them:

- “Structure of a cluster”
- “Configuration data”
- “Updating configuration data” on page 94
- “Conditions necessary for changing configuration data” on page 95
- “Administering a load balancing cluster” on page 96
- “Using a load-balanced cluster” on page 97

Related concepts

“Administering a load balancing cluster” on page 92

Structure of a cluster

A cluster consists of a group of Web GUI servers, an HTTP server, and a DB2 database.

- The servers carry out service requests from users. In addition, each server is configured to trust the other servers in the cluster and is able to communicate with all the other cluster members. This enables them to cooperate as a single unit.
- The HTTP server distributes user HTTP sessions among the servers. It allocates requests among the servers either randomly or in a round robin fashion. The method that the HTTP server uses depends on how it was set up when installed.
- The DB2 database holds the configuration data for the cluster.

Configuration data

Configuration data defines how a Web GUI server operates. It is held differently in a cluster than it is in a standalone server.

A standalone Web GUI server holds its configuration data in the local file system. In a cluster, the DB2 database holds the configuration data for the entire cluster. This is the master copy of the data that is shared by all the cluster nodes. A single set of configuration data means that each node is configured identically. There is no configuration data that is specific to a cluster node.

Although the database holds the master copy, each node also has a copy in its local file system. This is for fault tolerance reasons and allows the cluster to continue operation should the configuration database become unavailable during operation.

When a node starts it reads a complete set of configuration data from the database into the local file system and loads it into memory to improve performance.

The configuration data that the database holds includes:

- Data sources
- Users, groups, and roles
- Page layouts, customized page information, and portlet descriptors
- Deployment descriptors
- Filters and views
- All items in the configstore:
 - AEL menus and menu configuration data
 - Metrics for gauges
 - Prompts
 - Tools
 - User preferences
- AEL preferences such as the refresh time and the number of rows to display
- Web GUI properties such as the default time zone and the timeout period
- Maps and resources, together with their properties
- Gauges and their properties
- Charts and their properties
- Predictive eventing information
- TADDM events
- Access information for the Inline frame portlet

Updating configuration data

Changes to the configuration data need to be coordinated across the entire cluster irrespective of which node initiated the changes.

The configuration data can change in any of three ways:

- The facilities of the Web GUI itself (for example, setting a set of preferences for a portlet)
- By editing the configuration files directly (for example, setting the metrics for a gauge)
- By using WAAPI commands (for example, enabling predictive eventing)

A change could originate on any cluster node. However, this change needs to be propagated to the entire cluster to maintain the commonality of the cluster's configuration.

Updating the database

The process for changing configuration data is as follows:

1. A user on a node changes an item in the configuration and requests that the node saves the change.
2. The node writes the new information (for example a configuration file) to the database.
3. The node notifies all the other cluster nodes that there is revised configuration information.
4. The node updates its local copy of the configuration data to reflect the change.

5. The other cluster nodes read the new information from the database and update the copies in their local file systems.
6. The cluster continues to operate with the new configuration settings.

Detecting changes to the configuration files

It is not always necessary to restart the cluster or any of its nodes to pick up the new configuration information. Instead, revised configuration data is automatically applied when it occurs. This is achieved through:

- The Web GUI timed tasks facility
- A file that lists the files to be monitored and an associated set of monitor processes

Timed tasks determine when each node loads changed files from the database.

The file is named *webgui_home_dir/etc/system/stores.list* and contains a list of all the configuration files that are kept in the database. When a node starts or joins the cluster it creates a set of processes that monitor each of the files listed in *stores.list*. Whenever a change occurs to one of those files, the corresponding process propagates the changed file to the DB2 database and notifies other nodes of the change.

This file monitoring capability means that an individual component of the Web GUI (such as a portlet) does not need to know whether configuration information is maintained in a database or in the local file system. Instead, the component always writes changes to its configuration directly to the local file system. The monitoring processes take care of updating the database.

There are some exceptions where a restart of a node, and usually the cluster, is necessary. Changing any of the following files requires a restart of the server:

- *server.init*
- *ncwDataSourceDefinitions.xml*

Conditions necessary for changing configuration data

To be able to operate correctly, certain conditions need to be met before the cluster can allow changes to its configuration data.

For the cluster to operate successfully, the DB2 database must be available. The database is the key coordination point of the cluster because it contains the configuration data.

If the database becomes unavailable after the cluster has started, operations continue, with each node using their local copy of the configuration data. However, each node prevents any changes occurring to the configuration data. This state continues until the DB2 database becomes available again. At this point, the cluster nodes refresh their locally stored configuration from the database and allow changes to configuration data to take place once more. The policy of allowing changes to occur only when the database is available helps to ensure that the cluster remains synchronized and that common behavior is maintained across the cluster.

When a node starts and joins the cluster, it reads the configuration data from the database, even though it may have data in its local file system. Nodes do this to ensure that they always have the latest set of configuration data. If the database is

unavailable when a node starts, it cannot continue because it cannot be sure that the local copy of the configuration data is up to date.

In addition to the configuration data, each cluster node must be run the same version of the Web GUI, with the same set of features, and set up in the same way. As with the data, this is the only way to provide a common service to the users of the Web GUI.

Administering a load balancing cluster

Administration of a load balancing cluster has two aspects that you need to be aware of:

- “Day-to-day administration”
- “Cluster administration”

Day-to-day administration

In day-to-day administration, bear in mind that any change you make always applies to the entire cluster not just the node where you make the change. For instance, adding a user to one node adds that user to all nodes. There may be a short time delay before a change is applied to all the nodes. This depends on how often the timed tasks interval is set, and how long it is until the next execution of the timed tasks facility.

An advantage of this propagation of configuration data is that it simplifies your administration job. You need only to make each change once, and the cluster ensures that all nodes receive it. If the database is unavailable, you cannot make any change to the configuration data. When using the Web GUI itself, the system prevents you from saving any changes to the data. When editing files or using WAAPI, the node you are using will not propagate the changed information until the database becomes available once more.

Cluster administration

After set up, a cluster requires little administration over and above the day-to-day administration that any Web GUI installation requires. However, the Web GUI provides a comprehensive set of tools for you to administer the cluster. These tools enable you to do the following:

- Enable load balancing after installation
- Administer the timed tasks facility
- Add and remove nodes
- Resynchronize a cluster node
- Export configuration information from a test environment into production
- Maintain the list of files to be monitored and propagated to the database whenever they change

Related tasks

“Administering timed tasks” on page 7

Using a load-balanced cluster

To users, the Web GUI behaves almost identically in a clustered environment as a standalone server. All that many users might notice, after a move to a clustered environment, is an increased responsiveness of the product. This is due to the overall increases in performance that the cluster provides.

Maintaining the list of files to monitor

On each cluster member has a copy of a file named `stores.lst` that holds a list of files to monitor for changes. When any of these files changes, the monitoring process copies it to the database.

You can add further files to the list for monitoring and saving to the database. Carry out the following procedure on each member of the cluster.

Important: You can add further files to store in the database only. Do not modify or remove any of the supplied entries in the file. Doing so adversely affects the operation of the cluster.

1. In a text editor, open the file `webgui_home_dir/etc/system/stores.lst`.
2. Add entries for any other directories that you want to include in the database. Specify all directories relative to the path: `webgui_home_dir/etc/configstore`.
3. Save the file and exit from the text editor.

The revised content of the file is copied to the database and propagated to all nodes in the cluster.

Related reference

“Load balancing best practices” on page 98

Cluster administration tools

Use the cluster tools to administer the members of the cluster.

Monitoring a load balancing cluster

If synchronized data fails to be committed to a node in the cluster, that node should be removed from the cluster for corrective action. Use the diagnosis tool to identify any unsynchronized nodes in the load balancing cluster.

To determine if changes to global data are not committed to any of the nodes, use the **HATool** command script to check the synchronization of modules and repositories on the nodes in a cluster. For the HATool, you must provide the DB2 administrator's credentials.

Query synchronization of modules

Use this command to determine if all nodes have identical sets of modules deployed.

```
HATool.bat/sh modules username password -byNodes -showAll
```

The following parameters are optional.

- **-byNodes**

Specifies that the results of the command are ordered by the node in the cluster. This parameter is optional. The default is to list the results by module.

- **-showAll**

Specifies that all modules and nodes in the cluster should be returned. This parameter is optional. The default is to return only modules for unsynchronized nodes.

Query the synchronization of global repositories

Use this command to determine if all repositories are synchronized on all nodes.

```
HATool.bat/sh repositories username password -byNodes -showAll
```

The following parameters are optional.

- **-byNodes**

Specifies that the results of the command are ordered by the node in the cluster. This parameter is optional. The default is to list the results by repository.

- **-showAll**

Specifies that all modules and nodes in the cluster should be returned. This parameter is optional. The default is to return only repositories for unsynchronized nodes.

Release the global lock

Use this command to manually release the global lock placed on all of the console nodes when the cluster is in maintenance mode. This command is used when a node cannot commit a change during synchronization and has to be taken offline.

```
HATool.bat/sh release-lock username password
```

Load balancing best practices

When administering the Web GUI in a load balancing environment, there are a number of practices you can use to avoid problems occurring in the cluster.

Overview

Administrative items that need special attention in a load balancing environment are:

- “Timed tasks”
- “The configuration database” on page 99
- “The list of files to maintain in the database” on page 99
- “Custom web content” on page 99

Timed tasks

Timed tasks are an essential element in the smooth running of a load balancing cluster. They ensure that all changes to files in *webgui_home_dir/etc/configstore/* and its subdirectories are detected and loaded into the server, without the need to restart the Web GUI server.

As a minimum, ensure that the **timedtasks.enabled** property in *theserver.init* file is set to true.

In most cases the schedules supplied for filters and views and for other components are adequate. However, you can change the schedules to suit your specific needs. If you do this in a load balancing environment, you are recommended to create identical schedules for the same set of components on all nodes in the load balancing cluster.

The configuration database

A load balancing cluster uses a database to hold the configuration data. Individual nodes in the cluster hold only a copy of this data, primarily for performance reasons. The master copy of the configuration data is always the one in the database.

Always make sure that the database is available before making any changes to the Web GUI configuration. This is especially important if you are modifying the configuration files directly, such as defining the metrics for the gauge page. Without the database, the node where you make the change is unable to put the change into the database and then propagate that to all other nodes. The result is an inconsistent configuration in the cluster. In extreme cases this could affect the performance of the cluster.

Where ever possible, use the Web GUI itself to change the configuration. The Web GUI always checks that the database is available before allowing you to save any changes to the configuration. This avoids many of the potential data inconsistencies that could arise were you to edit the files directly.

The list of files to maintain in the database

A cluster includes a file named `stores.lst` that holds a list of directories whose content is to be stored in the configuration database. As supplied, the file specifies all the directories that contain information which must be synchronized across a cluster. You can further directories to this list for other files that you want synchronized on all nodes.

Periodically check the `stores.lst` to ensure it is fully up to date. The file itself is one of those synchronized across all nodes. So any changes you make to it are automatically propagated to the other nodes.

Take care when editing `stores.lst` to change only those entries you have added. Do not remove any of the supplied entries, as this can adversely affect the operation of the cluster.

Custom web content

Place any custom web content, such as HTML file, in subdirectories of `tip_home_dir/profiles/TIPProfile/installedApps/TIPCell/isc.ear/OMNIBusWebGUI.war`. In addition, add these directories to the list of files to maintain in the database.

Related concepts

“The Web GUI in a load balancing environment” on page 93

“Overview of timed tasks” on page 8

Related tasks

“Administering timed tasks” on page 7

“Maintaining the list of files to monitor” on page 97

Troubleshooting

Use the troubleshooting notes to help correct problems with a load balancing cluster.

Resynchronizing a node with the cluster

In rare circumstances a fault may cause the configuration data of a node to become corrupted. Use this procedure to bring the data of a node back in synchronization with the cluster.

To resynchronize a node with the cluster restart the node.

Related tasks

“Restarting the server” on page 1

Recovering from a database corruption

An event such as a power failure may corrupt the content of the load-balancing database. If at least one node in the cluster still has valid configuration data, use this procedure to recover the database and the remaining nodes.

To recover the database and any nodes with corrupt configuration data, carry out the following on a node with valid configuration data:

1. Stop all nodes on the cluster.
2. Using a suitable SQL tool, access the load-balancing database, and issue the following commands:

```
DELETE FROM OMNIBUS_WEB_GUI.NODES;  
DELETE FROM OMNIBUS_WEB_GUI.NODES_CONFIG_ITEMS;  
DELETE FROM OMNIBUS_WEB_GUI.CONFIG_ITEMS;
```
3. Start a node that has valid configuration data. The node populates the database with its configuration data.
4. Start each of the other nodes in the cluster. Each node reads the configuration data from the database.

Related tasks

“Restarting the server” on page 1

Remotely administering the Web GUI server

To expedite complex configuration procedures and to instantly deploy predefined batch configuration settings to the Web GUI server, use the Web GUI Administration Application Program Interface (WAAPI) client.

The WAAPI client is a Java-based utility that you can use to remotely administer the Web GUI server.

Configuration instructions are written in XML and are stored in a text command file. The WAAPI client sends the command file directly to the Web GUI server which, after client authentication, updates accordingly.

Related concepts

“The Web GUI in a load balancing environment” on page 93

Related tasks

“Editing filter collections” on page 293

The Web GUI Administration API (WAAPI)

Read an introduction to WAAPI and how it communicates with the Web GUI server. Also read about the security mechanisms that WAAPI provides.

About WAAPI

WAAPI is an XML-based API that enables remote or local administration of the Web GUI server.

WAAPI enables administration of the server without having to use the graphical user interface of the Web GUI itself. It is particularly useful for carrying out bulk updates of information which would take a long time to achieve using the interactive facilities. For example, adding or modifying a number filters may be more efficient to do using WAAPI rather than the interactive facilities of the Web GUI.

The interface is installed with the Web GUI server enabling administration from the server itself. You can also install the WAAPI client on a remote server and manage the Web GUI from there.

Types of request

An administration command using WAAPI is known as a request. WAAPI allows you to manage many of the facilities available through the interactive interface. For example, you can add, modify, and delete filters.

There are also some features that are available only using WAAPI for specialized administration tasks. For example, you can reload all the system's filters and views.

WAAPI groups all the available requests into the following types:

- User
Modify the characteristics of any number of users, get a list of users, and remove Web GUI configuration data from users that no longer have active Web GUI roles.
- Views
Create, modify, delete, and list views.
- Maps
Create, modify, delete and list maps. In addition, you can add visuals to a map, modify and delete them.
- Resources
Resources are graphics files that contain items you want to appear on a map. Resources include background images for maps and graphics you want to use as map objects. You can add remove and list resources.
- Files
Add and remove directories and files on the Web GUI server.
- Menus
Create, modify, delete, and list menus to appear on Active Event Lists (AEL).
- Tools
Add, modify, and delete event management tools.
- Prompts

The Web GUI provides several types of prompt that event management tools can use to obtain information from the user. You can add, modify, delete, and list prompts.

- CGI scripts
Register, modify, and deregister CGI scripts.
- Filters
Add, modify, delete, and list filters. In addition, you can default view for a filter.
- Filter collections
Add, modify, delete, and list filter collections. In addition, you can add filters too and remove them from a filter collection.
- Metrics (gauges)
Create, modify, delete, and list metrics that the Web GUI displays in the form of gauges.
- Other
Miscellaneous requests that do not fit in any other category:
 - Resynchronize the Web GUI cache with the ObjectServer.
 - Remove a node from a load balancing cluster.
 - Generate a system status report.
 - Reload all filters and views.

Comparative procedures for modifying the Web GUI:

Most of the Web GUI components that you can modify in Tivoli Integrated Portal have an equivalent XML configuration instruction defined in the DTD.

The following examples describe the comparative procedures for modifying a field view in an Active Event List (AEL).

Web GUI procedure

The following example procedure describes how to modify a field view in an AEL through the Web GUI in Tivoli Integrated Portal:

1. Click **Administration > Event Management Tools > Views**.
2. From the **Views** list, select the view that you want to modify.
3. In the **Available Fields** list, select the field that you want to modify.
4. Change the justification values of "Title" and "Data" to **Left**.
5. Set the column width to **12**.
6. Click **Save**.

Equivalent WAAPI procedure

The following example procedure describes how to modify a field view in an AEL through the WAAPI client:

1. Create an XML command file in accordance with the rules of the WAAPI DTD.
For example:

```
<methodCall>
  <method methodName="view.modifyView">
    <view viewName="myview" acl="*">
      <columns>
        <visualEntry fieldName="myfield"
          fieldTitle="myfieldtitle"
          dataJustify="left">
```

```

        titleJustify="left"
        columnWidth="12"
    />
</columns>
</view>
</method>
</methodCall>

```

2. Start the WAAPI client and send the command file to the Web GUI server.

Communications between the WAAPI client and the Web GUI server

How the WAAPI client communicates with the Web GUI server.

Characteristics of the communication method

Communication between the WAAPI client and the Web GUI server has the following characteristics:

- Uses a request/response model.
- Is synchronous.
- Requests are in XML format.
- Responses are in text format.
- Uses an HTTP or HTTPS connection between the client and the server.

Communications overview

Whether you use the WAAPI client installed with the Web GUI server or a remote installation of the client, the way it communicates with the server is exactly the same:

1. The administrator creates one or more requests in one or more XML files.
2. The administrator runs the WAAPI client and supplies it with the XML files.
3. WAAPI sends the files to the server over a HTTP connection.

This connection can use SSL and can use encryption to help maintain the security of the data.

4. The server receives the requests and carries them out.
5. The server returns any output from the requests to the client over the same HTTP connection.
6. The WAAPI client receives the output. How it processes this depends on whether the administrator specified an output file to use when sending the requests.

If the administrator specified an output file, WAAPI sends the output to that file, creating it if necessary. Otherwise WAAPI sends the output to the screen where the administrator ran the client.

Security

The WAAPI client has a number of security features that help to protect the integrity of the data it exchanges with the Web GUI client.

WAAPI provides three ways you can use to protect the data it exchanges with the server:

- Securing the connection to the server
- Password encryption
- Protecting the WAAPI properties file

Secure Connections to the Web GUI server

In place of an unprotected HTTP connection, you can set up a secure connection with the Web GUI server using SSL. You can set up this connection in any of the following ways:

- Server-only authentication without FIPS 140-2
- Server and client authentication without FIPS 140-2
- Server-only authentication with FIPS 140-2
- Server and client authentication with FIPS 140-2

Password encryption

Independently of any secure connection you might use, WAAPI provides the means for encrypting the passwords that it uses. An unprotected HTTP connection can use AES password encryption. When using a secure connection, you can specify AES or FIPS 140-2 encryption. When the connection uses FIPS 140-2, only FIPS 140-2 password encryption is available.

Protecting the WAAPI properties file

The WAAPI properties file (`waapi.init`) contains a number of sensitive items of data. For example, it often holds the username and password of the administrative user on the server that runs WAAPI requests. It is important that this data is kept away from unauthorized users and is available only to administrators. So you can use the access control mechanisms of the operating system to limit access to the file.

Related reference

“WAAPI security” on page 194

Using WAAPI

How to set up the WAAPI properties file for your environment and to use the WAAPI client to send requests to the Web GUI server.

Setting up the WAAPI properties file

Use the WAAPI properties (`waapi.init`) to match your environment.

Set the properties in the WAAPI initialization file to match your environment. Set values for those properties whose value changes rarely such as:

- `waapi.host`
- `waapi.port`
- `waapi.user`
- `waapi.password`
- The secure connection and encryption properties

This minimizes the number of options that you need to enter on the command line.

Note: If you set the `waapi.user` and `waapi.password` properties, make sure that the properties file is protected against access for users other than authorized administrators.

If you manage several Web GUI servers from one location, you can have multiple properties files, one for each server. This enables you to tailor the properties for each server.

You can also use the properties file to hold default values for properties. You can always override any property setting using a command line option.

Sending requests to the server

To send requests to the Web GUI server, prepare the WAAPI command file that contains the requests and run the WAAPI client to send it to the server.

Preparing the WAAPI command file:

To prepare the WAAPI command file:

1. Create an 8-bit Unicode Transformation format (UTF) text file with a `.xml` suffix.
2. Follow the guidance in “WAAPI requests” on page 106 to add the necessary requests to the file.

Tip: You can use the sample WAAPI files provided with the client as templates for your command file.

3. Save the file using UTF-8 to a suitable directory.

Running the WAAPI client:

To run the client and send your file to the server:

Enter the following command, dependent on the operating system you use:

- **Linux** **UNIX** `webgui_home_dir/waapi/bin/runwaapi options -file waapi_command_file`
- **Windows** `webgui_home_dir\waapi\bin\runwaapi.cmd options -file waapi_command_file`

Replace:

`web_gui_home_dir`

with the installation directory of the Web GUI. If you are running the client on a remote system, specify the location where you installed the client.

options

With any additional command line options that you require. Commonly used options are:

Table 15. Frequently used WAAPI command line options

Function	Option
Redirect output to a file	<code>-outfile <i>filepath</i></code> Replace <i>filepath</i> with the path of the file to receive output from the WAAPI command file.
Specifying a username and password to run the command file under	<code>-user <i>username</i> -password <i>password</i></code> Replace <i>username</i> with the name of the account to use, and <i>password</i> with the password for that account. Note: If you use these options make sure you clear the screen and the command history to ensure the credentials remain secure.
Using an alternative properties file	<code>-props <i>propsfile</i></code> Replace <i>propsfile</i> with the path name of the WAAPI properties file to use.

waapi_command_file

with the name of your WAAPI command file.

Example

For example, on a Windows system:

```
c:\ibm\tivoli\netcool\omnibus_webgui\waapi\bin\runwaapi.cmd -file newFilters.xml
```

WAAPI requests

A WAAPI request is an XML document that contains instructions to administer the Web GUI server.

You can administer the following items using WAAPI requests:

- “User requests” on page 110
- “View requests” on page 118
- “Map requests” on page 124
- “Resource requests” on page 143
- “File requests” on page 146
- “Menu requests” on page 149
- “Tool requests” on page 153
- “Prompt requests” on page 164
- “CGI requests” on page 172
- “Filter requests” on page 174
- “Filter collection requests” on page 180
- “Metric requests” on page 185
- “Other requests” on page 192

Note: The names of some attributes are long. In the following syntax descriptions, these long names are divided over 2 or more lines. However, in the XML you produce, enter each attribute name as a single character sequence without any line breaks.

Structure of a WAAPI request

A WAAPI request is an XML document that contains an optional XML declaration followed by the `<methodCall>` root element. A request has a number of additional characteristics that you need to bear in mind.

The XML declaration:

Optionally, you can begin a WAAPI request with an XML declaration. For this release of the Web GUI the declaration is:

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

The `<methodCall>` root element:

The root element holds the content of the request. For tool, prompt, and metric requests, the root element also defines a namespace.

Basic form of the root element

The basic form of the root element is:

```
<methodCall>

</methodCall>
```

The root element for tool, prompt, and metric requests

For tool, prompt, and metric requests, the form of the root element is:

```
<methodCall xmlns:type=namespace-url>

</methodCall>
```

Here, *type* is the type of request (tool, prompt, or metric) and *namespace-url* is the fully qualified URL for the namespace.

The namespace URLs for prompt, tool, and metric requests are as follows:

Table 16. Namespace URLs

Type of request	Namespace URL
Tool	"http://www.ibm.com/tivoli/netcool/webtop/tools/2.1"
Prompt	"http://www.ibm.com/tivoli/netcool/webtop/prompts/2.2"
Metric	"http://www.ibm.com/tivoli/netcool/webtop/metrics/7.3.1"

Content of the root element

The root element, whatever form it takes, contains one or more `<method>` elements. Each of these elements contains a request to manipulate an item of Web GUI data. Later sections set out the format and content of the `<method>` element for each type of data that WAAPI can operate on. The `<method>` element contains a `methodName`

attribute that defines the type of data item and the operation to perform on that data. Within the <method> element are child elements that define the item of data.

For example, when the methodCall attribute has the value view.createView the <method> element contains one or more <view> elements each of which defines the characteristics of a view to add to the collection of Web GUI views.

The root element can contain method calls for any mix of Web GUI data items. When the element contains any combination of tool, prompt, and metric requests, include each of the corresponding xmlns attributes in the <methodCall> element.

For example, if a <methodCall> element contains <method> elements for views, tools, and prompts, specify the <methodCall> element as follows:

```
<methodCall xmlns="http://www.ibm.com/tivoli/netcool/webtop/tools/2.1"
  xmlns="http://www.ibm.com/tivoli/netcool/webtop/prompts/2.2" >

  <!-- <method> elements appear here -->

</methodCall>
```

Characteristics of WAAPI XML documents:

XML documents that contain WAAPI requests have a number of characteristics that you need to bear in mind:

- “Document Type definition (DTD)”
- “The order of XML elements”
- “Case sensitivity” on page 109
- “Content and value restrictions” on page 109
- “Comments” on page 109

Document Type definition (DTD):

Each WAAPI request must be well-formed and is validated against the document type definition (DTD) by the WAAPI client. The DTD defines a set of rules for the syntax of elements that appear in the request.

The DTD defines the statements that you can put in the XML request, the order in which elements must appear, which elements can be nested, which elements have attributes, and so forth. The WAAPI DTD is in the following location:

webgui_home_dir/waapi/etc/waapi.dtd

The order of XML elements:

In general, the WAAPI DTD is order tolerant. Providing that the request files are well formed, the DTD allows you to parse the majority of child statements. However, some elements required that you put child elements in a particular order.

An example of an order-tolerant element is <supermenu>. The <supermenu> element can contain three types of child element: <tool>, <separator>, and <menu>. These elements can appear in any order within the <supermenu> element hierarchy. The order in which elements are placed in the appearance of the AEL tool menu that is created by these instructions.

On the other hand, the child elements of the <method> and <view> elements must appear in the correct order. The following table defines the order in which child elements for the <method> and <view> must appear.

Table 17. Child element order

Element	Child element order
method	<ol style="list-style-type: none">1. user2. view3. map4. resources5. supermenu6. entityview7. entitygroup8. file9. session10. metric
view	<ol style="list-style-type: none">1. columns2. sorting

Case sensitivity:

In XML documents, element and attribute names are case sensitive. Use the same case as specified in the method definitions. In addition, the content of some elements and the value of some attributes are also case sensitive. The descriptions of element content and attribute values in the method definitions includes information on case sensitivity. In particular be sure to specify enumerated values exactly as they appear in the method definition.

Content and value restrictions:

There are restrictions on the content of some elements and the value of some attributes. For example, there may be a limit on the number of characters that an attribute value can have. The description of elements and attributes in each method definition define any content restrictions that apply.

Comments:

You can include comments in the WAAPI request using the standard XML syntax. Begin the comment with the syntax <!-- and terminate the comment with the syntax -->. Put the comment between the beginning and terminating syntax. A comment can be on a single line or multiple lines.

For example:

```
<!-- This is a comment -->

<!--
  This is a comment that has more
  than one line.
-->
```

Sample Requests:

There are a wide range of sample requests supplied with the Web GUI.

The installation of the Web GUI server includes set of example requests. You can use these as models for your requests. The examples are in the following directory:

`webgui_home_dir/waapi/etc/samples`

Here, `webgui_home_dir` is the installation directory of the Web GUI. For example, `ibm/tivoli/netcool/omnibus_webgui`.

User requests

User requests operate on Web GUI users. There are functions to modify a user, get a list of users, and remove configuration information for users that do not have Web GUI user privileges.

WAAPI provides three methods for working on users defined in the system:

- “Modify a user”
- “Get a list of users” on page 118
- “Maintain users” on page 118

Modify a user:

The format of the `<method>` element for modifying a user is:

```
<method methodName="user.modifyUser">
```

Use this method call to change any number of the following characteristics of a user:

- Default filter
- Home page
- AEL characteristics, including:
 - Access to the AEL
 - Default and minimum refresh time
 - Items to display in the AEL

The `<method>` element contains one or more `<user>` elements each of which identifies a user and the characteristics of the user that are to be modified. Include only attributes of the `<user>` that correspond to the characteristics you want to change. When you omit an attribute the corresponding characteristic is unchanged.

`<user>`

The `<user>` element defines the characteristics of a user and has the following attributes:

Table 18. Attributes of the `<user>` element

Attribute name	Required or optional	Description
name	Required	Identifies the user to modify. Value: The username of a Web GUI user. Default value: None.

Table 18. Attributes of the <user> element (continued)

Attribute name	Required or optional	Description
filter	Optional	Defines the event severity levels that appear in the user's AEL. Value: String Default value: None.
homepage	Optional	The URL of the user's home page, relative to the context root of the Web GUI. Value: The name and location of the home page. Default value: /index.html
ael_user	Optional	Specifies whether the user can access the Active Event List (AEL). Value: true or false Default value: true
ael_user_properties_allow_select	Optional	Specifies whether the user can set their own preferences in the ACL. Value: true or false Default value: true
ael_user_properties_allow_custom_refresh	Optional	Specifies whether the user can refresh the AEL display. The attribute has one of the following values: Value: true or false Default value: false
ael_user_properties_refresh_time	Optional	Specifies the refresh time (in seconds) of the AEL for this user. Value: Integer Default value: 60
ael_user_properties_minimum_refresh_time	Optional	Specifies the minimum refresh time (in seconds) that the user can specify. The value of this attribute must be less than or equal to the value of ael_user_properties_refresh_time. Value: Integer. Default value: 30
ael_user_properties_show_colors	Optional	Specifies whether colors are used in the user's AEL. Value: true or false Default value: true
ael_user_properties_show_info	Optional	Specifies whether the Alerts menu of the AEL includes the option to display the Information window. Value: true or false Default value: true

Table 18. Attributes of the <user> element (continued)

Attribute name	Required or optional	Description
ael_user_properties_show_journal	Optional	Specifies whether the Information window for an alert includes the Journal tab. Value: true or false Default value: true
ael_user_properties_show_details	Optional	Specifies whether the Information window for an alert includes the Details tab. Value: true or false Default value: true
ael_user_properties_show_menubar	Optional	Specifies whether the AEL portlet includes a menu bar. Value: true or false Default value: false
ael_user_properties_showgraphicconversions	Optional	Specifies whether the AEL displays severity icons. Value: true or false Default value: false
ael_user_properties_show_preferences	Optional	Specifies whether the user can access the Preferences window from the AEL Edit menu or the Preferences icon on the toolbar. Value: true or false Default value: true
ael_user_properties_monitor_show_number	Optional	Specifies whether the total number of events appears on monitor boxes in the AEL. Value: true or false Default value: true
ael_user-properties_monitor_show_highest	Optional	Specifies whether the highest severity level appears on monitor boxes in the AEL. Value: true or false Default value: true
ael_user_properties_monitor_show_highest_color	Optional	Specifies whether the color of the highest severity level appears on monitor boxes in the AEL. Value: true or false Default value: true
ael_user_properties_monitor_show_lowest	Optional	Specifies whether the lowest severity level appears on monitor boxes in the AEL. Value: true or false Default value: true

Table 18. Attributes of the <user> element (continued)

Attribute name	Required or optional	Description
ael_user_properties_monitor_show_lowest_color	Optional	Specifies whether the color of the lowest severity level appears on monitor boxes in the AEL. Value: true or false Default value: false
ael_user_properties_monitor_show_border	Optional	Specifies whether the border around monitor boxes in the Event Dashboard, or in an AEL displayed in monitor box style using SmartPage commands, has the same color as the maximum severity displayed in the box or AEL. Value: true or false Default value: true
ael_user_properties_monitor_show_metric	Optional	Specifies whether the metric appears on monitor boxes in the AEL. Value: true or false Default value: true
ael_user_properties_monitor_distribution_meter	Optional	Specifies the type of distribution meter to use for representing alerts on the user's AEL monitor boxes. Value: histogram, lavalamp, or none. Default value: histogram
ael_user_properties_flash_time	Optional	Specifies the flash time in milliseconds for alerts in the AEL. This attribute is effective only when the value of ael_user_properties_flash_enabled is true. Value: Integer Default value: 400
ael_user_properties_flash_brightness	Optional	Specifies the flash brightness of alerts in the AEL. This attribute is effective only when the value of ael_user_properties_flash_enabled is true. Value: Integer Default value: 0
ael_user_properties_flash_enabled	Optional	Specifies whether unacknowledged events in the AEL flash. Value: true or false Default value: false
ael_user_properties_show_summarybar	Optional	Specifies whether the summary bar appears at the foot of the AEL. Value: true or false Default value: true

Table 18. Attributes of the <user> element (continued)

Attribute name	Required or optional	Description
ael_user_properties_show_toolbar	Optional	Specifies whether the toolbar appears above the AEL. Value: true or false Default value: true
ael_user_properties_monitor_font_name	Optional	Specifies the name of the font to use for text on monitor boxes in the AEL. Value: String Default value: Dialog
ael_user_properties_monitor_font_size	Optional	Specifies the size of the font used on monitor boxes in the AEL. Value: Integer Default value: 12
ael_user_properties_timeformat	Optional	Specifies the format for the date and time as it appears on the AEL. Value: short, long, or a date and time format. Default value: short
ael_user_properties_eventlist_font_name	Optional	Specifies the name of the font to use on the AEL. Value: String Default value: Dialog
ael_user_properties_eventlist_font_size	Optional	Specifies the size of the font to use on the AEL. Value: Integer Default value: 12
ael_user_properties_eventlist_width	Optional	Specifies the total width, in pixels, of the AEL. Value: Integer Default value: 600
ael_user_properties_eventlist_height	Optional	Specifies the total height, in pixels, of the AEL. Value: Integer Default value: 450
ael_user_properties_notify_enabled	Optional	Specifies whether notifications (for example, sounds) occur when events are added to or modified on the AEL. Value: true or false Default value: false
ael_user_properties_notify_when_iconized	Optional	When the user has iconized the AEL, specifies whether to redisplay it when a new event arrives. Value: true or false Default value: true

Table 18. Attributes of the <user> element (continued)

Attribute name	Required or optional	Description
ael_user_properties_notify_always	Optional	Specifies whether the user receives a notification of the arrival of an event irrespective of whether the AEL is open, closed, or iconized. Value: true or false Default value: true
ael_user_properties_notify_insert	Optional	Specifies whether the user receives a notification when an event is added to the AEL. Value: true or false Default value: false
ael_user_properties_notify_delete	Optional	Specifies whether the user receives a notification when an event is removed from the AEL. Value: true or false Default value: true
ael_user_properties_notify_update	Optional	Specifies whether the user receives a notification when any event in the AEL is updated. Value: true or false Default value: false
ael_user_properties_notify_play_sound	Optional	Specifies whether a notification to use includes playing a sound. Value: true or false Default value: true
ael_user_properties_notify_sound_url	Optional	Specifies the URL of a sound to play as a notification. Value: String Default value: none
ael_user_properties_notify_flash_icon	Optional	Specifies whether the AEL icon flashes as part of a notification. Value: true or false Default value: true
ael_user_properties_notify_open_window	Optional	Specifies whether a window opens as part of a notification. Value: true or false Default value: false
ael_user_properties_notify_open_url	Optional	Specifies whether to open a URL as part of a notification. Value: true or false Default value: false

Table 18. Attributes of the <user> element (continued)

Attribute name	Required or optional	Description
ael_user_properties_notify_url	Optional	Specifies the URL to open when the value of ael_user_properties_notify_open_url is true. Value: String Default value: none
ael_user_properties_notify_url_target	Optional	Specifies the target in the browser where the URL specified in ael_user_notify_url opens. Where frames have been defined on the HTML you can specify the target as the name of a frame (for example, UpperFrame). Value: _blank, the name of a frame Default value: none
ael_user_properties_monitor_num_cols	Optional	Specifies the number of columns of monitor box applets to display on the AEL. Value: Integer Default value: 4
ael_user_properties_allow_journal_edit	Optional	Specifies whether the user can edit the journals associated with the AEL. Value: true or false Default value: true
ael_user_properties_allow_filter_builder_access	Optional	Specifies whether the user can access the Filter Builder from the AEL. Value: true or false Default value: false
ale_user_properties_allow_view_builder_access	Optional	Specifies whether the user can access the View Builder from the AEL. Value: true or false Default value: true
ael_user_properties_allow_views_and_filters_use	Optional	Specifies whether the user can select pre-defined filters and views in the AEL. Value: true or false Default value: true
map_editor_user_properties_show_grid	Optional	Specifies whether a grid appears in the Map Editor. Value: true or false Default value: true
map_editor_user_properties_snap_to_grid	Optional	Specifies whether objects in the Map Editor snap to the grid. Value: true or false Default value: false

Table 18. Attributes of the <user> element (continued)

Attribute name	Required or optional	Description
map_editor_user_properties_grid_size	Optional	Specifies the size of the grid in the Map Editor. Value: Integer Default value: 5
map_editor_user_properties_editor_width	Optional	Specifies the total width, in pixels, of the display area in the Map Editor. Value: Integer Default value: 600
map_editor_user_properties_editor_height	Optional	Specifies the total height, in pixels, of the display area in the Map Editor. Value: Integer Default value: 400

Example

This example modifies the user named user1, and defines the following characteristics:

- The default filter shows all severities.
- The following AEL characteristics are modified:
 - Refresh time is 90 seconds
 - The minimum refresh is 70 seconds
 - The user can access the AEL Preferences window
 - The user cannot force a refresh of the AEL
 - The user can set their own preferences
 - The AEL Information window includes the **Details** and **Journal** tabs
 - The **Alerts** menu includes an option to open the Information window.

```
<methodCall>
  <method methodName="user.modifyUser">
    <user name="user1"
      filter="Severity>=0"
      ael_user_properties_refresh_time="90"
      ael_user_properties_minimum_refresh_time="70"
      ael_user_properties_show_preferences="true"
      ael_user_properties_allow_custom_refresh="false"
      ael_user_properties_allow_select="true"
      ael_user_properties_show_details="true"
      ael_user_properties_show_info="false"
      ael_user_properties_show_journal="true">
    </user>
  </method>
</methodCall>
```

Related tasks

“Modifying the AEL date and time format” on page 246

Get a list of users:

The format of the <method> element for getting a list of users is:

```
<method methodName="user.getList" />
```

Use this method to obtain a list containing the User IDs of all users defined in the Web GUI that have either the ncw_user or ncw_admin roles.

Example

```
<methodCall>  
  <method methodName="user.getList" />  
</methodCall>
```

Maintain users:

The format of the <method> element for maintaining users is:

```
<method methodName="user.maintainUsers" />
```

Use this method to remove Web GUI configuration data from all users that no longer have the ncw_user or ncw_admin roles. The configuration data includes:

- Preferences
- Filter definitions
- View definitions

Example

```
<methodCall>  
  <method methodName="user.maintainUsers" />  
</methodCall>
```

View requests

View requests operate on Web GUI views. There are functions to create, modify, and delete views. In addition you can obtain a list of the views defined on the Web GUI server.

WAAPI provides five methods for working with views:

- "Create a view"
- "Create or replace a view" on page 122
- "Modify a view" on page 123
- "Delete a view" on page 124
- "Get a list of views" on page 124

Create a view:

The format of the <method> element for creating a view is:

```
<method methodName="view.create">
```

Use this method to create a new view for use on the AEL, LEL, Table View, and Event Dashboard. The <method> element contains one or more <view> elements each of which defines the characteristics of a new view. The <view> element can contain up to one of each of the <columns> and <sorting> elements.

`<view>`:

The `<view>` element defines a view and has the following attributes:

Table 19. Attributes of the `<view>` element

Attribute name	Required or optional	Description
viewName	Required	Provides a unique name for a view. Value: String Default value: None
datasource	Optional	The name of the data source that provides events for the view. To specify multiple data sources use a comma-separated list. Value: String Default value: NCOMS
user	Optional	For user views, this attribute identifies the users that the view is associated with. The value of the attribute is a comma-separated list of User IDs. Value: List of user IDs Default value: None
type	Optional	The type of view. Value: global, system, or user Default value: None

`<columns>`:

The `<columns>` element is a child the `<view>` element and can occur once or not at all. If present, the element contains any number of `<visualEntry>` elements.

`<visualEntry>`:

The `<visualEntry>` element is a child of the `<columns>` element that defines the appearance of fields in the view. The element has the following attributes:

Table 20. Attributes of the `<visualEntry>` element

Attribute name	Required or optional	Description
fieldName	Required	The name of a field to include in the view. This is analogous to an entry from the Available fields list in the View Builder. Value: The name of field from a ObjectServer table. Default value: None
fieldTitle	Required	The name to use for the column title in the view for this field. Value: String Default value: None

Table 20. Attributes of the <visualEntry> element (continued)

Attribute name	Required or optional	Description
dataJustify	Required	Specifies how to justify the text in the column. Value: centre, left, or right. Default value: left
titleJustify	Required	Specifies how to justify the column title. Value: centre, left, or right Default value: left
columnWidth	Required	The width of the column, in pixels. Value: Integer Default value: 25
columnLocked	Optional	Specifies whether the position of a column is locked or whether the user can rearrange the order of the column. Value: true or false Default value: false
datasource	Optional	The name of the datasource that provides data for the field. Value: Name of a data source Default value: NCOMS

<sorting>:

The <sorting> element is a child of the <view> element and can occur once or not at all. If present, the element contains any number of <sortColumn> elements that define the sorting order for events in the view.

<sortColumn>:

The sortColumn element defines a field to use when sorting the list events in a view. When there are multiple <sortColumn> the entries in the view are sorted in the order that the <sortColumn> elements appear.

The element has the following attributes:

Table 21. Attributes of the <sortColumn> element

Attribute name	Required or optional	Description
fieldName	Required	The name of a field in the view. Value: The name of a field. Default value: None.
order	Required	The order to sort the column entries. Value: asc (for ascending order) or desc (for descending order). Default value: asc

Table 21. Attributes of the <sortColumn> element (continued)

Attribute name	Required or optional	Description
datasource	Optional	The name of the data source that provides the named field. Value: The name of a data source. Default value: NCOMS

Example:

The following example creates a user view named SeveritySummary that has the following characteristics:

- The view includes the following fields formatted as shown:

Table 22. Create view example: Fields and their formatting

Field	Field title	Data justify	Title justify	Column width
Severity	Sev	Center	Center	5
Acknowledged	Ack	Center	Center	3
Node	Node	Left	Left	12
AlertGroup	Alert Group	Left	Left	10
Summary	Summary	Left	Left	40
LastOccurrence	Last Occurrence	Left	Left	14

- The columns for the Severity and Acknowledged columns are locked.
- The view uses the default data source.
- The view sorts entries in descending order of severity.
- The view is available to the ncoadmin and tipadmin users.

```
<methodCall>
<method methodName="createView">
  <view viewName="SeveritySummary"
    user="ncoadmin,tipadmin"
    type="user">
    <columns>
      <visualEntry fieldName="Severity"
        fieldTitle="Sev"
        dataJustify="centre"
        titleJustify="center"
        columnWidth="5"
        columnLocked="true" />
      <visualEntry fieldName="Acknowledged"
        fieldTitle="Ack"
        dataJustify="centre"
        titleJustify="center"
        columnWidth="3"
        columnLocked="true" />
      <visualEntry fieldName="Node"
        fieldTitle="Node"
        dataJustify="left"
        titleJustify="left"
        columnWidth="12" />
      <visualEntry fieldName="AlertGroup"
        fieldTitle="Alert Group"
        dataJustify="left"
        titleJustify="left"
        columnWidth="10" />
    </columns>
  </view>
</method>
</methodCall>
```

```

        <visualEntry fieldName="Summary"
          fieldTitle="Summary"
          dataJustify="left"
          titleJustify="left"
          columnWidth="40" />
        <visualEntry fieldName="LastOccurrence"
          fieldTitle="Last Occurrence"
          dataJustify="left"
          titleJustify="left"
          columnWidth="14" />
      </columns>
    <sorting>
      <sortColumn fieldName="Severity" order="desc" />
    </sorting>
  </view>
</method>
</methodCall>

```

Create or replace a view:

The format of the <method> element for creating or replacing a view is:

```
<method methodName="createOrReplaceView">
```

Use this to replace an existing view or create a new one if it does not already exist. The <method> element contains one or more <view> elements each of which defines the characteristics of a view. Each <view> element can contain up to one <columns> element and up to one <sorting> element. If present, the <columns> element contains any number of <visualEntry> elements and, if present, each <sorting> element contains any number of <sortColumn> elements.

Related reference

“<view>” on page 119

“<columns>” on page 119

“<sorting>” on page 120

Example:

This example creates or replaces a view named OccurrenceSummary that summarizes the last occurrence of alerts. It contains four columns: Severity, Node, Summary, and LastOccurrence. These columns are formatted in the same way as the corresponding columns in the example of creating a view.

```

<methodCall>
  <method methodName="createOrReplaceView">
    <view viewName="OccurrenceSummary"
      user="ncoadmin,tipadmin"
      type="user">
        <columns>
          <visualEntry fieldName="Severity"
            fieldTitle="Sev"
            dataJustify="centre"
            titleJustify="center"
            columnWidth="5"
            columnLocked="true" />
          <visualEntry fieldName="Node"
            fieldTitle="Node"
            dataJustify="left"
            titleJustify="left"
            columnWidth="12" />
          <visualEntry fieldName="Summary"
            fieldTitle="Summary"
            dataJustify="left"
            titleJustify="left"

```



```

        columnWidth="40" />
<visualEntry fieldName="LastOccurrence"
  fieldTitle="Last Occurrence"
  dataJustify="left"
  titleJustify="left"
  columnWidth="14" />
</columns>
<sorting>
  <sortColumn fieldName="LastOccurrence" order="desc" />
</sorting>
</view>
</method>
</methodCall>

```

Modify a view:

The format of the <method> element for modifying a view is:

```
<method methodName="view.Modify">
```

Use this method to modify the characteristics of an existing view. Include either or both of the <columns> and <sorting> elements with their child elements as required. For example, to change the sorting method of the view, include the <sorting> element with the <sortColumn> elements necessary to define the sorting method.

Example

This example modifies the view named view1 and makes the following changes:

- The view includes two fields:

Node

Serial

- The view sorts entries on ascending order of the Severity field.

```

<methodCall>
<method methodName="view.modifyView">
  <view viewName="view1">
    <columns>
      <visualEntry fieldName="Node"
        fieldTitle="Node"
        dataJustify="left"
        titleJustify="left"
        columnWidth="18" />
      <visualEntry fieldName="Serial"
        fieldTitle="Serial"
        dataJustify="left"
        titleJustify="left"
        columnWidth="12" />
    </columns>
    <sorting>
      <sortColumn fieldName="Severity" order="asc" />
    </sorting>
  </view>
</method>
</methodCall>

```

Related reference

“<view>” on page 119

“<columns>” on page 119

“<sorting>” on page 120

Delete a view:

The format of the <method> element for deleting a view is:

```
<method methodName="view.delete">
```

The <method> element contains one or more <view> elements each of which identifies a view to delete. In the view element, include only the viewName attribute.

Example

This example deletes the view named viewsample2.

```
<methodCall>  
  <method methodName="view.deleteView">  
    <view viewName="viewsample2">  
    </view>  
  </method>  
</methodCall>
```

Related reference

"<view>" on page 119

Get a list of views:

The format of the <method> element for getting a list of views is:

```
<method methodName="view.getList">
```

The method returns a list of the names of all views defined in the Web GUI. There is a separate section in the listing for each type of view.

Example

```
<methodCall>  
  <method methodName="view.getList">  
  </method>  
</methodCall>
```

Map requests

Maps provide a visual means of viewing events and their locations. There are functions to create, modify, and delete maps and map visuals. In addition, you can obtain a list of the maps defined on the server.

WAAPI provides five methods for working with maps and four methods for working with map visuals:

- "Create a map" on page 125
- "Create or replace a map" on page 139
- "Modify a map" on page 139
- "Delete a map" on page 140
- "Get a list of maps" on page 140
- "Add a map visual" on page 141
- "Add or replace a map visual" on page 141
- "Modify a map visual" on page 142
- "Delete a map visual" on page 143

Create a map:

The format of the `<method>` element for creating a map is:

```
<method methodName="map.createMap">
```

Use this method to create a new map. The `<method>` element contains one or more `<map>` elements each of which defines the characteristics of a new map. The `<map>` element contains any number of the `<text>`, `<button>`, `<monitor>`, `<line>`, and `<icon>` elements.

`<map>`:

The `<map>` element defines a map and has the following attributes:

Table 23. Attributes of the `<map>` element

Attribute name	Required or optional	Description
name	Required	Provides a unique name for a map. Value: String Default value: None
bgImage	Optional	The path of an image to use as the background for the map. Value: String Default value: None
bgColor	Optional	The name of a color to use as the background for the map. Value: String Default value: None
h	Optional	The height of the map in pixels. Value: Integer Default value: None
w	Optional	The width of the map in pixels. Value: Integer Default value: None

`<text>`:

The `<text>` element is a child of the `<map>` element and defines the characteristics of a text label on a map. The element can occur any number of times and, if present, has the following attributes:

Table 24. Attributes of the `<text>` element

Attribute name	Required or optional	Description
filter	Optional	Defines a filter to associate with the field. Value: String Default value: None

Table 24. Attributes of the <text> element (continued)

Attribute name	Required or optional	Description
filterType	Optional	The type of the filter defined in the filter attribute. Value: global or system Default value: None
name	Required	The name of the text label. Value: String Default value: None
label	Optional	The text to appear on the label. Value: String Default value: None
datasource	Optional	The name of the data source to associate with the label. To specify more than one data source, use a comma-separated list. Value: String Default value: NCOMS
x	Optional	The horizontal position of the label in pixels from the left of the map Value: Integer Default value: None
y	Optional	The vertical position of the label in pixels from the bottom of the map. Value: Integer Default value: None
translucency	Optional	The level of translucency for the label. Value: Integer between 0 and 100 (inclusive). Default value: None
rotate	Optional	The angle in degrees to rotate the label. Value: Integer between 0 and 360 (inclusive) Default value: None
show_shadow	Optional	Specifies whether the label appears on the map with a shadow. Value: true or false Default value: false
font	Optional	The name of the font to use for the label. Value: String Default value: None

Table 24. Attributes of the <text> element (continued)

Attribute name	Required or optional	Description
size	Optional	The point size of the font to use for the label. Value: Integer Default value: None
justify	Optional	Defines alignment of the text in the label. Value: center, left, or right Default value: None
style	Optional	The text style (for example, bold) for the label. Value: String Default value: None
color	Optional	The name of the color to use for the text. Value: String Default value: None
action	Optional	The action that takes place when the user clicks on the text label. Value: String Default value: None
url	Optional	The uniform resource locator that is the action target for the text label. Value: String Default value: None
target	Optional	The target window in the browser for output from the value of the url attribute. When the attribute is not present, output replaces the content of the current browser window. Value: String Default value: _blank
flash	Optional	Defines whether the text label flashes. Value: true or false Default value: false

<button>:

The *<button>* element is a child of the *<map>* element and defines the characteristics of a button on a map. The element can occur any number of times and, if present, has the following attributes:

Table 25. Attributes of the <button> element

Attribute name	Required or optional	Description
filter	Optional	Defines a filter to associate with the button. Value: String Default value: None
filterType	Optional	The type of the filter defined in the filter attribute. Value: global or system Default value: None
name	Required	The name of the button. Value: String Default value: None
label	Optional	The text to appear on the button. Value: String Default value: None
datasource	Optional	The name of the data source to associate with the button. To specify more than one data source, use a comma-separated list. Value: String Default value: NCOMS
x	Optional	The horizontal position of the button in pixels from the left of the map Value: Integer Default value: None
y	Optional	The vertical position of the button in pixels from the bottom of the map. Value: Integer Default value: None
translucency	Optional	The level of translucency for the button. Value: Integer between 0 and 100 (inclusive). Default value: None
show_shadow	Optional	Specifies whether the button appears on the map with a shadow. Value: true or false Default value: false

Table 25. Attributes of the <button> element (continued)

Attribute name	Required or optional	Description
font	Optional	The name of the font to use for the button text. Value: String Default value: None
size	Optional	The point size of the font to use for the button text. Value: Integer Default value: None
font_color	Optional	The name of the color to use for the button text. Value: String Default value: None
style	Optional	The text style (for example, bold) for the button text. Value: String Default value: None
color	Optional	The name of the color to use for the button. Value: String Default value: None
action	Optional	The action that takes place when the user clicks on the button. Value: String Default value: None
url	Optional	The uniform resource locator that is the action target for the button. Value: String Default value: None
target	Optional	The target window in the browser for output from the value of the url attribute. When the attribute is not present, output replaces the content of the current browser window. Value: String Default value: _blank
flash	Optional	Defines whether the button flashes. Value: true or false Default value: false
w	Optional	The width of the button in pixels. Value: Integer Default value: None

Table 25. Attributes of the <button> element (continued)

Attribute name	Required or optional	Description
h	Optional	The height of the button in pixels. Value: Integer Default value: None
type	Optional	The type of button, for example a rectangle. Value: rectangle, rounded, or ellipse Default value: rectangle
arc_diameter	Optional	The arc diameter of the button, in degrees. Value: Integer Default value: None
transparent	Optional	Defines whether the button is transparent. Value: true or false Default value: false
legend	Optional	The legend for the button. Value: String Default value: None

<monitor>:

The <monitor> element is a child of the <map> element and defines the characteristics of a monitor box on a map. The element can occur any number of times and, if present, has the following attributes:

Table 26. Attributes of the <monitor> element

Attribute name	Required or optional	Description
filter	Optional	Defines a filter to associate with the monitor box. Value: String Default value: None
filterType	Optional	The type of the filter defined in the filter attribute. Value: global or system Default value: None
name	Required	The name of the monitor box. Value: String Default value: None
label	Optional	The text to appear on the monitor box. Value: String Default value: None

Table 26. Attributes of the <monitor> element (continued)

Attribute name	Required or optional	Description
datasource	Optional	The name of the data source to associate with the monitor box. To specify more than one data source, use a comma-separated list. Value: String Default value: NCOMS
x	Optional	The horizontal position of the monitor box in pixels from the left of the map Value: Integer Default value: None
y	Optional	The vertical position of the monitor box in pixels from the bottom of the map. Value: Integer Default value: None
translucency	Optional	The level of translucency for the monitor box. Value: Integer between 0 and 100 (inclusive). Default value: None
show_shadow	Optional	Specifies whether the monitor box appears on the map with a shadow. Value: true or false Default value: false
font	Optional	The name of the font to use for the monitor box text. Value: String Default value: None
size	Optional	The point size of the font to use for the monitor box text. Value: Integer Default value: None
style	Optional	The text style (for example, bold) for the monitor box text. Value: String Default value: None
color	Optional	The name of the color to use for the button. Value: String Default value: None
action	Optional	The action that takes place when the user clicks on the monitor box. Value: String Default value: None

Table 26. Attributes of the <monitor> element (continued)

Attribute name	Required or optional	Description
url	Optional	The uniform resource locator that is the action target for the monitor box. Value: String Default value: None
target	Optional	The target window in the browser for output from the value of the url attribute. When the attribute is not present, output replaces the content of the current browser window. Value: String Default value: _blank
flash	Optional	Defines whether the button flashes. Value: true or false Default value: false
w	Optional	The width of the monitor box in pixels. Value: Integer Default value: None
h	Optional	The height of the monitor box in pixels. Value: Integer Default value: None
type	Optional	The type of monitor box, for example a histogram. Value: lavalamp or histogram Default value: histogram
flash	Optional	Defines whether the monitor box flashes. Value: true or false Default value: false
show_label	Optional	Defines whether to display a label on the monitor box. Value: true or false Default value: false
show_total	Optional	Defines whether to display the total number of alerts on the monitor box. Value: true or false Default value: false
show_highest	Optional	Defines whether to display the highest severity on monitor box. Value: true or false Default value: false

Table 26. Attributes of the <monitor> element (continued)

Attribute name	Required or optional	Description
show_lowest	Optional	Defines whether to display the lowest severity on the monitor box. Value: true or false Default value: false
show_metric	Optional	Defines whether to display a metric on the monitor box. Value: true or false Default value: false
show_highest_severity_as_border	Optional	Defines whether to use the color of the highest severity alert that the filter captures as the border of the monitor box. Value: true or false Default value: false
foreground_color	Optional	The name of the color to use for the foreground of the monitor box. Value: String Default value: None
background_color	Optional	The name of the color to use as for the background of the monitor box. Value: String Default value: None

<icon>:

The <icon> element is a child of the <map> element and defines the characteristics of an icon on a map. The element can occur any number of times and, if present, has the following attributes:

Table 27. Attributes of the <icon> element

Attribute name	Required or optional	Description
filter	Optional	Defines a filter to associate with the icon. Value: String Default value: None
filterType	Optional	The type of the filter defined in the filter attribute. Value: global or system Default value: None
name	Required	The name of the icon. Value: String Default value: None

Table 27. Attributes of the <icon> element (continued)

Attribute name	Required or optional	Description
label	Optional	The text to appear on the icon. Value: String Default value: None
datasource	Optional	The name of the data source to associate with the icon. To specify more than one data source, use a comma-separated list. Value: String Default value: NCOMS
x	Optional	The horizontal position of the icon in pixels from the left of the map Value: Integer Default value: None
y	Optional	The vertical position of the icon in pixels from the bottom of the map. Value: Integer Default value: None
translucency	Optional	The level of translucency for the icon. Value: Integer between 0 and 100 (inclusive). Default value: None
show_shadow	Optional	Specifies whether the icon appears on the map with a shadow. Value: true or false Default value: false
font	Optional	The name of the font to use for the icon text. Value: String Default value: None
font_color	Optional	The name of the color to use for the icon text. Value: String Default value: None
size	Optional	The point size of the font to use for the icon text. Value: Integer Default value: None
style	Optional	The text style (for example, bold) for the icon text. Value: String Default value: None

Table 27. Attributes of the <icon> element (continued)

Attribute name	Required or optional	Description
url	Optional	The uniform resource locator that is the action target for the icon. Value: String Default value: None
action	Optional	The action that takes place when the user clicks on the icon. Value: String Default value: None
target	Optional	The target window in the browser for output from the value of the url attribute. When the attribute is not present, output replaces the content of the current browser window. Value: String Default value: _blank
flash	Optional	Defines whether the icon flashes. Value: true or false Default value: false
w	Optional	The width of the icon in pixels. Value: Integer Default value: None
h	Optional	The height of the icon in pixels. Value: Integer Default value: None
arc_diameter	Optional	The arc diameter of the icon, in degrees. Value: Integer Default value: None
legend	Optional	The legend for the icon. Value: String Default value: None
image	Optional	The name of an image file to use for the icon. Value: String Default value: None
entity_status_indicator	Optional	Defines the feedback property for the icon. Value: Highlight Bar, Fill Background, or Glow Background. Default value: None

<line>:

The *<line>* is a child of the *<map>* element and defines the characteristics of a line on a map. The element can occur any number of times and, if present, has the following attributes:

Table 28. Attributes of the <line> element

Attribute name	Required or optional	Description
filter	Optional	Defines a filter to associate with the line. Value: String Default value: None
filterType	Optional	The type of the filter defined in the filter attribute. Value: global or system Default value: None
name	Required	The name of the line. Value: String Default value: None
label	Optional	The text to appear on the line. Value: String Default value: None
datasource	Optional	The name of the data source to associate with the line. To specify more than one data source, use a comma-separated list. Value: String Default value: NCOMS
x	Optional	The horizontal position of the line in pixels from the left of the map Value: Integer Default value: None
y	Optional	The vertical position of the line in pixels from the bottom of the map. Value: Integer Default value: None
translucency	Optional	The level of translucency for the line. Value: Integer between 0 and 100 (inclusive). Default value: None
show_shadow	Optional	Specifies whether the line appears on the map with a shadow. Value: true or false Default value: false

Table 28. Attributes of the <line> element (continued)

Attribute name	Required or optional	Description
action	Optional	The action that takes place when the user clicks on the line. Value: String Default value: None
url	Optional	The uniform resource locator that is the action target for the line. Value: String Default value: None
target	Optional	The target window in the browser for output from the value of the url attribute. When the attribute is not present, output replaces the content of the current browser window. Value: String Default value: _blank
flash	Optional	Defines whether the line flashes. Value: true or false Default value: false
thickness	Optional	The thickness of the line in pixels. Value: Integer Default value: None
x2	Optional	The ending, horizontal position of the line in pixels from the left of the map. Value: Integer Default value: None
y2	Optional	The ending, vertical position of the line in pixels from the bottom of the map. Value: Integer Default value: None
color	Optional	The name of the color to use for the line. Value: String Default value: None

Example:

The following example creates a map named map1 that contains two text labels, three buttons, and three monitor boxes.

```
<methodCall>
<method methodName="map.createMap">
  <map name="map1">
    <text name="map1" label="map1" x="547" y="30" font="Helvetica" size="22"
      justify="center" style="bi" color="black" rotate="315" />

    <text name="ael_instructions" label="Click Monitor Box for Tableview"
      x="550" y="141" font="Helvetica" size="10" justify="center"
      style="b" color="black" />

    <button name="button1" label="Active Event List" x="296" y="146"
      action="ael" filter="Example_LastDay" filtertype="system"
      target="hidden" w="98" h="20" color="lightGray" type="rectangle"
      arc_diameter="20" transparent="false" legend="Label" font="Helvetica"
      size="11" font_color="black" show_shadow="true" />

    <button name="button2" label="Active Event List" x="144" y="146"
      action="ael" filter="Example_Critical" filtertype="system"
      target="hidden" w="98" h="20" color="lightGray" type="rectangle"
      arc_diameter="20" transparent="false" legend="Label" font="Helvetica"
      size="11" font_color="black" show_shadow="true" />

    <button name="button3" label="Active Event List" x="21" y="146"
      action="ael" filter="Example_Unassigned" filtertype="system"
      target="hidden" w="98" h="20" color="lightGray" type="rectangle"
      arc_diameter="20" transparent="false" legend="Label" font="Helvetica"
      size="11" font_color="black" show_shadow="true" />

    <monitor name="lastday" label="Last Day" x="294" y="10" action="table"
      target="_blank" filter="Example_LastDay" filtertype="system" w="100"
      h="126" type="histogram" show_label="true" show_total="true"
      show_highest="false" show_lowest="false" show_metric="true"
      foreground_color="red" background_color="lightGray" font="Helvetica"
      size="10" style="p" show_highest_severity_as_border="false" />

    <monitor name="critical" label="Critical" x="142" y="10" action="table"
      target="_blank" filter="Example_Critical" filtertype="system" w="100"
      h="126" type="histogram" show_label="true" show_total="true"
      show_highest="false" show_lowest="false" show_metric="true"
      foreground_color="black" background_color="lightGray" font="Helvetica"
      size="10" style="p" show_highest_severity_as_border="false" />

    <monitor name="unassigned" label="Unassigned" x="17" y="10" action="table"
      target="_blank" filter="Example_Unassigned" filtertype="system" w="100"
      h="126" type="histogram" show_label="true" show_total="true"
      show_highest="false" show_lowest="false" show_metric="true"
      foreground_color="black" background_color="lightGray" font="Helvetica"
      size="10" style="p" show_highest_severity_as_border="false" />
  </map>
</method>
</methodCall>
```


Create or replace a map:

The format of the <method> element when creating or replacing a map is:

```
<method methodName="map.createOrReplaceMap">
```

Use this method to replace an existing map or create one if it does not already exist. The <method> element contains one or more <map> elements each of which define the characteristics of a map. Each <map> element contains any number of the <text>, <button>, <monitor>, <icon>, and <line> elements.

Related reference

"<map>" on page 125

"<text>" on page 125

"<button>" on page 128

"<monitor>" on page 130

"<icon>" on page 133

"<line>" on page 136

Example:

This example creates or replaces a map named map3 that has two monitor boxes and a text label.

```
<methodCall>
  <method methodName="map.createOrReplaceMap">
    <map name="map3">
      <text name="ael instructions" label="Click Monitor Box for Tableview"
        x="550" y="141" font="Helvetica" size="10" justify="center"
        style="b" color="black" />

      <monitor name="critical" label="Critical" x="142" y="10" action="table"
        target="_blank" filter="Example_Critical" filtertype="system" w="100"
        h="126" type="histogram" show_label="true" show_total="true"
        show_highest="false" show_lowest="false" show_metric="true"
        foreground_color="black" background_color="lightGray" font="Helvetica"
        size="10" style="p" show_highest_severity_as_border="false" />

      <monitor name="unassigned" label="Unassigned" x="17" y="10" action="table"
        target="_blank" filter="Example_Unassigned" filtertype="system" w="100"
        h="126" type="histogram" show_label="true" show_total="true"
        show_highest="false" show_lowest="false" show_metric="true"
        foreground_color="black" background_color="lightGray" font="Helvetica"
        size="10" style="p" show_highest_severity_as_border="false" />
    </map>
  </method>
</methodCall>
```

Modify a map:

The format of the <method> element when modifying a map is:

```
<method methodName="map.modifyMap">
```

Use this method to modify an existing map. The <method> element contains one or more <map> elements each of which identifies the map and the characteristics to change. Each <map> element contains any number of the <text>, <button>, <monitor>, <icon>, and <line> elements.

Related reference

"<map>" on page 125

"<text>" on page 125

"<button>" on page 128

"<monitor>" on page 130

"<icon>" on page 133

"<line>" on page 136

Example:

This example makes the following changes to a map named map1:

- Set the background color to white.
- Change the width of the map to 850 pixels
- Change the height of the map to 490 pixels.

```
<methodCall>  
  <method methodName="map.modifyMap">  
    <map name="map1" bgImage="" bgColor="white" w="850" h="490" >  
    </map>  
  </method>  
</methodCall>
```

Delete a map:

The format of the <method> element when deleting a map is:

```
<method methodName="map.deleteMap">
```

Use this method to delete an existing map. The <method> element contains one or more <map> elements each of which identifies a map to delete. In the <map> element include only the name attribute.

Related reference

"<map>" on page 125

Example:

This example deletes the map named map2.

```
<methodCall>  
  <method methodName="map.deleteMap">  
    <map name="map2">  
    </map>  
  </method>  
</methodCall>
```

Get a list of maps:

The format of the <method> element for getting a list of maps is:

```
<method methodName="map.getList">
```

The method returns a list of all the maps defined in the Web GUI.

Example:

```
<methodCall>
  <method methodName="map.getList">
  </method>
</methodCall>
```

Add a map visual:

The format of the `<method>` element when adding a visual to a map is:

```
<method methodName="map.addMapVisual">
```

Use this method to add one or more objects to an existing map. The `<method>` element contains one or more `<map>` elements that identify maps to add new objects to. Each `<map>` element contains any number of `<text>`, `<button>`, `<monitor>`, `<icon>`, and `<line>` elements each of which define a new object to add to the map.

Related reference

"`<map>`" on page 125

"`<text>`" on page 125

"`<button>`" on page 128

"`<monitor>`" on page 130

"`<icon>`" on page 133

"`<line>`" on page 136

Example:

This example adds a text label and a monitor box to the map named map2.

```
<methodCall>
  <method methodName="map.addMapVisual">
    <map name="map2">
      <text name="map2" label="map2" x="147" y="30" font="Helvetica" size="22"
        justify="center" style="bi" color="black" />

      <monitor name="lastday" label="Last Day" x="131" y="92" action="table"
        url="" target="_blank" filter="ExampleLastDay" filtertype="system"
        w="100" h="126" type="histogram" show_label="true" show_total="true"
        show_highest="false" show_lowest="false" show_metric="true"
        foreground_color="red" background_color="lightGray" font="Helvetica"
        size="10" style="p" show_highest_severity_as_border="false" />
    </map>
  </method>
</methodCall>
```

Add or replace a map visual:

The format of the `<method>` element when adding or replacing a map object is:

```
<method methodName="map.createOrReplaceMapVisual">
```

Use this method to add a new object to a map, or replace the object if it already exists. The `<method>` element contains one or more `<map>` elements whose contents are to be modified. Each `<map>` element contains any number of `<text>`, `<button>`, `<monitor>`, `<icon>`, and `<line>` elements each of which defines an object to add or replace. in the `<map>` element include only the name attribute.

Related reference

"<map>" on page 125

"<text>" on page 125

"<button>" on page 128

"<monitor>" on page 130

"<icon>" on page 133

"<line>" on page 136

Example:

This example creates or replaces an icon on the map named map2.

```
<methodCall>
  <method methodName="map.createOrReplaceMapVisual">
    <map name="map2">
      <icon name="visual_3" label="Active Icon" x="75" y="233" action="ael"
        filter="Example_Critical" filtertype="system" target="hidden" w="30" h="19"
        type="rectangle" arc_diameter="10" legend="None" font="Helvetica"
        size="10" font_color="black" image="blocks.png"
        entity_status_indicator="Glow Background"/>
    </map>
  </method>
</methodCall>
```

Modify a map visual:

The format of the <method> element when modifying an object on a map is:

```
<method methodName="map.modifyMapVisual">
```

Use this method to change the characteristics of an object on a map. The <method> element contains one or more <map> elements that identify the maps to modify. Each <map> element contains one or more <text>, <button>, <monitor>, <icon>, and <line> elements which define the objects to modify. In the <map> element include only the name attribute.

Related reference

"<map>" on page 125

"<text>" on page 125

"<button>" on page 128

"<monitor>" on page 130

"<icon>" on page 133

"<line>" on page 136

Example:

This example makes the following changes to a text label named visual_1 on a map named map1:

- Set the label to My Active Button
- Change the horizontal position of the button to 230 pixels
- Change the vertical position of the button to 188 pixels

```
<methodCall>
  <method methodName="map.modifyMapVisual">
    <map name="map1">
```

```

        <button name="visual_1" label="My Active Button" x="230" y="188" />
    </map>
</method>
</methodCall>

```

Delete a map visual:

The format of the `<method>` element when deleting an object from a map is:

```
<method methodName="map.deleteMapVisual">
```

Use this method to remove one or more objects from a map. The `<method>` element contains one or more `<map>` elements each of which identify a map to modify. Each `<map>` element can contains one or more `<text>`, `<button>`, `<monitor>`, `<icon>`, or `<line>` elements that identify the object to remove. In all of these elements, include only the name attribute.

Related reference

“`<map>`” on page 125

“`<text>`” on page 125

“`<button>`” on page 128

“`<monitor>`” on page 130

“`<icon>`” on page 133

“`<line>`” on page 136

Example:

This example removes a text label name `visual_4` from a map named `map1`.

```

<methodCall>
  <method methodName="map.deleteMapVisual">
    <map name="map1">
      <text name="visual_4" />
    </map>
  </method>
</methodCall>

```

Resource requests

Resources are images that can appear on Web GUI maps. There are functions to add and remove resources as well as obtain a list of the resources defined on the Web GUI server.

WAAPI provides four methods for working with map resources:

- “Add a resource” on page 144
- “Create or replace a resource” on page 145
- “Remove a resource” on page 145
- “Get a list of resources” on page 146

Add a resource:

The format of the `<method>` element for adding a resource is:

```
<method methodName="resource.addResource">
```

Use this method to add a resource to the system. The `<method>` element contains one or more `<resources>` elements each of which defines a map to add the resource to. The `<resources>` element contains one or more `<resource>` elements each of which identifies a resource.

`<resources>`:

The `<resources>` element defines a set of resources for a map and has the following attribute:

Table 29. Attributes of the `<resources>` element

Attribute name	Required or optional	Description
mapName	Required	The name of a map to add the resource to. Value: String Default value: None

`<resource>`:

The `<resource>` element is a child element of `<resources>`. The element defines a resource for a map and has the following attribute:

Table 30. Attributes of the `<resource>` element

Attribute name	Required or optional	Description
name	Required	The name of the resource to add to the map. Value: String Default value: None

Example:

This example adds an image named `ny.gif` to a map named `map1`.

```
<methodCall>  
  <method methodName="resource.addResource">  
    <resources mapName="map1">  
      <resource name="ny.gif" />  
    </resources>  
  </method>  
</methodCall>
```

Create or replace a resource:

The format of the `<method>` element for creating or replacing a resource is:

```
<method methodName="createOrReplaceResource">
```

Use this method to add a resource to a map, if the map does not contain the resource, or to replace the existing instance of the resource on the map. The `<method>` element contains any number of `<resources>` elements that identify the maps to modify. Each `<resources>` element contains one or more `<resource>` elements that each identify the resources to add or replace.

Example

This example adds or replaces an image named `ny.gif` to a map named `map1`.

```
<methodCall>
  <method methodName="resource.createOrReplaceResource">
    <resources mapName="map1">
      <resource name="ny.gif" />
    </resources>
  </method>
</methodCall>
```

Related reference

“`<resources>`” on page 144

“`<resource>`” on page 144

Remove a resource:

The format of the `<method>` element for removing a resource is:

```
<method methodName="removeResource">
```

Use this method to remove a resource from a map. The `<method>` element contains any number of `<resources>` elements that identify the maps to modify. Each `<resources>` element contains one or more `<resource>` elements that each identify the resources to add or replace.

Example

This example removes the resource `ny.gif` from a map named `map1`.

```
<methodCall>
  <method methodName="resource.removeResource">
    <resources mapName="map1">
      <resource name="ny.gif" />
    </resources>
  </method>
</methodCall>
```

Related reference

“`<resources>`” on page 144

“`<resource>`” on page 144

Get a list of resources:

The format of the <method> element for getting a list of resources is:

```
<method methodName="resource.getList">
```

This method returns a list of all the resources defined on the Web GUI server and has no child elements. Each map has its own section in the listing.

Example

```
<methodCall>  
  <method methodName="resource.getList">  
  </method>  
</methodCall>
```

File requests

File requests enable you to work on files and directories on the Web GUI server. You can create and delete files and directories.

WAAPI provides six methods for working with files:

- “Add a directory”
- “Add a file” on page 147
- “Create or replace a file” on page 148
- “Delete a file” on page 148
- “Remove a directory” on page 148
- “Recursively remove a directory” on page 149

Add a directory:

The format of the <method> element for adding a directory to the Web GUI server is:

```
<method methodName="file.addDir">
```

Use this method to create a new directory on the Web GUI server. The <method> element contains one or more <file> elements each of which defines a directory to create on the system. In each <file> element include the dirName attribute to define the directory to create.

<file>:

The <file> element defines the characteristics of a file or directory to work with. The element has the following attributes:

Table 31. Attributes of the <file> element

Attribute name	Required or optional	Description
fileName	Optional	The name of a file to work with. Value: String Default value: None
dirName	Optional	The specification of a directory to work with. Value: String Default value: None

Table 31. Attributes of the <file> element (continued)

Attribute name	Required or optional	Description
toDir	Optional	The destination directory for a command. Value: String Default value: None
fromDir	Optional	The source directory for a command. Value: String Default value: None

Example:

This example creates a directory named data on the C: drive of a Windows system.

```
<methodCall>
  <method methodName="file.addDir">
    <file dirName="C:\data">
    </file>
  </method>
</methodCall>
```

Add a file:

The format of the <method> element for adding a file to the Web GUI server is:

```
<method methodName="file.addFile">
```

Use this method to create a file on the Web GUI server. The <method> element contains one or more <file> elements that each define the name and location of a file to create. The content of the <file> element becomes the content of the new file.

Example

This example creates a file named data.txt in the directory C:\data\ on a Windows system.

```
<methodCall>
  <method methodName="file.addFile">
    <file fileName="data.txt" toDir="C:\data">
    </file>
  </method>
</methodCall>
```

Related reference

"<file>" on page 146

Create or replace a file:

The format of the <method> element for creating or replacing a file is:

```
<method methodName="file.createOrReplaceFile">
```

Use this method to replace an existing file with the same name or to create the file, if it does not exist. The <method> element contains one or more <file> elements each of which identifies a file to create or replace. The content of the <file> element becomes the content of the new file.

Example

This example creates or replaces a file named hello.txt in the directory C:\data and sets the contents of the file to the text "Hello world".

```
<methodCall>
  <method methodName="file.createOrReplaceFile">
    <file fileName="hello.txt" toDir="C:\data">
      </file>
    </method>
  </methodCall>
```

Related reference

"<file>" on page 146

Delete a file:

The format of the <method> element for deleting a file is:

```
<method methodName="file.deleteFile">
```

Use this method to delete a file from the Web GUI server. The <method> element contains one or more <file> elements each of which defines the name and location of a file to delete.

Example

This example deletes the file hello.txt from the directory C:\data.

```
<methodCall>
  <method methodName="file.deleteFile">
    <file fileName="hello.txt" fromDir="C:\data">
      </file>
    </method>
  </methodCall>
```

Related reference

"<file>" on page 146

Remove a directory:

The format of the <method> element for removing a directory is:

```
<method methodName="deleteDir">
```

Use this method to remove a directory from the Web GUI server. The <method> element contains one or more <file> elements each of which defines a directory to remove. Each directory must be empty.

Example

This example removes the directory named C:\test.

```

<methodCall>
  <method methodName="deleteDir">
    <file dirName="test">
    </file>
  </method>
</methodCall>

```

Related reference

“<file>” on page 146

Recursively remove a directory:

The format of the <method> element for removing a directory recursively is:

```
<method methodName="file.recurseRemove">
```

Use this method to remove a directory and all its contents. This method is useful for deleting directory trees. The <method> element contains one or more <file> elements each of which defines a directory to delete, along with all its contents.

Example

This example recursively removes the directory named C:\data.

```

<methodCall>
  <method methodName="recurseRemove">
    <file dirName="C:\data">
    </file>
  </method>
</methodCall>

```

Related reference

“<file>” on page 146

Menu requests

Menus provide users with access to Web GUI functions. There are functions to create, modify, and delete menus. You can also get a list of all the menus on the system.

WAAPI provides five methods for working with menus:

- “Create a menu”
- “Create or replace a menu” on page 151
- “Modify a menu” on page 152
- “Delete a menu” on page 153
- “Get a list of menus” on page 153

Create a menu:

The format of the <method> element for creating a menu is:

```
<method methodName="menu.createMenu">
```

Use this method to create a new menu that can contain tools, submenus, and separators. The <method> element contains one or more <supermenu> elements. Each <supermenu> element contains any number of <separator>, <menu>, and <tool> elements that define the content of the menu.

<supermenu>:

The *<supermenu>* element is the container for a menu and has the following attributes:

Table 32. Attributes of the <supermenu> element

Attribute name	Required or optional	Description
name	Required	Provides a unique name for a menu. Value: String Default value: None
label	Optional	The title of the menu as it appears in the AEL. Value: String Default value: The value of the name attribute.
mnemonic	Optional	A single character that allows the user to select the menu using the Alt key. Value: A single alphabetic character Default value: None

<separator>:

The *<separator>* element is a child of the *<supermenu>* element and defines the location of a separator line in a menu and can occur any number of times. The element has no content and no attributes.

<menu>:

The *<menu>* element is a child of the *<supermenu>* element and defines the location of a submenu. The element can occur any number of times and has the following attributes:

Table 33. Attributes of the <menu> element

Attribute name	Required or optional	Description
name	Required	Provides a unique name for a menu. Value: String Default value: None
label	Required	The title of the menu as it appears in the AEL. Value: String Default value: None.
mnemonic	Optional	A single character that allows the user to select the menu using the Alt key. Value: A single alphabetic character Default value: None

`<tool>`:

The `<tool>` element is a child of the `<supermenu>` element that defines a tool to appear as an option on the menu. The element can appear any number of times and has the following attributes:

Table 34. Attributes of the `<tool>` element

Attribute name	Required or optional	Description
name	Required	Provides a unique name for a tool. Value: String Default value: None
label	Required	The title of the tool as it appears on the menu in the AEL. Value: String Default value: None.
mnemonic	Optional	A single character that allows the user to select the menu that holds the tool using the Alt key. Value: A single alphabetic character Default value: None
shortcut	Optional	A single character that allows the user to select the tool using the Ctrl key. Value: A single alphabetic character Default value: None

Example:

This example creates empty menus named `menu1` and `toolMenu1` that other functions add items to.

```
<methodCall>
  <method methodName="menu.createMenu">
    <supermenu name="menu1" >
    </supermenu>
    <supermenu name="toolMenu1">
    </supermenu>
  </method>
</methodCall>
```

Create or replace a menu:

The format of the `<method>` element for creating or replacing a menu is:

```
<method methodName="menu.createOrReplaceMenu">
```

Use this method to create a new menu or replace an existing one that can contain tools, submenus, and separators. The `<method>` element contains one or more `<supermenu>` elements. Each `<supermenu>` element contains any number of `<separator>`, `<menu>`, and `<tool>` elements that define the content of the menu.

Example

This example creates or replaces a menu named `toolMenu2` and adds two tools to it.

```

<methodCall>
  <method methodName="menu.createOrReplaceMenu">
    <supermenu name="toolMenu2">
      <tool shortcut="" label="Ping" name="Ping" mnemonic="" />
      <tool shortcut="" label="commandTool1" name="commandTool1" mnemonic="" />
    </supermenu>
  </method>
</methodCall>

```

Related reference

“<supermenu>” on page 150

“<separator>” on page 150

“<menu>” on page 150

“<tool>” on page 151

Modify a menu:

The format of the <method> element for modifying a menu is:

```

<method methodName="menu.modifyMenu">

```

Use this method to modify an existing menu. The <method> element contains one or more <supermenu> elements. Each <supermenu> element contains any number of <separator>, <menu>, and <tool> elements that define the content of the menu.

Example

This example adds a menu and two tools to menu1 and then adds that menu, together with further tools, to toolMenu2.

```

<methodCall>
  <method methodName="menu.modifyMenu">

    <supermenu name="menu1" mnemonic="n" >
      <menu name="alerts" label="Alerts" mnemonic="a" />
      <separator />
      <tool name="acknowledge" label="Acknowledged" mnemonic="a"
        shortcut="ctrl+a" />
      <separator />
      <tool name="prioritise" label="Prioritize" mnemonic="p" shortcut="" />
    </supermenu>

    <supermenu name="toolMenu1">
      <tool shortcut="" label="Ping" name="Ping" mnemonic="" />
      <separator />
      <menu name="menu1" label="menu1"/>
      <separator />
      <tool shortcut="" label="commandTool1" name="commandTool1" mnemonic="" />
    </supermenu>
  </method>
</methodCall>

```

Related reference

“<supermenu>” on page 150

“<separator>” on page 150

“<menu>” on page 150

“<tool>” on page 151

Delete a menu:

The format of the <method> element for deleting a menu is:

```
<method methodName="menu.deleteMenu">
```

Use this method to delete an existing menu. The <method> element contains one or more <supermenu> elements each of which identifies a menu to delete. In the <supermenu> element include only the name attribute.

Example

This example deletes a menu named menu1

```
<methodCall>
  <method methodName="menu.deleteMenu">
    <supermenu name="menu1">
    </supermenu>
  </method>
</methodCall>
```

Get a list of menus:

The format of the <method> element for getting a list of maps is:

```
<method methodName="menu.getList">
```

The method returns a list of all the menus defined in the Web GUI.

Example

```
<methodCall>
  <method methodName="menu.getList">
  </method>
</methodCall>
```

Usage notes:

You can add menus and tools to the **Alerts** and **Tools** menus of the AEL only. At the top level, add the menus you define to these menus. Then you can add further menus and tools to those you have defined, if required.

Tool requests

Tools provide enhanced abilities for user to manage events in the AEL. There are functions to create, modify, and delete menus. You can also get a list of all the tools defined on the system.

WAAPI provides five methods for working with tools:

- "Create a tool" on page 154
- "Create or replace a tool" on page 161
- "Modify a tool" on page 162
- "Delete a tool" on page 163
- "Get a list of tools" on page 163

Create a tool:

The format of the `<method>` element for creating a tool is:

```
<method methodName="createTool">
```

Use this method to create a new tool. The `<method>` element contains one or more `<tool:tool>` elements. Each `<tool:tool>` element contains a `<tool:access>` element followed by any number of the following elements:

- `<tool:sql>`
- `<tool:journal>`
- `<tool:cgiurl>`
- `<tool:cmdline>`
- `<tool:script>`

Each of those elements defines a particular tool.

`<tool:tool>`:

The `<tool:tool>` element is the container for a tool and can occur any number of times. The element has the following attributes:

Table 35. Attributes of the `<tool:tool>` element

Attribute name	Required or optional	Description
name	Required	Provides a unique name for a tool. Value: String Default value: None
datasource	Optional	The data sources that a tool uses. To specify more than one data source use a comma-separated list. Value: String Default value: The default data source

`<tool:access>`:

The `<tool:access>` element is a child of the `<tool:tool>` element and defines the access criteria for a tool. The element has no attributes but contains the following child elements in this order:

- `<tool:osfield>`
- `<tool:security>`

`<tool:osfield>`:

The `<tool:osfield>` contains the ObjectServer fields that a tool uses. The element has no attributes but contains any number of `<tool:criterion>` elements.

`<tool:criterion>`:

The `<tool:criterion>` element is a child of the `<tool:osfield>` and `<tool:security>` elements. It defines an access criterion for a tool and contains one or more `<tool:equals>` elements. In addition, the element has the following attribute:

Table 36. Attributes of the `<tool:criterion>` element

Attribute name	Required or optional	Description
name	Required	The type of access for a tool. Value: Class or Group. When this element is a child of <code><tool:osfield></code> the value of this attribute must be Class. When the element is a child of <code><tool:security></code> the value of the attribute must be Group. Default value: None

`<tool:equals>`:

The `<tool:equals>` element is a child of `<tool:criterion>` and assigns a value to an object defined by its parent element. The element has no content but has the following attributes:

Table 37. Attributes of the `<tool:equals>` element

Attribute name	Required or optional	Description
value	Required	Provides a value for the access criterion. Value: String Default value: None
datasource	Optional	The data sources that the criterion uses. To specify more than one data source use a comma-separated list. Value: String Default value: The default data source

`<tool:security>`:

The `<tool:security>` element defines the access criteria for an AEL tool. The tool contains any number of `<tool:criterion>` elements that define the access criteria. The element has no attributes.

Related reference

“`<tool:criterion>`” on page 155

`<tool:sql>`:

The `<tool:sql>` element contains the SQL command and associated options for an SQL AEL tool. The element is empty but has the following attributes:

Table 38. Attributes of the `<tool:sql>` element

Attribute name	Required or optional	Description
foreach	Optional	Defines whether a tool acts on each of the selected rows in the AEL. Value: true or false Default value: false
command	Required	The SQL command for the tool. Value: String Default value: None

`<tool:journal>`:

The `<tool:journal>` element contains the journal entry made when an AEL tool is run. The element is empty but has the following attributes:

Table 39. Attributes of the `<tool:journal>` element

Attribute name	Required or optional	Description
foreach	Optional	Defines whether a tool acts on each of the selected rows in the AEL. Value: true or false Default value: false
entry	Required	The text for the journal entry. Value: String Default value: None

`<tool:cgiurl>`:

The `<tool:cgiurl>` element defines a CGI/URL tool. The element contains the `<tool:fieldlist>` element and has the following attributes:

Table 40. Attributes of the `<tool:cgiurl>` element

Attribute name	Required or optional	Description
foreach	Optional	Defines whether a tool acts on each of the selected rows in the AEL. Value: true or false Default value: false
windowforeach	Optional	Defines whether to display an output window for each row when the tool is run against the AEL. Value: true or false Default value: false
method	Optional	The method that the tool uses to send data to the server. Value: GET or POST Default value: GET
target	Optional	The target window in the browser for output from tool linked to a URL, specified in the url attribute. Value: String Default value: <code>_blank</code>
url	Required	The uniform resource locator that is the action target for the tool. Value: String Default value: None

`<tool:fieldlist>`:

The `<tool:fieldlist>` element is a child of `<tool:cgiurl>` and defines the ObjectServer fields to use with the tool. The element contains one or more `<tool:field>` elements each identifying an ObjectServer field and has no attributes.

`<tool:field>`:

The `<tool:field>` element is a child of `<tool:fieldlist>` and defines the name of an ObjectServer field to use in the tool. the element is empty but has the following attribute:

Table 41. Attributes of the `<tool:field>` element

Attribute name	Required or optional	Description
name	Required	The name of an ObjectServer field. Value: String Default value: None

<tool:cmdline>:

The *<tool:cmdline>* element defines a command line tool. The element contains one or more *<tool:command>* elements and has no attributes.

<tool:command>:

The *<tool:command>* element is a child of *<tool:cmdline>* and defines the platform-specific command associated with a command line tool. The element is empty but has the following attributes:

Table 42. Attributes of the <tool:command> element

Attribute name	Required or optional	Description
enabled	Optional	Defines whether is enabled or disabled. It operates in conjunction with the platform attribute to determine if a tool is available on a specific platform Value: true or false Default value: false
platform	Required	Identifies the platform that the tool applies to. It operates in conjunction with the enabled attribute to determine if a tool is available on a specific platform. Value: Windows, Solaris, Linux, HPUX, or AIX Default value: None
executable	Required	The command that executes when the tool is run. Value: String Default value: None
foreach	Optional	Defines whether a tool acts on each of the selected rows in the AEL. Value: true or false Default value: false

<tool:script>:

The *<tool:script>* element defines a script tool. The element is empty but has the following attributes:

Table 43. Attributes of the <tool:script> element

Attribute name	Required or optional	Description
foreach	Optional	Defines whether a tool acts on each of the selected rows in the AEL. Value: true or false Default value: false

Table 43. Attributes of the <tool:script> element (continued)

Attribute name	Required or optional	Description
command	Required	The script for the tool. Value: String Default value: None

Usage notes:

The <tool:journal> element is typically used in conjunction with other tools to create journal entries recording what the tools have done. For example, you can use the <tool:journal> element to record the action taken by a <tool:sql> element.

To create a command line tool that can run on any platform in your organization, use multiple <tool:cmdline> elements. Each of these contains a platform-specific command all of which achieve the same, or similar, result. For example, you could have an element for Windows and another Linux. The example in “Create or replace a tool” on page 161 shows this technique.

Examples:

SQL tool

The following example creates a SQL tool named sqlSample1.

```
<methodCall xmlns:tool="http://www.ibm.com/tivoli/netcool/webtop/tools/2.1">
  <method methodName="tool.createTool">
    <!--
      SQL tool with Class access criteria set to Tivoli TEC (Oracle) (7450)
      and Tivoli TEC (Sybase) (7500), and Group access criteria set to restricted
      and Desktop.
    -->
    <tool:tool name="sqlSample1">
      <tool:access>
        <tool:osfield>
          <tool:criterion name="Class">
            <tool>equals value="7450"/>
            <tool>equals value="7500"/>
          </tool:criterion>
        </tool:osfield>
        <tool:security>
          <tool:criterion name="Group">
            <tool>equals value="restricted"/>
            <tool>equals value="Desktop"/>
          </tool:criterion>
        </tool:security>
      </tool:access>
      <tool:sql foreach="true"
        command="update alerts.status set Summary='sqlTest1' where Serial=@Serial;"
      />
      <tool:journal foreach="true"
        entry="Tool executed on event Class CONVERSION(@Class)."/>
      </tool:tool>
    </method>
  </methodCall>
```

CGI/URL tool

The following example creates a CGI/URL tool named cmsSample1.

```

<methodCall xmlns:tool="http://www.ibm.com/tivoli/netcool/webtop/tools/2.1">
  <method methodName="tool.createTool">
    <!--
      Command Line tool with Group access criteria set to restricted and Desktop.
      Class access criteria is undefined, therefore executable against events of
      any Class.
    -->
    <tool:tool name="cmdSample">
      <tool:access>
        <tool:osfield/>
        <tool:security>
          <tool:criterion name="Group">
            <tool>equals value="restricted"/>
            <tool>equals value="Desktop"/>
          </tool:criterion>
        </tool:security>
      </tool:access>
      <tool:cmdline>
        <tool:command foreach="true" enabled="true" platform="Windows"
          executable="start cmd /k c:\temp\sample.bat {@Class} {@Node}"/>
        <tool:command foreach="true" enabled="true" platform="Solaris"
          executable="xterm -e /bin/sh -c '/tmp/sample.sh {@Class} {@Node}'; read a'"
          />
        <tool:command foreach="true" enabled="false" platform="AIX"
          executable="xterm -e /bin/sh -c '/tmp/sample.sh {@Class} {@Node}'; read a'"
          />
      </tool:cmdline>
    </tool:tool>
  </method>
</methodCall>

```

Script tool

The following example creates a script tool named scriptSample1.

```

<methodCall xmlns:tool="http://www.ibm.com/tivoli/netcool/webtop/tools/2.1">
  <method methodName="tool.createTool">
    <!--
      Script tool with Group access criteria set to restricted and Desktop. Class
      access criteria is undefined, therefore executable against events of
      any Class.
    -->
    <tool:tool name="scriptSample">
      <tool:access>
        <tool:osfield/>
        <tool:security>
          <tool:criterion name="Group">
            <tool>equals value="restricted"/>
            <tool>equals value="Desktop"/>
          </tool:criterion>
        </tool:security>
      </tool:access>
      <tool:script foreach="false" command="alert('{@Node}');"/>
    </tool:tool>
  </method>
</methodCall>

```

Create or replace a tool:

The format of the <method> element for creating or replacing a tool is:

```
<method methodName="createOrReplaceTool">
```

Use this method to create a new tool or replace an existing tool. The <method> element contains one or more <tool:tool> elements. Each <tool:tool> element contains a <tool:access> element followed by any number of one of the following elements:

- <tool:sql>
- <tool:journal>
- <tool:cgiurl>
- <tool:cmdline>
- <tool:script>

Each of those elements defines a particular tool.

Example

This example creates or replaces a tool named cmdSample.

```
<methodCall xmlns:tool="http://www.ibm.com/tivoli/netcool/webtop/tools/2.1">
  <method methodName="tool.createOrReplaceTool">
    <!-- Create or replace a tool called cmdSample. -->
    <tool:tool name="cmdSample">
      <tool:access>
        <tool:osfield/>
        <tool:security/>
      </tool:access>
      <tool:cmdline>
        <tool:command foreach="true" enabled="true" platform="Windows"
          executable="start cmd /k c:\temp\sample.bat {@Class} {@Node}"/>
        <tool:command foreach="true" enabled="true" platform="Linux"
          executable="xterm -e /bin/sh -c '/tmp/sample.sh {@Class} {@Node}; read a'"/>
        <tool:command foreach="true" enabled="true" platform="AIX"
          executable="xterm -e /bin/sh -c '/tmp/sample.sh {@Class} {@Node}; read a'"/>
      </tool:cmdline>
    </tool:tool>
  </method>
</methodCall>
```

Related reference

"<tool:tool>" on page 154

"<tool:access>" on page 154

"<tool:sql>" on page 156

"<tool:journal>" on page 156

"<tool:cgiurl>" on page 157

"<tool:cmdline>" on page 158

"<tool:script>" on page 158

"Usage notes" on page 159

Modify a tool:

The format of the <method> element for modifying a tool is:

```
<method methodName="modifyTool">
```

Use this method to modify an existing tool. The <method> element contains one or more <tool:tool> elements. Each <tool:tool> element contains a <tool:access> element followed by any number of one of the following elements:

- <tool:sql>
- <tool:journal>
- <tool:cgiurl>
- <tool:cmdline>
- <tool:script>

Example

The following example modifies tools named sqlSample1 and cgiSample.

```
<methodCall xmlns:tool="http://www.ibm.com/tivoli/netcool/webtop/tools/2.1">
  <method methodName="tool.modifyTool">
    <tool:tool name="sqlSample1">
      <tool:access>
        <tool:osfield>
          <tool:criterion name="Class">
            <tool>equals value="7450"/>
          </tool:criterion>
        </tool:osfield>
        <tool:security/>
      </tool:access>
      <tool:sql foreach="true"
        command="update alerts.status set Summary='sqlTest1' where Serial=@Serial;"
      />
      <tool:journal foreach="true" entry="Tool executed on event Class @Class."/>
    </tool:tool>

    <tool:tool name="cgiSample">
      <tool:access>
        <tool:osfield>
          <tool:criterion name="Class">
            <tool>equals value="7500"/>
          </tool:criterion>
        </tool:osfield>
        <tool:security/>
      </tool:access>
      <tool:cgiurl foreach="true" windowforeach="true" target="_blank"
        method="GET" url="$(SERVER)/cgi-bin/sample.cgi">
        <tool:fieldlist>
          <tool:field name="Class"/>
          <tool:field name="Node"/>
        </tool:fieldlist>
      </tool:cgiurl>
    </tool:tool>
  </method>
</methodCall>
```


Related reference

"<tool:tool>" on page 154

"<tool:access>" on page 154

"<tool:sql>" on page 156

"<tool:journal>" on page 156

"<tool:cgiurl>" on page 157

"<tool:cmdline>" on page 158

"<tool:script>" on page 158

"Usage notes" on page 159

Delete a tool:

The format of the <method> element when deleting a tool is:

```
<method methodName="tool.deleteTool">
```

Use this method to delete an existing tool. The <method> element contains one or more <tool:tool> elements each of which identifies a tool to delete. In the <tool:tool> element include only the name attribute.

Example

The following example deletes the tools named sqlSample1, sqlSample2, cgiSample, cmdSample, scriptSample, and scriptSample2.

```
<methodCall xmlns:tool="http://www.ibm.com/tivoli/netcool/webtop/tools/2.1">
```

```
  <!-- Delete tools if they exist. -->
  <method methodName="tool.deleteTool">
    <tool:tool name="sqlSample1"/>
    <tool:tool name="sqlSample2"/>
    <tool:tool name="cgiSample"/>
    <tool:tool name="cmdSample"/>
    <tool:tool name="scriptSample"/>
    <tool:tool name="scriptSample2"/>
  </method>
</methodCall>
```

Get a list of tools:

The format of the <method> element for getting a list of tools is:

```
<method methodName="tool.getList">
```

The method returns a list of all the tools defined in the Web GUI.

Example

```
<methodCall xmlns:tool="http://www.ibm.com/tivoli/netcool/webtop/tools/2.1">
```

```
  <!-- Get a list of tools available in the server. -->
  <method methodName="tool.getList"/>
</method>
</methodCall>
```

Prompt requests

Prompts are associated with certain types of tools. They generate a prompt window or a pop-up menu for users to enter or select information that a tool requires. There are functions to create, modify, and delete prompts. In addition you can obtain a list of prompts defined on the server.

WAAPI provides four methods for working with prompts:

- “Create a prompt”
- “Create or replace a prompt” on page 170
- “Modify a prompt” on page 171
- “Delete a prompt” on page 171
- “Get a list of prompts” on page 172

Create a prompt:

The format of the <method> element for creating a prompt is:

```
<method methodName="prompt.createPrompt">
```

Use this method to create a new prompt. The <method> elements contains one or more <prompt:prompt> elements each of which defines the characteristics of a new prompt. The <prompt:prompt> element contains an optional <prompt:parameters> element followed by an optional <prompt:choice> element.

<prompt:prompt>:

The <prompt:prompt> element defines a prompt and has the following attributes:

Table 44. Attributes of the <prompt:prompt> element

Attribute name	Required or optional	Description
name	Required	Provides a unique name for a prompt. Value: String Default value: None
type	Required	The type of prompt. Value: One of the following values: String, Integer, Float, Password, Time, Lookup, FixedChoice, DynamicChoice, MultiLlneString, FormattedString, RealTimeDynamicChoice Default value: None

<prompt:parameters>:

The <prompt:parameters> element is a child of <prompt:prompt> and defines the parameters for a prompt. The element can occur once or be omitted, if the prompt does not need parameters. For certain prompt types the element contains a <prompt:additionalParameters> element. The <prompt:parameters> element has the following attributes:

Table 45. Attributes of the <prompt:parameters> element

Attribute name	Required or optional	Description
label	Required	The text displayed in a prompt window. It gives the user guidance on the information required. Value: String Default value: None
order	Required	The order of a prompt relative to other prompts in a prompt window. Value: Integer Default value: None
localized	Required	Defines whether a prompt can be localized Value: true or false Default value: None
errorMessage	Optional	The text displayed if a user supplies inappropriate information to a prompt. Value: String Default value: None
defaultValue	Optional	The default response to a prompt that the user can select. Value: String Default value: None

<prompt:additionalParams>:

The <prompt:additionalParams> is a child of <prompt:parameters> that provides additional parameters the following types of tool require:

- Lookup
- Dynamic choice
- Formatted string
- Real-time dynamic choice

The element contains any number of <prompt:param> elements, each of which defines an additional parameter, and has no attributes.

<prompt:param>:

The <prompt:param> element is a child of <prompt:additionalParams> and defines the name and value of an additional parameter for a tool. The usage notes contain more information on how to use this element. The element is empty and has the following attributes:

Table 46. Attributes of the <prompt:param> element

Attribute name	Required or optional	Description
name	Required	The name of the parameter. Value: String Default value: None
value	Required	The value of the parameter. Value: Dependent on the type of prompt but typically a String Default value: None

<prompt:choice>:

The <prompt:choice> element is a child of <prompt:prompt> and defines the choices for a menu in a fixed-choice prompt. It has no attributes but contains one or more <prompt:item> entries, each defining a choice on the menu.

<prompt:item>:

The <prompt:item> element is a child of <prompt:choice> and occurs one or more times. Each occurrence defines a choice on the menu for a fixed-choice prompt. The element has the following attributes:

Table 47. Attributes of the <prompt:item> element

Attribute name	Required or optional	Description
value	Required	The value that the prompt returns when the user selects this option. Value: String Default value: None
label	Required	The text that appears on the menu for this choice. Value: String Default value: None

Usage notes:

The following notes show how to use the child elements of <prompt:prompt> for each type of prompt.

String

Include a single <prompt:parameters> element that contains the string of the prompt to display, its order with in relation to other prompts, an error message, the localized indicator, and the default value. Omit all other child elements.

Integer

Include a single <prompt:parameters> element that contains the string of the prompt to display, its order with in relation to other prompts, an error message, the localized indicator, and the default value. Omit all other child elements.

Float

Include a single `<prompt:parameters>` element that contains the string of the prompt to display, its order with in relation to other prompts, an error message, the localized indicator, and the default value. Omit all other child elements.

Password

Include a single `<prompt:parameters>` element that contains the string of the prompt to display, its order in relation to other prompts, and the localized indicator. Omit all other child elements.

Time

Include a single `<prompt:parameters>` element that contains the string of the prompt to display, its order in relation to other prompts, and the localized indicator. Omit all other child elements.

Lookup

Include a single `<prompt:parameters>` element that contains the string of the prompt to display, its order in relation to other prompts, and the localized flag. In addition supply a `<prompt:additionalParams>` element containing a single `<prompt:param>` element with the following values for its attributes:

Table 48. Values for the `<prompt:param>` element in a lookup prompt

Attribute	Value
name	file
value	The path of the file that contains the items to appear on the lookup list. For example: /tmp/lookup.txt

Fixed choice

Include a single `<prompt:parameters>` element containing the string of the prompt to display, its order in relation to other prompts, and the localized flag. In addition, supply a `<prompt:choice>` element with a `<prompt:item>` element for each choice on the menu.

Dynamic choice

Include a single `<prompt:parameters>` element that contains the string of the prompt to display, its order in relation to other prompts, and the localized flag. In addition supply a `<prompt:additionalParams>` element containing a single `<prompt:param>` element with the following values for its attributes:

Table 49. Values for the `<prompt:param>` element in a dynamic choice prompt

Attribute	Value
name	sqlCommand

Table 49. Values for the <prompt:param> element in a dynamic choice prompt (continued)

Attribute	Value
value	An ObjectServer SQL command for up to two columns from a table. Each row that the ObjectServer returns is shown to the user as an item in a submenu or list. When run against multiple ObjectServers, the SQL command can select only columns and column values that are common to all of them. For example: select Conversion, Value from alerts.conversions where Colname='Severity' order by Value desc;

Multiline string

Include a single <prompt:parameters> element that contains the string of the prompt to display, its order with in relation to other prompts, an error message, the localized indicator, and the default value. Omit all other child elements.

Formatted string

Include a single <prompt:parameters> element that contains the string of the prompt to display, its order in relation to other prompts, an error message, the localized indicator, and the default value. In addition, supply a <prompt:additionalParams> element containing a single <prompt:param> item with the following values for its attributes:

Table 50. Values for the <prompt:param> element in a formatted string prompt

Attribute	Value
name	format
value	A regular expression that the string the user enters must match. For example: ^[a-zA-Z]{3}\d{3}\$

Real-time dynamic choice

Include a single <prompt:parameters> element that contains the string of the prompt to display, its order in relation to other prompts, and the localized flag. In addition supply a <prompt:additionalParams> element containing a single <prompt:param> element with the following values for its attributes:

Table 51. Values for the <prompt:param> element in a formatted string prompt

Attribute	Value
name	sqlCommand
value	An ObjectServer SQL command for up to two columns from a table. Each row that the ObjectServer returns is shown to the user as an item in a submenu or list. When run against multiple ObjectServers, the SQL command can select only columns and column values that are common to all of them. For example: select Conversion, Value from alerts.conversions where Colname='Type' order by Value desc;

The real-time dynamic choice prompt creates a scrollable list containing the results of the ObjectServer query generated when the tool executes. This prompt is meant to be used for data from an ObjectServer table that is frequently changeable. As

this prompt type is executed in real time, use it sparingly to reduce the load on the Web GUI server.

Examples:

The following example creates one of each type of prompt.

```
<methodCall xmlns:prompt="http://www.ibm.com/tivoli/netcool/webtop/prompts/2.2">
  <!-- Create prompts. The prompts must not already exist. -->
  <method methodName="prompt.createPrompt">
    <!-- String prompt -->
    <prompt:prompt type="String" name="testString">
      <prompt:parameters label="Type in a string value" order="100"
        errorMessage="String value cannot be empty" localized="false"
        defaultValue="Default string"/>
    </prompt:prompt>

    <!-- Integer prompt -->
    <prompt:prompt type="Integer" name="testInteger">
      <prompt:parameters label="Type in an integer value" order="110"
        errorMessage="Invalid integer value" localized="false" defaultValue="5"/>
    </prompt:prompt>

    <!-- Float prompt -->
    <prompt:prompt type="Float" name="testFloat">
      <prompt:parameters label="Type in a float value" order="120"
        errorMessage="Invalid float value" localized="false" defaultValue="1.0"/>
    </prompt:prompt>

    <!-- Password prompt -->
    <prompt:prompt type="Password" name="testPassword">
      <prompt:parameters label="Type in the password" order="130"
        localized="false"/>
    </prompt:prompt>

    <!-- Time prompt -->
    <prompt:prompt type="Time" name="testTime">
      <prompt:parameters label="Specify a date/time" order="140"
        errorMessage="Invalid date/time value" localized="false"/>
    </prompt:prompt>

    <!-- Lookup prompt -->
    <prompt:prompt type="Lookup" name="testLookup">
      <prompt:parameters label="Select a lookup option" order="150"
        localized="false">
        <prompt:additionalParams>
          <prompt:param name="file" value="/tmp/lookup.txt"/>
        </prompt:additionalParams>
      </prompt:parameters>
    </prompt:prompt>

    <!-- Fixed Choice prompt -->
    <prompt:prompt type="FixedChoice" name="testFixedChoice">
      <prompt:parameters label="Select a fixed option" order="160"
        localized="false"
        defaultValue="fixed2"/>
      <prompt:choice>
        <prompt:item value="fixed1" label="Fixed One"/>
        <prompt:item value="fixed2" label="Fixed Two"/>
        <prompt:item value="fixed3" label="Fixed Three"/>
      </prompt:choice>
    </prompt:prompt>

    <!-- Dynamic Choice prompt -->
    <prompt:prompt type="DynamicChoice" name="testDynamicChoice">
      <prompt:parameters label="Select a dynamic option" order="170"
        localized="false">
```

```

    <prompt:additionalParams>
      <prompt:param name="sqlCommand"
        value="select Conversion, Value from alerts.conversions
        where Colname='Severity' order by Value desc;"/>
    </prompt:additionalParams>
  </prompt:parameters>
</prompt:prompt>

<!-- Multiline String prompt - useful for forced Journal entry -->
<prompt:prompt type="MultilineString" name="testMultilineString">
  <prompt:parameters label="Type in a journal entry" order="0"
    errorMessage="Journal entry cannot be empty" localized="false"
    defaultValue="Journal entry"/>
</prompt:prompt>

<!-- Formatted String prompt -->
<prompt:prompt type="FormattedString" name="testFormattedString">
  <prompt:parameters label="Type in the ticket ID (3 letters, 3 digits)"
    order="180" errorMessage="Invalid ticket ID format" localized="false"
    defaultValue="ABC123">
    <prompt:additionalParams>
      <prompt:param name="format" value="^[a-zA-Z]{3}\d{3}$"/>
    </prompt:additionalParams>
  </prompt:parameters>
</prompt:prompt>

<!-- Real-Time Dynamic Choice prompt - can impact server performance,
therefore use sparingly
-->
<prompt:prompt type="RealTimeDynamicChoice" name="testRealTimeDynamicChoice">
  <prompt:parameters label="Select a real-time dynamic option" order="190"
    localized="false">
    <prompt:additionalParams>
      <prompt:param name="sqlCommand" value="select Conversion, Value from
        alerts.conversions where Colname='Type' order by Value desc;"/>
    </prompt:additionalParams>
  </prompt:parameters>
</prompt:prompt>
</method>
</methodCall>

```

Create or replace a prompt:

The format of the <method> element for creating or replacing a prompt is:

```
<method methodName="prompt.createOrReplacePrompt">
```

Use this method to create a new prompt or replace an existing one. The <method> elements contains one or more <prompt:prompt> elements each of which defines the characteristics of a new prompt. The <prompt:prompt> element contains an optional <prompt:parameters> element followed by an optional <prompt:choice> element.

Example

This example replaces the definition of the existing prompt named testFormattedString.

```

<methodCall xmlns:prompt="http://www.ibm.com/tivoli/netcool/webtop/prompts/2.2">
  <method methodName="prompt.createOrReplacePrompt">
    <!-- testFormattedString exists, so it is replaced with new settings. -->
    <prompt:prompt type="FormattedString" name="testFormattedString">
      <prompt:parameters label="Type in the ticket ID (3 letters, 3 digits)"
        order="180" errorMessage="Invalid ticket ID format" localized="false"
        defaultValue="Bbc123">
        <prompt:additionalParams>
          <prompt:param name="format" value="^[a-zA-Z]{3}\d{3}$"/>
        </prompt:additionalParams>
      </prompt:parameters>
    </prompt:prompt>
  </method>
</methodCall>

```



```

        </prompt:additionalParams>
    </prompt:parameters>
</prompt:prompt>
</method>
</methodCall>

```

Related reference

“<prompt:prompt>” on page 164

“<prompt:parameters>” on page 164

“<prompt:choice>” on page 166

“Usage notes” on page 166

Modify a prompt:

The format of the <method> element for modifying a prompt is:

```
<method methodName="prompt.modifyPrompt">
```

Use this method to modify an existing prompt. The <method> elements contains one or more <prompt:prompt> elements each of which defines the characteristics of a new prompt. The <prompt:prompt> element contains an optional <prompt:parameters> element followed by an optional <prompt:choice> element.

Example

This example modifies the prompt named testFixedChoice.

```

<methodCall xmlns:prompt="http://www.ibm.com/tivoli/netcool/webtop/prompts/2.2">
  <method methodName="prompt.modifyPrompt">
    <prompt:prompt type="FixedChoice" name="testFixedChoice">
      <prompt:parameters label="Select a fixed option" order="160"
        localized="false" defaultValue="fixed6"/>
      <prompt:choice>
        <prompt:item value="fixed4" label="Fixed Four"/>
        <prompt:item value="fixed5" label="Fixed Five"/>
        <prompt:item value="fixed6" label="Fixed Six"/>
      </prompt:choice>
    </prompt:prompt>
  </method>
</methodCall>

```

Related reference

“<prompt:prompt>” on page 164

“<prompt:parameters>” on page 164

“<prompt:choice>” on page 166

“Usage notes” on page 166

Delete a prompt:

The format of the <method> element for deleting a prompt is:

```
<method methodName="prompt.deletePrompt">
```

Use this method to delete an existing prompt. The <method> element contains one or more <prompt:prompt> elements each of which identifies a metric to delete. In the <prompt:prompt> element include only the name attribute.

Example

This example deletes the prompt named testString.

```
<methodCall xmlns:prompt="http://www.ibm.com/tivoli/netcool/webtop/prompts/2.2">
  <method methodName="prompt.deletePrompt">
    <prompt:prompt name="testString"/>
  </method>
</methodCall>
```

Get a list of prompts:

The format of the <method> element for getting a list of prompts is:

```
<method methodName="prompt.getList" />
```

Use this method to obtain a list containing the names of the defined prompts.

Example

```
<methodCall xmlns:prompt="http://www.ibm.com/tivoli/netcool/webtop/prompts/2.2">

  <!-- Get a list of prompts available in the server. -->
  <method methodName="prompt.getList"/>
</methodCall>
```

CGI requests

CGI requests operate on CGI scripts used as tools in the Web GUI. There are functions to register, modify, and unregister a script.

WAAPI provides four methods for working with CGI scripts:

- “Register a CGI script”
- “Create or replace a CGI script” on page 173
- “Modify a CGI script” on page 173
- “Unregister a CGI script” on page 174

Register a CGI script:

The format of the <method> element for registering a CGI script is:

```
<method methodName="cgi.registerCGI">
```

Use this method to register a CGI script with the Web GUI server. The <method> element contains one or more <cgi> elements each defining a script to register.

<cgi>:

The <cgi> element defines the characteristics of a CGI script and has the following attributes:

Table 52. Attributes of the <cgi> element

Attribute name	Required or optional	Description
name	Required	Provides a unique name for a CGI script. Value: String Default value: None
useSmartPageCommands	Optional	Defines whether to use SmartPage commands when running the script. Value: true or false Default value: false

Table 52. Attributes of the <cgi> element (continued)

Attribute name	Required or optional	Description
fileName	Optional	The path name of the file that contains the script. Value: String Default value: If this attribute is omitted, the content of the <cgi> element contains the script.

Example:

This example registers a script named myping contained in a file named nco_myping.cgi.

```
<methodCall>
  <method methodName="cgi.registerCGI">
    <cgi name="myping" acl="*" useSmartPageCommands="true"
      fileName="/home/nco_myping.cgi">
    </cgi>
  </method>
</methodCall>
```

Create or replace a CGI script:

The format of the <method> element for creating or replacing a CGI script is:

```
<method methodName="cgi.createOrReplaceCGI">
```

Use this method to register a new CGI script with the Web GUI server or replacing the existing definition if it already exists. The <method> element contains one or more <cgi> elements each defining a script to register.

This example creates or replaces a script named myping1.

```
<methodCall>
  <method methodName="cgi.createOrReplaceCGI">
    <cgi name="myping1" acl="*" useSmartPageCommands="false"
      fileName="/home/nco_myping1.cgi">
    </cgi>
  </method>
</methodCall>
```

Related reference

"<cgi>" on page 172

Modify a CGI script:

The format of the <method> element for modifying a CGI script is:

```
<method methodName="cgi.modifyCGI">
```

Use this method to modify a previously registered CGI script. The <method> element contains one or more <cgi> elements each modifying the characteristics of a CGI script.

Example

This example changes the value of the useSmarypage attribute for the script named myping.

```

<methodCall>
  <method methodName="cgi.modifyCGI">
    <cgi name="myping" acl="example" useSmartPageCommands="false" >
    </cgi>
  </method>
</methodCall>

```

Related reference

“<cgi>” on page 172

Unregister a CGI script:

The format of the <method> element for unregistering a CGI script is:

```
<method methodName="cgi.unregisterCGI">
```

Use this method to unregister a previously registered CGI script. The <method> element contains one or more <cgi> elements each identifying a CGI script to unregister. In the <cgi> element, include only the name attribute.

Example

This example unregisters a script named myping.

```

<methodCall>
  <method methodName="cgi.unregisterCGI">
    <cgi name="myping" >
    </cgi>
  </method>
</methodCall>

```

Filter requests

Filter requests operate on filters used to display events in the Active Event List (AEL), Lightweight Event List (LEL), Table View, and on monitor boxes in the Event Dashboard. There are functions to create, modify, remove, and list filters. you can also set the default view.

WAAPI provides five methods for operating on filters:

- “Add a filter”
- “Create or replace a filter” on page 177
- “Modify a filter” on page 178
- “Delete a filter” on page 179
- “Get a list of filters” on page 179
- “Set the default view” on page 180

Add a filter:

The format of the <method> element for adding a filter is:

```
<method methodName="filter.addFilter">
```

Use this method to define a new filter for use on the AEL, LEL, Table View, and the Event Dashboard. The <method> element contains one or more <filter> elements each of which defines the characteristics of a new filter.

For a dependent filter, the <filter> element contains one or more instances of the <dependentlist> element.

`<filter>`:

The `<filter>` element defines the characteristics of a filter and has the following attributes:

Table 53. Attributes of the `<filter>` element

Attribute name	Required or optional	Description
name	Required	Provides a unique name for a filter. Value: String Default value: None
user	Optional	For user filters, this attribute identifies the users that the filter is associated with. The value of the attribute is a comma-separated list of User IDs. Value: List of User IDs Default value: None
sql	Optional	The value of the SQL WHERE clause that defines the fields to use and their values when filtering events. Value: A SQL WHERE clause Default value: None
mode	Optional	Defines whether the filter is a dependent filter. Include this attribute for dependent filters only. Value: dependent Default value: None
description	Optional	Defines a textual description of the filter. Value: String Default value: None
filtercollection	Optional	Specifies the name of a filter collection that the filter is a member of. Value: String Default value: None
type	Optional	Identifies the type of the filter. Value: global, system, or user Default value: None
view	Optional	Identifies the view associated with the filter. Value: String Default value: None
viewtype	Optional	Identifies the type of view. Value: global, system, or user Default value: None

Table 53. Attributes of the <filter> element (continued)

Attribute name	Required or optional	Description
datasource	Optional	Identifies the data source that supplies events for filtering. To specify more than one data source use a comma-separated list. This attribute is required when adding a filter. Value: String Default value: None
metriclabel	Optional	Defines the label to describe the aggregate statistic data displayed on monitor boxes in the Event Dashboard. Value: String Default value: Metric:
metricshow	Optional	Defines the calculation criteria for the metric value displayed on monitor boxes in the Event Dashboard. Value: Average, Count, Sum, Minimum, or Maximum Default value: Average
metricof	Optional	Defines the ObjectServer field to use in calculating the metric value displayed on monitor boxes in the Event Dashboard. Value: Field name Default value: Severity

<dependentlist>:

The <dependentlist> element identifies the list of filters that a dependent filter uses and has the following attributes.

Table 54. Attributes of the <dependentList> element

Attribute name	Required or optional	Description
type	Required	Identifies the type of filters in this list of dependent filters. Value: Global or System Default value: None
list	Required	Defines the filters associated with a dependent filter. The attribute's value contains a comma-separated list of one or more filter names. Value: List of filter names Default value: None

Examples:

Event filter

This example adds a filter named `exampleFilter1` that has the following characteristics:

- The SQL WHERE clause is `Severity>=0`
- The metric label is `Metric:`
- The name of the view associated with the filter is `advanced`
- The filter is a global filter
- The name of the data source is `NCOMS`

```
<methodCall>
  <method methodName="filter.addFilter">
    <filter name="exampleFilter1"
      sql="Severity ;&gt;=0"
      metriclabel="Metric : "
      view="advanced"
      type="global"
      datasource="NCOMS"
    />
  </method>
</methodCall>
```

Dependent filter

This example adds a dependent filter named `exampleFilter3` that has the following characteristics:

- The metric label is `Metric:`
- The name of the view associated with the filter is `advanced`
- The filter is a global filter
- The name of the data source is `NCOMS`
- The filter is dependent on the filters named `dependentfilter1` and `dependentfilter2`

```
<methodCall>
  <method methodName="filter.addFilter">
    <filter name="exampleFilter3"
      metriclabel="Metric : "
      view="advanced"
      type="global"
      mode="dependent"
      datasource="NCOMS">
      <dependentlist type="global" list="dependentFilter1,dependentFilter2"/>
    </filter>
  </method>
</methodCall>
```

Create or replace a filter:

The format of the `<method>` element for creating or replacing a filter is:

```
<method methodName="filter.createOrReplaceFilter">
```

Use this method to create a new filter or replace one that already exists. The `<method>` element contains one or more `<filter>` elements each of which defines the characteristics of a filter.

For a dependent filter, the `<filter>` element contains one or more instances of the `<dependentlist>` element.

This examples creates or replaces a filter named exampleFilter2 that has the following characteristics:

- The SQL WHERE clause is Severity>=1
- The metric label is My Metric:
- The name of the view associated with the filter is advanced
- The filter is a global filter
- The name of the data source is NCOMS

```
<methodCall>
  <method methodName="filter.createOrModifyFilter">
    <filter name="exampleFilter2"
      sql="Severity ;>=1"
      metriclabel="My Metric : "
      view="advanced"
      type="global"
      datasource="NCOMS"
    />
  </method>
</methodCall>
```

Related reference

"<dependentlist>" on page 176

"<filter>" on page 175

Modify a filter:

The format of the <method> element for modifying a filter is:

```
<method methodName="filter.modify">
```

Use this method to modify the characteristics of an existing filter. The <method> element contains one or more <filter> elements each of which defines the new characteristics of an existing filter. Include only attributes of the <filter> that correspond to the characteristics you want to change. When you omit an attribute the corresponding characteristic is unchanged.

For a dependent filter, the <filter> element contains one or more instances of the <dependentlist> element.

Example

This example modifies the filter named exampleFilter1 and makes the following changes to the filter's characteristics:

- The SQL WHERE clause becomes lastOccurrence >= getdate-1800
- The metric label becomes MyMetric:

In addition, the example adds the following characteristics to the filter:

- The view is a global view
- The metric value is the average of the SubDivision field

```
<methodCall>
  <method methodName="filter.modifyFilter">
    <filter name="exampleFilter1"
      sql="LastOccurrence ;>= getdate -1800"
      metriclabel="MyMetric:"
      metricshow="Average"
      metricof="SubDivision"
      view = "advanced"
      viewtype="global"
      type="global"
    />
  </method>
</methodCall>
```



```

        datasource="NCOMS"
    />
</method>
</methodCall>

```

Related reference

“<dependentlist>” on page 176

“<filter>” on page 175

Delete a filter:

The format of the <method> element for deleting a filter is:

```
<method methodName="filter.deleteFilter">
```

Use this method to delete an existing filter. The <method> element contains one or more <filter> elements each of which identifies a filter to delete. In the <filter> element, include the name and type attributes. In addition, for user filters, include the user attribute to define the users from whom the filter is to be removed.

Example

This example deletes the global filters name exampleFilter1, exampleFilter3, the user filter named exampleFilter2, and the global, dependent filters named dependentFilter1 and dependentFilter2. The user filter is removed from the User IDs webtopadminuser and root.

```

<methodCall>
  <method methodName="filter.deleteFilter">
    <filter name="exampleFilter1" type="global"/>
    <filter name="exampleFilter2" type="user" user="webtopadminuser,root"/>
    <filter name="exampleFilter3" type="global"/>
    <filter name="dependentFilter1" type="global"/>
    <filter name="dependentFilter2" type="global"/>
  </method>
</methodCall>

```

Related reference

“<filter>” on page 175

Get a list of filters:

The format of the <method> element for getting a list of filters is:

```
<method methodName="filter.getList" />
```

Use this method to obtain a list containing the names of the defined filters. The list contains a separate section for each type of filter (global, system, and user).

Example

```

<methodCall>
  <method methodName="filter.getList" />
</methodCall>

```

Set the default view:

The format of the <method> element for setting the default view is:

```
<method methodName="filter.setDefaultView">
```

Use this method to determine the default view associated with a filter. The <method> element contains one or more <filter> elements each of which identifies a filter to associate a default view with. In the <filter> element, include the name to identify the filter and the view and viewtype attributes to identify default view for the filter.

Example

This example associates the global view named Default with the filter named exampleFilter1:

```
<methodCall>  
  <method methodName="filter.setDefaultView">  
    <filter name="exampleFilter1" view="Default" viewtype="global"/>  
  </method>  
</methodCall>
```

Related reference

“<filter>” on page 175

Filter collection requests

Filter collections are logical groupings of filters that are typically used to for entity data migrated from Netcool/Webtop. There are functions to create, modify, delete, and list filter collections. In addition, you can add and delete filters from a collection, get a list of file collections, and set the default view.

WAAPI provides eight methods for operating on filter collections:

- “Create a filter collection”
- “Create or replace a filter collection” on page 181
- “Modify a filter collection” on page 182
- “Delete a filter collection” on page 183
- “Add a filter to a filter collection” on page 183
- “Delete a filter from a filter collection” on page 184
- “Get a list of filter collections” on page 184
- “Set the view for a filter collection” on page 184

Create a filter collection:

The format of the <method> element for creating a filter collection is:

```
<method methodName="filtercollection.createFilterCollection">
```

Use this method to define a new collection of filters. The <method> element contains one or more <filterCollection> elements each of which defines the characteristics of a new collection. Each of these elements contains one or more <filter> elements that identify filters to be part of the collection. You can specify only system and global filters that already exist in the system. The <filter> elements contain use only the name and type attributes to identify the filter.

Related reference

“<filter>” on page 175

`<filterCollection>`:

The `<filterCollection>` element defines the characteristics of a filter collection. The element contains one or more `<filter>` elements that identify the filters in the collection, using only the name and type attributes, and has the following attributes:

Table 55. Attributes of the `<filterCollection>` element

Attribute name	Required or optional	Description
name	Required	Provides a unique name for a filter collection. Value: String Default value: None
description	Optional	Defines a textual description of the filter collection. Value: String Default value: None
viewName	Optional	Identifies the view associated with the filter. Value: String Default value: None
viewType	Optional	Identifies the type of view. Value: global or system Default value: global

Example:

The following example creates a filter collection named `testcollection` that contains two filters named `Example_Critical` and `AllEvents`.

```
<methodCall>
  <method methodName="filtercollection.createFilterCollection">
    <filterCollection name="testcollection" viewName="Default"
      viewType="global" description="A collection of filters for testing">
      <filter name="Example_Critical" type="system"/>
      <filter name="AllEvents" type="global"/>
    </filterCollection>
  </method>
</methodCall>
```

Create or replace a filter collection:

The format of the `<method>` element for creating or replacing a filter collection is:

```
<method methodCall="filtercollection.createOrReplaceFilterCollection">
```

Use this method to define a new collection of filters or replace one that already exists. The `<method>` element contains one or more `<filterCollection>` elements each of which defines the characteristics of a collection. Each of these elements contains one or more `<filter>` elements that identify filters to be part of the collection. You can specify only system and global filters that already exist in the system. The `<filter>` elements contain use only the name and type attributes to identify the filter.

Example

The following example creates or replaces a filter collection named testcollection2.

```
<methodCall>
  <method methodName="filtercollection.createOrReplaceFilterCollection">
    <filterCollection name="testcollection2" viewName="Default"
      viewType="global" description="Another collection of filters for testing">
      <filter name="Escalated" type="global"/>
      <filter name="AllEvents" type="global"/>
    </filterCollection>
  </method>
</methodCall>
```

Related reference

"<filterCollection>" on page 181

"<filter>" on page 175

Modify a filter collection:

The format of the <method> element for modifying a filter collection is:

```
<method methodName="filtercollection.modifyFilterCollection">
```

Use this method to modify the characteristics of an existing collection of filters. The <method> element contains one or more <filterCollection> elements each of which defines the characteristics of a collection. Each of these elements contains one or more <filter> elements that identify filters to be part of the collection. You can specify only system and global filters that already exist in the system. The <filter> elements contain use only the name and type attributes to identify the filter.

Example

The following example modifies the filter collection named testCollection2.

```
<methodCall>
  <method methodName="filtercollection.modifyFilterCollection">
    <filterCollection name="testcollection2" viewName="Default" viewType="global"
      description="A new description for testcollection2">
      <filter name="TaskList" type="global"/>
      <filter name="NetcoolStatus" type="global"/>
    </filterCollection>
  </method>
</methodCall>
```

Related reference

"<filterCollection>" on page 181

"<filter>" on page 175

Delete a filter collection:

The format of the <method> element for deleting a filter collection is:

```
<method methodCall="filtercollection.deleteFilterCollection">
```

Use this method to delete an existing collection of filters. The <method> element contains one or more <filterCollection> elements each of which identifies a collection to delete. In the <filterCollection> element include only the name attribute.

Example

The following example deletes the filter collection named testCollection.

```
<methodCall>
  <method methodName="filtercollection.deleteFilterCollection">
    <filterCollection name="testcollection"/>
  </method>
</methodCall>
```

Related reference

"<filterCollection>" on page 181

Add a filter to a filter collection:

The format of the <method> element for adding a filter to a collection is:

```
<method methodCall="filtercollection.addFilter">
```

Use this method to add a filter to an existing collection of filters. The <method> element contains one or more <filterCollection> elements each of which identifies a collection. Each of these elements contains one or more <filter> elements that identify filters to add to the collection. You can specify only system and global filters that already exist in the system. The <filter> elements contain use only the name and type attributes to identify the filter.

Example

The following example adds filters named Last10Mins and Information to the collection named testcollection2.

```
<methodCall>
  <method methodName="filtercollection.addFilter">
    <filterCollection name="testcollection2">
      <filter name="Last10Mins" type="global"/>
      <filter name="Information" type="global"/>
    </filterCollection>
  </method>
</methodCall>
```

Related reference

"<filterCollection>" on page 181

"<filter>" on page 175

Delete a filter from a filter collection:

The format of the <method> element for deleting a filter from a collection is:

```
<method methodCall="filtercollection.deleteFilter">
```

Use this method to delete a filter to an existing collection of filters. The <method> element contains one or more <filterCollection> elements each of which identifies a collection. Each of these elements contains one or more <filter> elements that identify filters to delete from the collection. The <filter> elements contain use only the name and type attributes to identify the filters.

Example

The following example removes the filter named Information from the collection named testcollection2

```
<methodCall>
  <method methodName="filtercollection.deleteFilter">
    <filterCollection name="testcollection2">
      <filter name="Information" type="global"/>
    </filterCollection>
  </method>
</methodCall>
```

Related reference

“<filterCollection>” on page 181

“<filter>” on page 175

Get a list of filter collections:

The format of the <method> element for getting a list of filter collections is:

```
<method methodCall="filtercollection.getList">
```

Use this method to obtain a list of defined filter collections.

Example

```
<methodCall>
  <method methodName="filtercollection.getList />
</methodCall>
```

Set the view for a filter collection:

The format of the <method> element for setting the default view of a collection is:

```
<method methodName="filtercollection.setView">
```

Use this method to set the view associated with a filter. The <method> element contains one or more <filterCollection> elements each of which identifies a collection to associate a default view with. In the <filterCollection> element, include the name to identify the filter and the viewName and viewType attributes to identify the view for the filter.

Example

The following example sets the view for the filter collection named testcollection2.

```

<methodCall>
  <method methodName="filtercollection.setView">
    <filterCollection name="testcollection2" viewName="DefaultTable"
      viewType="global"/>
  </method>
</methodCall>

```

Related reference

“<filterCollection>” on page 181

Metric requests

Metric requests operate on performance metrics that the Web GUI displays in the form of gauges. There are functions to create, modify, replace, and delete metrics. You can also obtain a list of the currently defined metrics.

WAAPI provides five methods for working on metrics.

- “Create a metric”
- “Create or replace a metric” on page 189
- “Modify a metric” on page 190
- “Delete a metric” on page 191
- “Get a list of metrics” on page 191

Create a metric:

The format of the <method> element for creating a metric is:

```
<method methodName="metric.create">
```

Use this method to define a new metric for use on a gauge. The <method> element contains one or more <metric:metric> elements each defining the characteristics of a new metric.

Each <metric:metric> element contains one instance of the <metric:command> element. In turn, the <metric:command> element contains a <metric:text> element.

<metric:metric>:

The <metric:metric> element defines the characteristics of a metric and has the following attributes:

Table 56. Attributes of the <metric:metric> element

Attribute name	Required or optional	Description
name	Required	Provides a unique name for a metric. Value: String Default value: None
description	Optional	Defines a textual description of the metric that appears when the user hovers the mouse pointer over the gauge. To include the current value of the metric in the description, use {0} at the appropriate place. Value: String Default value: None

Table 56. Attributes of the <metric:metric> element (continued)

Attribute name	Required or optional	Description
descriptionKey	Optional	A key for the description attribute that uniquely identifies it among all defined metrics. Omit this attribute to allow the system to automatically generate a unique key for you. Value: String Default value: A system-generated value.
displayName	Optional	The name of the metric as it appears beneath a gauge on the Gauges page. Value: String Default value: None
displayNameKey	Optional	A key for the displayName attribute that uniquely identifies it among all defined metrics. Omit this attribute to allow the system to automatically generate a unique key for you. Value: String Default value: A system-generated value
units	Optional	Defines the units that the metric displays. For example the number of incidents that have occurred, or the number of client connections to a server. Value: String Default value: None
unitsKey	Optional	A key for the units attribute that uniquely identifies it among all defined metrics. Omit this attribute to allow the system to generate a unique key for you. Value: String Default value: A system-generated value
maxValue	Optional	Defines the maximum value that the metric can have, expressed in terms of the metric's units. Value: Integer Default value: None
minValue	Optional	Defines the minimum value that the metric can have, expressed in terms of the metric's units. Value: Integer Default value: None

Table 56. Attributes of the <metric:metric> element (continued)

Attribute name	Required or optional	Description
threshold1	Optional	Defines the threshold between the low and medium ranges of the metric. The attribute defines the threshold as a percentage of the metrics value range and its value must be lower than threshold2. Value: Integer Default value: None
threshold2	Optional	Defines the threshold between the medium and high ranges of the metric. The attribute defines the threshold as a percentage of the metric's value range and its value must be higher than threshold1. Value: Integer Default value: None

Each <metric::metric> can contain one instance of <metric:command>.

<metric:command>:

The <metric:command> element defines the command that generates values for the metric. Typically this is a SQL command that queries the ObjectServer database. The element can have the following attribute:

Table 57. Attributes of the <metric:command> element

Attribute name	Required or optional	Description
type	Optional	Defines the type of command that generates values for the metric. Currently the only valid value for this attribute is sql. Value: String Default value: sql
mode	Optional	Defines the mode to use when the metric is in use. In basic mode, any restriction filters active on the specified database tables are applied. In advanced mode any restriction filters are not applied when the user accesses the metric using a gauge. Value: basic or advanced Default value: advanced

If present, the <metric:command> element contains one instance of the <metric:text> element.

`<metric:text>`:

The `<metric:text>` element contains the command that generates values for the metric. The element has the following attribute:

Table 58. Attributes of the `<metric:text>` element

Attribute name	Required or optional	Description
data	See Description	Contains the text of the command used to generate values for the metric. This attribute is required when the mode attribute of the <code><metric:command></code> element is omitted or has a value of advanced. Value: String Default value: None
selectField	See Description	Contains the table field or an aggregate function of the SELECT clause of the command used to generate values for the metric. This attribute is required when the mode attribute of the <code><metric:command></code> element has a value of basic. Value: String Default value: None
whereClause	See Description	Contains the WHERE clause of the command used to generate values for the metric. This attribute is required when the mode attribute of the <code><metric:command></code> element has a value of basic. Value: String Default value: None
databaseName	See Description	Contains the database name clause of the command used to generate values for the metric. This attribute is required when the mode attribute of the <code><metric:command></code> element has a value of basic. Value: String Default value: None
tableName	See Description	Contains the table name clause of the command used to generate values for the metric. This attribute is required when the mode attribute of the <code><metric:command></code> element has a value of basic. Value: String Default value: None

Examples:

The following example creates a metric named `metricsample1` in advanced mode. The metric measures the number of critical events outstanding and has the following characteristics:

- A minimum value of 0 and a maximum of 10000
- A low to medium threshold of 30%
- A medium to high threshold of 70%

```
<methodCall xmlns:metric="http://www.ibm.com/tivoli/netcool/webtop/metrics/7.3.1">
  <method methodName="metric.createMetric">
    <metric:metric name="metricsample1"
      displayName="MetricSample1"
      description="Critical events. Current value: {0}"
      units="events"
      maxValue="10000"
      minValue="0"
      threshold1="30"
      threshold2="70">
      <metric:command type="sql">
        <metric:text data="select sum(Tally) from alerts.status where Severity=5;"/>
      </metric:command>
    </metric:metric>
  </method>
</methodCall>
```

The following example creates a similar metric in basic mode. So any restriction filters that are active when the user accesses the metric are applied.

```
<methodCall>
  <method methodName="metric.createMetric">
    <metric:metric name="metricsample1"
      displayName="MetricSample1"
      description="Critical events. Current value {0}"
      units="events"
      maxValue="10000"
      minValue="0"
      threshold1="30"
      threshold2="70">
      <metric:command type="sql" mode="basic">
        <metric:text selectField="sum(Tally)" whereClause="Severity=5"
          databaseName="alerts" tableName="status"/>
      </metric:command>
    </metric:metric>
  </method>
</methodCall>
```

Create or replace a metric:

The format of the `<method>` element for creating or replacing a metric is:

```
<method methodName="metric.createOrReplaceMetric">
```

Use this method to replace an existing metric or create a new one if it does not already exist. The `<method>` element contains one or more `<metric:metric>` elements each of which defines the characteristics of a new metric. Each `<metric:metric>` element contains a `<metric:command>` element which, in turn, contains a `<metric:text>` element.

Example

The following example creates or replaces a metric named `metricsample2` in advanced mode. The new metric measures the number of major events outstanding and has the following characteristics:

- A minimum value of 0
- A maximum value of 100
- A low to medium threshold of 40%
- A medium to high threshold of 80%

```
<methodCall xmlns:metric="http://www.ibm.com/tivoli/netcool/webtop/metrics/7.3.1">
  <method methodName="metric.createOrReplaceMetric">
    <metric:metric name="metricsample2"
      displayName="MetricSample2"
      description="Major events. Current value: {0}"
      units="events"
      maxVal="100"
      minVal="0"
      threshold1="40"
      threshold2="80">
      <metric:command type="sql">
        <metric:text data="select sum(Tally) from alerts.status where Severity=4;"/>
      </metric:command>
    </metric:metric>
  </method>
</methodCall>
```

To create or replace a metric in basic mode, use the same format of the `<metric:command>` and `<metric:text>` elements shown in the example of creating a metric in basic mode.

Related reference

“`<metric:metric>`” on page 185

“`<metric:command>`” on page 187

“`<metric:text>`” on page 188

Modify a metric:

The format of the `<method>` element for modifying a metric is:

```
<method methodName="metric.modifyMetric">
```

Use this method to modify the characteristics of an existing metric. The `<method>` element contains one or more `<metric:metric>` elements each of which defines the new characteristics of an existing metric. Include only attributes of the `<metric:metric>` that correspond to the characteristics you want to change. When you omit an attribute the corresponding characteristic is unchanged.

To modify the command associated with the metric, include a `<metric:command>` element which, in turn, contains the `<metric:text>` element.

Example

This example modifies the metric named `metricsample1` and makes the following changes to the metric's characteristics:

- The text of the description is enhanced
- The maximum value changes from 10000 to 250
- The low to medium threshold changes from 30% to 40%

- The medium to high threshold changes from 70% to 90%

```
<methodCall xmlns:metric="http://www.ibm.com/tivoli/netcool/webtop/metrics/7.3.1">
  <method methodName="metric.modifyMetric">
    <metric:metric name="metricsample1"
      displayName="MetricSample1"
      description="Critical events. Modified by WAAPI. Current value: {0}"
      units="events"
      maxValue="250"
      minValue="0"
      threshold1="40"
      threshold2="90">
      <metric:command type="sql">
        <metric:text data="select sum(Tally) from alerts.status where Severity=5;" />
      </metric:command>
    </metric:metric>
  </method>
</methodCall>
```

To modify a metric in basic mode, use the same format of the `<metric:command>` and `<metric:text>` elements shown in the example of creating a metric in basic mode.

Related reference

“`<metric:metric>`” on page 185

“`<metric:command>`” on page 187

“`<metric:text>`” on page 188

Delete a metric:

The format of the `<method>` element for deleting a metric is:

```
<method methodName="metric.deleteMetric">
```

Use this method to delete an existing metric. The `<method>` element contains one or more `<metric:metric>` elements each of which identifies a metric to delete. In the `<metric:metric>` element include only the name attribute.

Example

This example deletes the metric named `metricsample2`.

```
<methodCall xmlns:metric="http://www.ibm.com/tivoli/netcool/webtop/metrics/7.3.1">
  <method methodName="metric.deleteMetric">
    <metric:metric name="metricsample2" />
  </method>
</methodCall>
```

Related reference

“`<metric:metric>`” on page 185

Get a list of metrics:

The format of the `<method>` element for getting a list of metrics is:

```
<method methodName="metric.getList" />
```

Use this method to obtain a list containing the names of the defined metrics.

Example

```
<methodCall xmlns:metric="http://www.ibm.com/tivoli/netcool/webtop/metrics/7.3.1">
  <method methodName="metric.getList" />
</methodCall>
```

Other requests

There are three miscellaneous functions to resynchronize the Web GUI cache, to remove a node from a cluster, to generate a status report, and to reload the server's filters and views.

WAAPI provides four other functions:

- "Resynchronize the Web GUI cache with the ObjectServer's database"
- "Remove a node from a load balancing cluster"
- "Generate a system status report"
- "Reload filters and views" on page 193

Resynchronize the Web GUI cache with the ObjectServer's database:

The format of the `<method>` element for resynchronizing the Web GUI cache with the ObjectServer's database is:

```
<method methodName="osresync.refreshOSCache" />
```

Use this method to force the Web GUI to resynchronize its cache with the ObjectServer's database. This function is especially useful when a user's ObjectServer permissions have changed and it ensures that the Web GUI has the latest information and so can manage the user correctly.

Example

```
<methodCall>  
  <method methodName="osresync.refreshOSCache" />  
</methodCall>
```

Remove a node from a load balancing cluster:

The format of the `<method>` element for removing a node from a cluster is:

```
<method methodName="cluster.removeNode">
```

Use this method to remove a node from a load balancing cluster. The method removes the node that you send this request to. If this is the last node in the cluster, the method also removes all the load balancing configuration data.

Example

```
<methodCall>  
  <method methodName="cluster.removeNode" />  
</methodCall>
```

Generate a system status report:

The format of the `<method>` element for generating a system status report is:

```
<method methodName="webtopprobe.generateReport">
```

Use this method to generate a status report for the Web GUI. The generated report contains the same information as the **Troubleshooting and support > System Information for Tivoli Netcool/OMNIBus Web GUI** function of the Web GUI generates and contains the following sections:

- Version numbers of the Web GUI, the Dashboard Common Infrastructure (DCI), and Java Runtime Environment
- Memory usage statistics
- Runtime platform information

- Protocols in use
- ObjectServer properties and configuration data, including information about the cache
- All system properties, including those internal to the Web GUI

Example

```
<methodCall>
  <method methodName="webtopprobe.generateReport" />
</methodCall>
```

Reload filters and views:

The format of the `<method>` element to reload filters and views is:

```
<method methodName="xml dao.reloadFiltersAndViews">
```

Use this method to force the Web GUI to reload all currently active filters and views. In addition, this method forces AEL clients to update their filters and views during the next refresh.




Example

```
<methodCall>
  <method methodName="xml dao.reloadFiltersAndViews" />
</methodCall>
```

Installing the WAAPI client on a remote host


Additionally to the WAAPI client on the Web GUI server, you can install the WAAPI client on a remote host.

To install the WAAPI client on a remote host:

1. Log in to the Web GUI using any suitable user ID.
2. On the navigation pane, click **Welcome** and then click **WAAPI Client Information** in the work area.
3. Download the WAAPI client:
 -   Click **waapi.tar.gz**
 -  Click **waapi.zip**

Save the file to the directory where you want to install the client.

4. Change to the directory in which you want the files to be installed.
5. Uncompress the file.

Tip:  Use the **gtar xzvf waapi.tar.gz** command or the **gunzip waapi.tar.gz** and **tar -xvf waapi.tar** commands.

The files are installed in a subdirectory called `waapi` with the following structure:

- `waapi/`
- `waapi/bin`
- `waapi/etc/`
- `waapi/etc/default`
- `waapi/etc/docs`
- `waapi/etc/samples/`
- `waapi/log/`

- waapi/platform/java/bin
 - waapi/platform/java/lib
6. **UNIX** Set the execute-permissions on the **runwaapi** command:
- ```
chmod +x waapi/bin/runwaapi
```

## Results

You can now use the WAAPI client on the remote host.

## WAAPI security

WAAPI has a number of security features that help to ensure secure communication with the Web GUI server.

Administration of any system relies upon security. WAAPI provides a number of security features that you can use:

- A secure connection to the Web GUI server.  
Consider using an SSL connection, with or without FIPS-142 protection, when running the WAAPI client on a server remote from the Web GUI server.
- Encrypting WAAPI passwords.  
You can encrypt WAAPI passwords using either AES or FIPS-142 (this is required when the connection is secured using FIPS-142). You can use this feature whether the WAAPI client is installed with or remote from the Web GUI server.
- Securing the waapi.init file  
The properties file for the WAAPI client contains a number of sensitive entries. For example, there is the username and password of the user to use on the Web GUI server to run WAAPI commands. So it is important to ensure that the file is available only to authorized administrators.

WAAPI provides the following security tasks:

- “Creating secure WAAPI connections”
- “Enabling WAAPI password encryption” on page 200
- “Securing the waapi.init properties file” on page 202

### Creating secure WAAPI connections

You can configure the Web GUI to use the Secure Sockets Layer (SSL) protocol for secure WAAPI communication, using either server-only authentication, or server-and-client authentication. Optionally, you can enable the FIPS 140-2 mode.

The following WAAPI connection modes can be configured for Web GUI:

#### SSL not enabled

The WAAPI client connects to the Web GUI via standard HTTP. This mode does not require any additional configuration.

#### SSL enabled, server-only authentication (without FIPS 140-2)

The WAAPI client connects to the Web GUI via HTTPS using server-only authentication, but not in FIPS 140-2 mode.

#### SSL enabled, server-and-client authentication (without FIPS 140-2)

The WAAPI client connects to the Web GUI via HTTPS using server-and-client authentication, but not in FIPS 140-2 mode.



### SSL enabled, server-only authentication with FIPS 140-2

The WAAPI client connects to the Web GUI via HTTPS using server-only authentication in FIPS 140-2 mode.

### SSL enabled, server-and-client authentication with FIPS 140-2

The WAAPI client connects to the Web GUI via HTTPS using server-and-client authentication in FIPS 140-2 mode.

### Creating a WAAPI SSL connection (server-only authentication):

To create a secure, server-authenticated connection between WAAPI and the Web GUI deployed within Tivoli Integrated Portal (without FIPS 140-2), you must reference Tivoli Integrated Portal in the WAAPI truststore as well as the `waapi.init` file.

To create a secure, server-authenticated connection between WAAPI and the Web GUI:

1. Using the Tivoli Integrated Portal GUI, extract the default truststore signer certificate.
  - a. Click **Settings > WebSphere Admin Console**, and click **Launch WebSphere Admin Console**.
  - b. Click **Security > SSL certificate and key management > Key stores and certificates > NodeDefaultKeyStore > Personal certificates**.
  - c. Select the default (Alias) truststore certificate and click **Extract**.
  - d. Type a name, for example, `/example/tipcert.arm`.
  - e. Select **Base64-encoded ASCII data** and click **Ok**.
2. Using the Tivoli Integrated Portal Ikeyman utility, add the new certificate to the WAAPI truststore.
  - a. Go to `tip_home_dir/bin` and start Ikeyman.
  - b. Click **KeyDatabaseFile > New** and select **PKCS** as the key database type.
  - c. Type a truststore name, for example `/example/waapiTruststore.p12`.
  - d. Enter the default password WebAS and click **Ok**.
  - e. Select **Signer Certificates** from the dropdown list and click **Add**.
  - f. Point to the signer certificate, in this example `/example/tipcert.arm`, and click **Ok**. Make a note of the signer certificate CN (common name) value.
3. Edit the `waapi.init` file.
  - a. Open `webgui_home_dir/waapi/etc/waapi.init` and go to the WAAPI Secure Modes section.
  - b. Set **`waapi.secure:on`**.
  - c. Ensure that the host name in **`waapi.host`** is the same as the CN (common name) value in the signer certificate.
  - d. Provide the truststore name, in this example `/example/waapiTruststore.p12`.
  - e. Enter the password WebAS.

### What to do next

To test if you have successfully set up the WAAPI SSL connection, execute a WAAPI example.

### Creating a WAAPI SSL connection (client-server authentication):

To create a secure, client- and server-authenticated connection between WAAPI and the Web GUI deployed within Tivoli Integrated Portal (without FIPS 140–2), you reference Tivoli Integrated Portal in the WAAPI truststore and WAAPI in the Tivoli Integrated Portal truststore. You then enable SSL authentication in WAAPI and add the WAAPI keystore certificate to your browser's truststore. Lastly, you enable client authentication in Tivoli Integrated Portal.

1. Using the Tivoli Integrated Portal GUI, extract the default truststore signer certificate.
  - a. Click **Settings > WebSphere Admin Console**, and click **Launch WebSphere Admin Console**.
  - b. Click **Security > SSL certificate and key management > Key stores and certificates > NodeDefaultKeyStore > Personal certificates**.
  - c. Select the default (Alias) truststore certificate and click **Extract**.
  - d. Type a name, for example, */example/tipcert.arm*.
  - e. Select **Base64-encoded ASCII data** and click **Ok**.
2. Using the Tivoli Integrated Portal Ikeyman utility, add the new certificate to the WAAPI truststore.
  - a. Go to *tip\_home\_dir/bin* and start Ikeyman.
  - b. Click **KeyDatabaseFile > New** and select **PKCS** as the key database type.
  - c. Provide a truststore name, for example */example/waapiTruststore.p12*.
  - d. Enter the default password WebAS and click **Ok**.
  - e. Select **Signer Certificates** from the dropdown list and click **Add**.
  - f. Point to the signer certificate, in this example */example/tipcert.arm*, and click **Ok**. Make a note of the signer certificate CN (common name) value.
3. Using the Tivoli Integrated Portal Ikeyman utility, extract a self-signed personal keystore certificate from the WAAPI keystore.
  - a. Go to *tip\_home\_dir/bin* and start Ikeyman.
  - b. Click **KeyDatabaseFile > New** and select **PKCS** as the key database type.
  - c. Provide a keystore name, for example *waapiKeystore.p12*.
  - d. Enter the default password WebAS and click **Ok**.
  - e. Select **Personal Certificates** from the dropdown list and click **New Self-Signed**.
  - f. Enter a key label, for example *WAAPI\_cert*, complete the other fields as required, then click **Ok**.
  - g. Select the new keystore certificate, in this example *WAAPI\_cert*, and click **Extract Certificate**.
  - h. Select **Base64-encoded ASCII data**.
  - i. Enter a certificate file name, for example *WAAPI\_cert.arm*, and define a location, in this example */example/*, then click **Ok**.
4. Using the Tivoli Integrated Portal GUI, add the new WAAPI keystore certificate to the Tivoli Integrated Portal truststore.
  - a. Click **Settings > WebSphere Admin Console**, and click **Launch WebSphere Admin Console**.
  - b. Click **Security > SSL certificate and key management > Key stores and certificates > NodeDefaultTrustStore > Signer certificates**.
  - c. Click **Add** and enter an alias of *WAAPI\_cert* (for this example).
  - d. Point to the previously-generated *WAAPI\_cert*, click **Ok**, then **Save**.

5. Using your browser's security management functionality, add the new keystore certificate to the browser's truststore.  
**Warning:** If you do not complete this step, you will no longer be able to access Tivoli Integrated Portal after you enable client authentication in the next step.
6. Using the Tivoli Integrated Portal GUI, enable client authentication.
  - a. Click **Settings > WebSphere Admin Console**, and click **Launch WebSphere Admin Console**.
  - b. Click **Security > SSL certificate and key management > SSL Configurations > NodeDefaultSSLSettings > Quality of protection (QoP) settings**.
  - c. Select **Required** from the **General Properties > Client authentication** drop-down list.
  - d. Click **Ok**, then **Save**.
7. Edit the `waapi.init` file.
  - a. Open `webgui_home_dir/waapi/etc/waapi.init` and go to the WAAPI Secure Modes section.
  - b. Set **`waapi.secure:on`**.
  - c. Ensure that the host name in `waapi.host` is the same as the CN (common name) value in the signer certificate.
  - d. Provide the keystore name, in this example `/example/waapiKeystore.p12`.
  - e. Provide the truststore name, in this example `/example/waapiTruststore.p12`.
  - f. Enter the password of WebAS.

**Note:** Windows When entering the location of keystore and truststore on a Windows system, use two backslashes as the path separator because a single backslash is interpreted as an escape character. For example to specify the truststore use `\\example\\waapiTruststore.p12`.

## What to do next

To test if you have successfully set up the WAAPI SSL connection, execute a WAAPI example.

## Creating a WAAPI SSL connection with FIPS 140-2 (server-only authentication):

To create a secure, server-authenticated connection between WAAPI and the Web GUI deployed within Tivoli Integrated Portal with FIPS 140-2 mode enabled, you must reference Tivoli Integrated Portal in the WAAPI truststore as well as the `waapi.init` file. You must then enable FIPS 140-2 in Tivoli Integrated Portal.

If you have already enabled FIPS 140-2 in Tivoli Integrated Portal while setting up FIPS 140-2 for the Web GUI, you do not need to complete step four.

1. Using the Tivoli Integrated Portal GUI, extract the default truststore signer certificate.
  - a. Click **Settings > WebSphere Admin Console**, and click **Launch WebSphere Admin Console**.
  - b. Click **Security > SSL certificate and key management > Key stores and certificates > NodeDefaultKeyStore > Personal certificates**.
  - c. Select the default (Alias) truststore certificate and click **Extract**.
  - d. Give it a name, for example, `/example/tipcert.arm`.
  - e. Select **Base64-encoded ASCII data** and click **Ok**.

2. Using Tivoli Integrated Portal's Ikeyman utility, add the new certificate to the WAAPI truststore.
  - a. Go to *tip\_home\_dir/bin* and start Ikeyman.
  - b. Click **KeyDatabaseFile > New** and select **PKCS** as the key database type.
  - c. Provide a truststore name, for example */example/waapiTruststore.p12*.
  - d. Enter the default password WebAS and click **Ok**.
  - e. Select **Signer Certificates** from the dropdown list and click **Add**.
  - f. Point to the signer certificate, in this example */example/tipcert.arm*, and click **Ok**. Make a note of the signer certificate CN (common name) value.
3. Edit the *waapi.init* file.
  - a. Open *webgui\_home\_dir/waapi/etc/waapi.init* and go to the WAAPI Secure Modes section.
  - b. Set **waapi.secure:fips**.
  - c. Ensure that the host name in *waapi.host* is the same as the CN (common name) value in the signer certificate.
  - d. Provide the truststore name, in this example */example/waapiTruststore.p12*.
  - e. Enter the password of WebAS.
4. Enable FIPS 140-2 in the Tivoli Integrated Portal server.
  - a. Open *webgui\_home\_dir/java/jre/lib/security/java.security*.
  - b. In the list of providers and their order of preference, uncomment *security.provider:<x>=com.ibm.crypto.fips.provider.IBMJCEFIPS*
  - c. Replace the *<x>* variable with 1 and re-number the subsequent security providers.
  - d. Using the Tivoli Integrated Portal GUI, enable FIPS 140-2. Click **Security > SSL Certificate and key management** and select the **FIPS** checkbox under Configuration Settings, then click **Apply**.
  - e. Restart the Tivoli Integrated Portal server.

### Example

#### What to do next

To test if you have successfully set up the WAAPI SSL connection, execute a WAAPI example.

#### Creating a WAAPI SSL connection with FIPS 140-2 (client-server authentication):

To create a secure, client- and server-authenticated connection between WAAPI and the Web GUI deployed within Tivoli Integrated Portal with FIPS 140-2 mode enabled, you reference Tivoli Integrated Portal in the WAAPI truststore and WAAPI in the Tivoli Integrated Portal truststore. You then enable FIPS 140-2 authentication in WAAPI and add the WAAPI keystore certificate to your browser's truststore. Lastly, you enable client authentication and FIPS 140-2 in Tivoli Integrated Portal.

If you have already enabled FIPS 140-2 in Tivoli Integrated Portal while setting up FIPS 140-2 for the Web GUI, you do not need to complete step eight.

1. Using the Tivoli Integrated Portal GUI, extract the default truststore signer certificate.

- a. Click **Settings > WebSphere Admin Console**, and click **Launch WebSphere Admin Console**.
  - b. Click **Security > SSL certificate and key management > Key stores and certificates > NodeDefaultKeyStore > Personal certificates**.
  - c. Select the default (Alias) truststore certificate and click **Extract**.
  - d. Give it a name, for example, */example/tipcert.arm*.
  - e. Select **Base64-encoded ASCII data** and click **Ok**.
2. Using the Tivoli Integrated Portal Ikeyman utility, add the new certificate to the WAAPI truststore.
    - a. Go to *tip\_home\_dir/bin* and start Ikeyman.
    - b. Click **KeyDatabaseFile > New** and select **PKCS** as the key database type.
    - c. Provide a truststore name, for example */example/waapiTruststore.p12*.
    - d. Enter the default password WebAS and click **Ok**.
    - e. Select **Signer Certificates** from the dropdown list and click **Add**.
    - f. Point to the signer certificate, in this example */example/tipcert.arm*, and click **Ok**. Make a note of the signer certificate CN (common name) value.
  3. Using Tivoli Integrated Portal's Ikeyman utility, extract a self-signed personal keystore certificate from the WAAPI keystore.
    - a. Go to *tip\_home\_dir/bin* and start Ikeyman.
    - b. Click **KeyDatabaseFile > New** and select **PKCS** as the key database type.
    - c. Provide a keystore name, for example *waapiKeystore.p12*.
    - d. Enter the default password WebAS and click **Ok**.
    - e. Select **Personal Certificates** from the dropdown list and click **New Self-Signed**.
    - f. Enter a key label, for example *WAAPI\_cert*, complete the other fields as required, then click **Ok**.
    - g. Select the new keystore certificate, in this example *WAAPI\_cert*, and click **Extract Certificate**.
    - h. Select **Base64-encoded ASCII data**.
    - i. Enter a certificate file name, for example *WAAPI\_cert.arm*, and define a location, in this example */example/*, then click **Ok**.
  4. Using the Tivoli Integrated Portal GUI, add the new keystore certificate to the Tivoli Integrated Portal truststore.
    - a. Click **Security > SSL certificate and key management > Key stores and certificates > NodeDefaultTrustStore > Signer certificates**.
    - b. Click **Add** and enter an alias of *WAAPI\_cert* (for this example).
    - c. Point to the previously generated *WAAPI\_cert*, click **Ok**, then **Save**.
  5. Using your browser's security management functionality, add the new keystore certificate to the browser's truststore.  
**Warning:** If you do not complete this step, you will no longer be able to access Tivoli Integrated Portal after you enable client authentication in the next step.
  6. Using the Tivoli Integrated Portal GUI, enable client authentication.
    - a. Click **Security > SSL certificate and key management > SSL Configurations > NodeDefaultSSLSettings > Quality of protection (QoP) settings**.
    - b. Select **Required** from the **General Properties > Client authentication** drop-down list.
    - c. Click **Ok**, then **Save**.

7. Edit the `waapi.init` file.
  - a. Open `webgui_home_dir/waapi/etc/waapi.init` and go to the WAAPI Secure Modes section.
  - b. Set **`waapi.secure:fips`**.
  - c. Ensure that the host name in **`waapi.host`** is the same as the CN (common name) value in the signer certificate.
  - d. Provide the truststore name, in this example `/example/waapiTruststore.p12`.
  - e. Enter the password of WebAS.
8. Enable FIPS 140-2 in the Tivoli Integrated Portal server.
  - a. Open `tip_home_dir/java/jre/lib/security/java.security`.
  - b. In the list of providers and their order of preference, uncomment `security.provider:<x>=com.ibm.crypto.fips.provider.IBMJCEFIPS`
  - c. Replace the `<x>` variable with 1 and re-number the subsequent security providers.
  - d. Using the Tivoli Integrated Portal GUI, enable FIPS 140-2. Click **Security > SSL Certificate and key management** and select the **FIPS** checkbox under Configuration Settings, then click **Apply**.
  - e. Restart the Tivoli Integrated Portal server.

### What to do next

To test if you have successfully set up the WAAPI SSL connection, execute a WAAPI example.

## Enabling WAAPI password encryption

You can opt to store WAAPI passwords in the `waapi.init` file in encrypted format.

For password encryption for non-SSL and SSL connections, you use AES encryption, while for FIPS 140-2 connections, you use FIPS 140-2 mode encryption. The encryption types, and the tools required to enable them, are as follows:

### Non-SSL (HTTP) connections

Passwords can be AES encrypted using the **`ncw_aes_crypt`** tool.

### SSL (HTTPS) connections

Passwords can be AES encrypted using the **`ncw_aes_crypt`** tool.

### SSL (HTTPS) connections with FIPS 140-2 enabled

Passwords can be encrypted using the **`ncw_fips_crypt`** tool.

WAAPI passwords must be encrypted using the **`ncw_fips_crypt`** script in `webgui_home_dir/waapi/bin`. This script uses the vault key in `waapi_install_dir/etc/encrypt`. If it does not already exist, the `vault.key` file is automatically generated on the first execution of the script.

## Encrypting WAAPI passwords using AES:

To encrypt WAAPI passwords for non-SSL and SSL connections, use the **ncw\_aes\_crypt** tool. You can configure WAAPI password encryption for WAAPI clients installed on the same server as Tivoli Integrated Portal or on a different server.

### Before you begin

You can use AES password encryption only if FIPS 140–2 mode has not been enabled.

The default truststore password is WebAS.

The location of the `waapi.init` file, in which you must enter the encrypted password, differs depending on where the WAAPI client is installed. If the WAAPI client is installed on the same server as Tivoli Integrated Portal, the file is located at `webgui_home_dir/waapi/etc/waapi.init`. If the WAAPI client is installed on a different server than Tivoli Integrated Portal, the file is located at `waapi_install_dir/etc/waapi.init`.

1. Encrypt the WAAPI password:
  - a. Run `webgui_home_dir/waapi/bin/ncw_aes_crypt`.
  - b. Enter the default Tivoli Integrated Portal truststore password WebAS.  
An encrypted password is generated.
  - c. Copy the encrypted password.
2. Add the encrypted password:
  - a. Open the `waapi.init` file.
  - b. Set the **waapi.password.encryption** property to `aes`.
  - c. Set the **waapi.ssl.trustStorePassword** property to the password encrypted in step 1.
3. Repeat steps 2b and 2c for the following properties:
  - **waapi.password**
  - **waapi.ssl.keyStorePassword**

## Encrypting WAAPI passwords using FIPS 140–2 mode encryption:

To encrypt WAAPI passwords for non-SSL and SSL connections in FIPS 140–2 mode, use the **ncw\_fips\_crypt** tool. You can configure WAAPI password encryption whether the WAAPI client is installed on the same server as Tivoli Integrated Portal or on a different server.

### Before you begin

If FIPS 140–2 mode has been enabled, you can use only FIPS 140–2 mode password encryption. You must have IBM JRE installed to use the **ncw\_fips\_crypt** tool.

The default truststore password is WebAS.

The default WAAPI vault (secret) key used is located in `webgui_home_dir/waapi/etc/encrypt/vault.key`. The vault key is automatically generated on first use of the **ncw\_fips\_crypt** tool and stored in the `waapi_install_dir/etc/encrypt/vault.key` file.



The location of the `waapi.init` file, in which you must enter the encrypted password, differs depending on where the WAAPI client is installed. If the WAAPI client is installed on the same server as Tivoli Integrated Portal, the file is located at `webgui_home_dir/waapi/etc/waapi.init`. If the WAAPI client is installed on a different server than Tivoli Integrated Portal, the file is located at `waapi_install_dir/etc/waapi.init`.

1. Encrypt the WAAPI password:
  - a. Enter the following command:  

```
webgui_home_dir/waapi/bin/ncw_fips_crypt -password WebAS -key
webgui_home_dir/waapi/etc/encrypt/vault.key
```

  
If you use the default vault key, omit the **key** parameter. An encrypted password is generated.
  - b. Copy the encrypted password.
2. Add the encrypted password:
  - a. Open the `waapi.init` file.
  - b. Set the **waapi.password.encryption** property to `fips`.
  - c. Set the **waapi.ssl.trustStorePassword** property to the password generated in step 1.
3. Repeat steps 2b and 2c for the following properties:
  - **waapi.password**
  - **waapi.ssl.keyStorePassword**

#### What to do next

To generate a new vault key, use the **-genkey** parameter. Enter the following command: `webgui_home_dir/waapi/bin/ncw_fips_crypt -genkey <locationofvaultkeyfile>`. After the command has run, copy the new vault key file to the `waapi_install_dir/etc/encrypt/` directory.

### Securing the `waapi.init` properties file

Protect the `waapi.init` properties file from access by unauthorized users.

Use the facilities of the operating system to set the access permissions to the `waapi.init` file. Ensure that the file is accessible only to authorized administrators of the Web GUI server. This is especially important when the file contains the username and password of the Web GUI administrator.

---

## Troubleshooting

Consult these troubleshooting notes to help determine the cause of the problem and what to do about it.

### Checking system information

To determine the status of the server and client, and to help you with troubleshooting, access the system information for the Web GUI.

To access the system information, in the navigation pane, click **Troubleshooting and Support > System Information for Tivoli Netcool/OMNIBus Web GUI**.

### Results

The following information is displayed:

- Web GUI server version



- Java Runtime Environment
- Memory usage
- Operating system information
- Web GUI server properties
- ObjectServer properties
- System properties

## Viewing TIPProfile logs for login errors

In the event of a login error, review the system outage and system error logs to help determine the cause.

Follow these steps to open the system outage and system error logs:

1. At the command line, change to the *tip\_home\_dir*/profiles/TIPProfile/logs/server1 directory.
2. Open SystemOut.log and SystemErr.log in a text editor. On Windows, for example, the command notepad systemout.log opens the log in Windows Notepad.
3. Review the errors.
4. If the cause and solution to your login error is not apparent, send the SystemOut.log and SystemErr.log from this directory and the server1\_exception.log (and any other files that were modified within a few minutes of this one) from the sibling ffdc directory to your security administrator for further examination.

### Related tasks

“Viewing the application server profile” on page 2

## Editing a properties file

Properties files describe the environment and their settings are usually predefined or added during installation. You do not need to change these files unless instructed by IBM Software Support.

The properties files are on the computer where the Tivoli Integrated Portal Server is installed.

1. Locate the *tip\_home\_dir*/properties directory. For example, C:\IBM\tivoli\tipv2\properties is the default installation path on Windows; /opt/IBM/tivoli/tipv2/ is the default installation path on Linux or UNIX.
2. Open the desired properties file in a text editor.
3. Edit the file as needed, and then save and close it.
4. Stop the application server, and then restart it.

## Setting the log level

The log level determines the amount of detail that the Web GUI writes to the log files. As supplied, the product logs all events of the Information severity and above. Use these procedures to change the level if required.

You can change the log in either of the following ways:

- Setting the log level in the server initialization file
- Using the WebSphere administration console

## Related tasks

“Restarting the server” on page 1

## Editing the server initialization file

To change the logging level in the server initialization file:

1. Log in to the machine running the Web GUI using a command line interface.
2. Open *webgui\_home\_dir/etc/server.init* in a text editor.
3. Locate the **log.level** property.
4. Change its value to the required minimum severity level to write to the log files

The available log levels, in ascending order of ascending severity, are:

FINEST  
FINER  
PROFILE  
FINE  
CONFIG  
INFO  
WARNING  
SEVERE

5. Save the file.
6. Restart the Tivoli Integrated Portal server.

## Using the WebSphere administration console

You can also use the WebSphere administration console. Use this method when diagnosing an issue with the help of IBM Technical Support. To set the logging level using the WebSphere administration console:

1. Log in to the Web GUI as an administrative user.
2. Click **Settings > WebSphere Admin Console**.
3. On the Web Sphere Admin Console page, click **Launch WebSphere Admin Console**.

The console opens in a new browser window.

4. Click **Troubleshooting > Logs and Trace**.
5. In the Logging and Tracing page, click the name of the Tivoli Integrated Portal (for example, server1).
6. Click **Change Log Detail Levels** and click the **Runtime** tab.
7. Expand the tree of components under **All Components**, expand the tree under **com.ibm.tivoli.\*** and then the tree under **com.ibm.tivoli.ncw.\***.
8. For each of the elements that technical support have advised you to change the log level:
  - a. Click the name of the element.
  - b. Click **Message and Trace Levels** and choose the required minimum severity level to write to the log files.
9. Click **OK** and close the WebSphere administration console window.

## Generating performance logs

A performance log can help to identify bottlenecks in your system configuration. Use these procedures to create a performance log if required.

You can generate a performance log in either of the following ways:

- Setting the trace level in the server initialization file
- Using the WebSphere administration console

### Related tasks

“Setting the log level” on page 203

## Editing the server initialization file

Follow the instructions in “Editing the server initialization file” on page 204 in “Setting the log level” on page 203 to set **trace.level** to PROFILE.

## Using the WebSphere administration console

Follow the instructions in “Using the WebSphere administration console” on page 204 in “Setting the log level” on page 203 to set the log level to FINER.

## Setting a trace

Enable a trace of the Tivoli Integrated Portal Server when you want to keep a record of activity.

### Before you begin

The portal has a Troubleshooting Logs and Trace option for enabling a trace.

Follow these steps to set a trace that will record the Tivoli Integrated Portal Server actions in a log file: *tip\_home\_dir/profiles/TIPProfile/logs/server/trace.log*.

1. In the portal, click **Troubleshooting > Logs and Trace**.
2. Select the Tivoli Integrated Portal Server name (such as server1) in the Logging and Tracing portlet.
3. In the **Configuration** tab, click **Change Log Detail Levels**.
4. In the **Groups** list, expand **com.ibm.tivoli.\*** and click **com.ibm.tivoli.tip.\***.
5. Select a log level (such as **All Messages and Traces**) and click **OK** or **Apply**.
6. When prompted to save the configuration, click **Save**.
7. Stop and restart the Tivoli Integrated Portal Server:
  - a. In the *tip\_home\_dir/profiles/TIPProfile/bin* directory, depending on your operating system, enter one of the following commands:
    - **Windows** stopServer.bat server1
    - **UNIX** **Linux** stopServer.sh server1
  - Note:** On UNIX and Linux systems, you are prompted to provide an administrator username and password.
  - b. In the *tip\_home\_dir/profiles/TIPProfile/bin* directory, depending on your operating system, enter one of the following commands:
    - **Windows** startServer.bat server1
    - **UNIX** **Linux** startServer.sh server1

## Results

After the server has been stopped and restarted, trace entries are saved to the *tip\_home\_dir/profiles/TIPProfile/logs/server1/trace.log* file.

### Related tasks

“Stopping and starting the application server” on page 1

## Auditing the usage of objects

How to set up auditing of objects such as maps, filters, and views to determine which are being used.

To enable auditing of objects in the Web GUI:

1. Navigate to the following directory:

```
tip_home_dir/profiles/TIPProfile/config/cells/TIPCell/nodes/TIPNode/servers/server1
```

2. Edit *server1.xml*

3. Locate the `<services>` element with an `xmi:type` attribute of `loggingservice.http:HTTPAccessLoggingService`. For example:

```
<services xmi:type="loggingservice.http:HTTPAccessLoggingService"
 xmi:id="HTTPAccessLoggingService_1183077764084"
 enable="false" enableErrorLogging="true"
 enableAccessLogging="true">
 <errorLog xmi:id="LogFile_1183077764084"
 filePath="${SERVER_LOG_ROOT}/http_error.log"
 maximumSize="500"/>
 <accessLog xmi:id="LogFile_1183077764085"
 filePath="${SERVER_LOG_ROOT}/http_access.log"
 maximumSize="500"/>
</services>
```

4. Change the value of the `enable` attribute of the `<services>` element to `true`.
5. Add the following attributes to the `<services>` element:  
`errorLogLevel="DEBUG" accessLogFormat="COMBINED"`
6. Locate the 4 `<transportChannels>` elements that have an `xmi:type` attribute with a value `channel.service.channels:HTTPInboundChannel`.
7. In each of these elements, set the value of the `enableLogging` attribute to `true`.
8. Save the file.
9. Restart the server.

## Results

After restarting the server, page navigation is recorded in the following log files:

```
http_access.log
```

```
http_error.log
```

Both files are in *tip\_home\_dir/profiles/TIPProfile/logs/server1*.

### Related tasks

“Restarting the server” on page 1

## Web GUI log files

The Web GUI has several log files providing information on various aspects of its operation, as described here.

The following table describes the Web GUI log files.

Table 59. Log files

| Log file                                                                                                                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|---------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>webgui_home_dir/integration/migration_tool/log/migration.log</code>                                                 | <p>The migration log. Contains information relating to the migration process, if you have migrated to the Tivoli Netcool/OMNIBus Web GUI V7.3.1 from a previous version. If any errors caused the migration to fail at any point, these are listed in the migration log file. Any files, directories, or properties that could not be migrated are also listed in the migration log file.</p> <p>You can set the level of detail in this log in <code>webgui_home_dir/integration/migration_tool/etc/logging.properties</code> file.</p> |
| <code>tip_home_dir/profiles/TIPProfile/logs</code> and subdirectories                                                     | Several Tivoli Integrated Portal log files are contained in this directory.                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <code>tip_home_dir/profiles/TIPProfile/logs/ncw/ncw.n.log</code><br><br>In the file name a number replaces <i>n</i> .     | Contains event and user monitoring information. You can set the level of detail in this log in the initialization file.                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <code>tip_home_dir/profiles/TIPProfile/logs/ncw/ncw.n.profile</code><br><br>In the file name a number replaces <i>n</i> . | Contains performance metrics. You can set the level of detail in this log in the initialization file.                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <code>tip_home_dir/profiles/TIPProfile/logs/ncw/ncw.n.trace</code><br><br>In the file name a number replaces <i>n</i> .   | Contains trace information. You can set the level of detail in this log in the initialization file.                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <code>tip_home_dir/profiles/TIPProfile/logs</code> and subdirectories                                                     | <p>For more information about server logs, go to the IBM WebSphere® Application Server Information Center at the following Web address and search for <i>server logs</i>.</p> <p><a href="http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp">http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp</a>,</p>                                                                                                                                                                                                      |
| <code>tip_home_dir/profiles/TIPProfile/logs/dci-common/OMNIBusWebGUI/dci-security.log</code>                              | DCI Security component log file. You can set the level of detail in this log in <code>webgui_home_dir/etc/dci-common/security/dci-security.properties</code>                                                                                                                                                                                                                                                                                                                                                                             |

## Harmless authentication messages

Certain sign-on messages are routine and might not indicate that a problem has occurred.

For installations that have been configured to use the Tivoli Integrated Portal authentication service, it is possible that an authentication client receives CTGES1504E and CTGES1505E messages. These messages are generated when an unused single sign-on LTPA token is discarded, and might be insignificant.

An authentication client attempts to use all single sign-on tokens provided to it when authenticating to an authentication service. Some of these tokens might not apply to the configured authentication service, causing CTGES1504E and CTGES1505E messages to be generated on the client and CTGES1089E on the server. When not accompanied by other CTGES0008E authentication client errors, these messages indicate only that a particular single sign-on token was discarded.

## No user role assigned

Users cannot log in to the console if they do not have user roles assigned.

If you have a valid user ID and password but get a message that the login failed and to try again (even after successive retries), ask your administrator to review your role assignments.

Every user ID must be assigned the minimum roles necessary to log in to the application server: Monitor, Configurator, Operator, Deployer, or Administrator. Users should also have the minimum required product level roles or they might not see the contents of their default product pages after logging in.

## Slow network response

Performance issues can cause an unresponsive script message to display after login.

If, immediately after logging in, you get a message about an unresponsive script and you are asked whether to continue or cancel opening the Web page, click **Continue**. After a short time, the welcome page for the console is displayed.

Such messages can indicate a slow network link between your computer and the application server. Ping the server computer to see the round trip response time. Use response times of 40 ms or better.

Try using a remote desktop connection to a computer that has a better response time with the application server and logging in from there.

Consider using a caching HTTP proxy to improve speed and reduce network traffic.

## AEL fails to start and displays message W0025

Checks to make should the Active Event List (AEL) fails to start and the message identified as W0025 appears.

If there is a problem with the connection to the ObjectServer or an incorrect JVM or browser installed, the AEL may fail to start and the following message is displayed:

W0025 The event list is unavailable. Contact your administrator.

Should that message appear on a user's web browser, carry out the following checks:

1. Make sure that the ObjectServer is running.  
If not, start it and ask the user to try again.
2. Make sure that user's machine can connect to the ObjectServer.
3. Make sure that the user's machine has one of the supported web browsers,
4. Make sure that the latest support version of the Java Virtual Machine (JVM) is installed on the user's machine and that it is appropriate for the web browser in use.

## Event Dashboard unable to get data from server

For certain monitor boxes the Event Dashboard displays the error message: Unable to get data from the server. In addition, the Event Dashboard fails to display some filters correctly.

The Event Dashboard displays the following error message for certain monitor boxes:

Unable to get data from the server

Clicking on one of those monitor boxes causes the following error message to appear:

The event list is unavailable

However, the default Active Event List (AEL) can successfully display events.

This problem arises because the Event Dashboard fails to display some filters correctly. This is due to a mismatch between the data source name provided for these filters and the data source names defined in `ncwDataSourceDefinitions.xml`.

1. Carry out one of the following actions to correct this mismatch
  - Use the Filter Builder to correct the name in the relevant filters.
  - Edit `ncwDataSourceDefinitions.xml` to use the correct data source name.
2. Restart the server.

### Related tasks

"Setting up filters for event data" on page 286

"Restarting the server" on page 1

## Maps are slow to display severity colors

Filters in maps are slow to show severity colors. This may be accompanied by high CPU usage on the Web GUI server.

If the performance of the Web GUI is otherwise satisfactory, there are three likely causes of this condition:

- “Frequency of map refresh”
- “Result caching”
- “Map complexity”

### Frequency of map refresh

The frequency that a map refreshes is defined in two ways:

- The **maplet.refresh** property in the `server.init` file. This sets a global refresh rate for the system.
- The portlet preferences for individual maps.

In either case, make sure that the refresh frequency is no lower than 10 seconds. For complex maps, use a higher value.

### Result caching

Use the `<result-cache>` element in the data source configuration file for data sources that provide the map with data. This can help to reduce the load on the data sources.

### Map complexity

The complexity of a map is the most likely cause of this condition. Rather than use a small number of maps that contain many filters, limit the number of filters on a map to no more than 30 to 40. Then drill down from these filters to further maps.

#### Related tasks

“Setting preferences for the Map portlet” on page 237

#### Related reference

Appendix B, “Data source configuration file data reference,” on page 389

## Changing the height setting of a Map portlet has no effect

Changing the height of a map portlet using the **Height** box or the **Use Customizer** check box in the portlet preferences has no effect. The map always renders to the same size.

The **Height** and **Use Customizer** do not override the height of the Map portlet. That is controlled by the height settings available in the Map Editor. To set the height of a map, set the **Use Customizer** portlet preference, and configure the height of the map in the Map Editor.

#### Related tasks

“Setting preferences for the Map portlet” on page 237

“Customizing maps” on page 317



## The connection to the ObjectServer is lost or the AEL issues a timeout

If the connection to the ObjectServer persistently is lost or the AEL issues a timeout, adjust the query timeout setting for the datasource associated with the ObjectServer.

1. Edit `ncwDataSourceDefinitions.xml`.
2. Locate the `<ncwConnectionParameters>` element for the appropriate data source, and within that element, locate the `<ncwQueryTimeout>` element.
3. Increase the value of the `baseTime` attribute to increase the period of the timeout.
4. Restart the server.
5. If the problem persists, repeat this procedure to increase the timeout further.

### Related tasks

"Restarting the server" on page 1

### Related reference

Appendix B, "Data source configuration file data reference," on page 389

---

## Performance tuning tips for the Web GUI

Use this information to troubleshoot performance in the Web GUI.

### Increasing JVM memory on the Web GUI server

To improve Web GUI performance, you can increase the amount of memory on the server.

To increase the amount of memory available to the Java Virtual Machine (JVM), carry out the following steps:

1. Manually stop the application server.
2. Change to the `tip_home_dir/profiles/TIPProfile/bin` directory.
3. Use the **wsadmin** command to increase the heap size for the JVM, as follows:  
`wsadmin.sh -lang jython -conntype NONE`
4. At the `wsadmin>` prompt, issue the following commands, where `xxx` is the new heap size value, in megabytes.

```
jvm=AdminConfig.list("JavaVirtualMachine")
```

```
AdminConfig.modify(jvm, '[[initialHeapSize xxx]]')
```

```
AdminConfig.modify(jvm, '[[maximumHeapSize xxx]]')
```

```
AdminConfig.save()
```

```
exit
```

5. Clean the `server1` and `tnm` logs and restart the Tivoli Integrated Portal Server. The changes take effect when the Tivoli Integrated Portal Server is restarted.

**Attention:** If you attempt to start the Tivoli Integrated Portal Server with a maximum heap size that is too large, error messages that are similar to the following are generated in the `tip_home_dir/profiles/TIPProfile/logs/server1/native_stderr.log` file:

```
JVMJ9GC019E -Xms too large for -Xmx
```

```
JVMJ9VM015W Initialization error for library j9gc23(2): Failed to initialize
Could not create the Java virtual machine.
```

## Increasing memory on Web GUI client Java Virtual Machines

To increase event volumes in the Active Event List (AEL), or if a `java.lang.OutOfMemoryError` error occurs on the AEL, modify the heap sizes of the Java Virtual Machine (JVM) on the Web GUI client.

A `java.lang.OutOfMemoryError` error might occur during failover from the primary ObjectServer to the backup ObjectServer. This error might also occur if, depending on the configuration of your views, an AEL is running more than 20,000 events. If this error occurs, the following message is displayed in the AEL:

Entity Unavailable

By increasing the initial and maximum heap sizes, you can solve this error.

To increase heap sizes:

1. On the Web GUI client, open the Control Panel for the Java Plug-In.
2. Click **Advanced**.
3. In the field under **Java Runtime Parameters**, modify the heap sizes. For example:  
`-Xms64m -Xmx128m`
4. Click **Apply** and close the Control Panel.
5. Log out of the Web GUI and log back in.

### What to do next

Verify that the error no longer occurs. If the error persists then increase the heap size further, for example to `-Xms256m -Xmx512m`

## Performance tuning

To improve the throughput of events from a data source to the Web GUI, or to distribute the load between the ObjectServer and the Web GUI server, you can adjust several configuration settings.

**Important:** Several factors, such as the availability of system memory, event load and the number of products running in your network, influence performance. Consider these factors when changing the configuration settings of the Web GUI because they might negate any performance benefits achieved by changing the Web GUI configuration.

The following parameters in the `webgui_home_dir/etc/server.init` file can be modified to tune performance:

#### **ael.top-n.value**

This property limits the number of events to be displayed in any given Active Event List (AEL). Limiting this value, for example to 3000, might improve the time required to load or refresh an AEL.

#### **aelview.queries.enabled**

This property enables the creation of transient AEL views, but it also increases the load on the Web GUI. If you do not use this functionality, disable this property.

#### **maplet.refresh**

This property controls the frequency with which Web GUI maps are refreshed. If you lower the value of this property, for example to 10, event throughput might be increased. If you increase the value of this property, for example to

30, the load on the Web GUI server might be reduced. Consider this property alongside results caching in the `ncwDataSourceDefinitions.xml` file.

The following elements in the `ncwDataSourceDefinitions.xml` file can be modified to tune performance. These elements are child elements of the `<results-cache>` element.

#### **<chart>**

This property controls caching for Web GUI charts. If you set the **enabled** attribute to true, the Web GUI server caches the results of SQL queries for a configurable period of time. This reduces the load on the ObjectServer, but may increase the memory requirements of the Web GUI server, in order to store the cached data.

#### **<eventList>**

This property controls caching for event list results, that is, for the AEL, Lightweight Event List (LEL) and Table View. If you set the **enabled** attribute to true, the Web GUI server caches the results of SQL queries for a configurable period of time. This reduces the load on the ObjectServer, but may increase the memory requirements of the Web GUI server, in order to store the cached data.

#### **<eventSummary>**

This property controls caching for event summary results, that is, for the Event Dashboard and map pages. If you set the **enabled** attribute to true, the Web GUI server caches the results of SQL queries for a configurable period of time. This reduces the load on the ObjectServer, but may increase the memory requirements of the Web GUI server, in order to store the cached data.

#### **<ncwOSConnection>**

This property has two attributes that control the pool of JDBC connections to the data source: `minPoolSize` and `maxPoolSize`. By default, `minPoolSize` has a value of 5 and `maxPoolSize` a value of 10. To improve throughput, you can increase the values of these attributes. This increases the number of parallel requests to the data source and reduces the occurrence of requests being blocked through lack of an available connection.

#### **<metric>**

This property controls caching for the event metrics that provide information for the gauges on Gauges pages. If you set the **enabled** attribute to true, the Web GUI server caches the results of SQL queries for a configurable period of time. This reduces the load on the ObjectServer, but may increase the memory requirements of the Web GUI server, in order to store the cached data.

The following parameter in the `webgui_home_dir/etc/system/userdefaults.props` file can be modified to tune performance:

#### **ael\_user\_properties\_refresh\_time**

This property controls the default AEL refresh frequency for new users. To improve event throughput, set the value of this property as low as possible. The lowest configurable value is 30. Individual users can override this value by setting their AEL user preferences.

#### **Related tasks**

“Restricting the number of rows displayed in the Active Event List” on page 243

---

## CGI support

Use the initialization parameters to control the behavior of CGIServlet.

### CGIServlet

CGI scripts run on a Web server and use the Common Gateway Interface (CGI) to perform tasks. The support for CGI in Tivoli Integrated Portal is provided by *CGIServlet*, extracted from Apache Tomcat. The Tomcat CGI support is largely compatible with the Apache HTTP Server but there are some limitations (such as only one cgi-bin directory). To change the configuration, edit `web.xml` in the directory where the CGI application is installed.

### Servlet initialization parameters

Several initialization parameters are available for configuring the behavior of the CGIServlet.

#### **cgiPathPrefix**

The CGI search path will start at the Web application root directory + `File.separator` + this prefix. Default setting: `cgiPathPrefix` is `Web-INF/cgi`.

**debug** Determines the level of debugging detail for messages that are logged by the servlet. Default setting: 0.

#### **executable**

This is type of the program to be used to run the script. Default setting: `perl`.

#### **parameterEncoding**

Names the parameter encoding to be used with the CGI servlet. Default setting: `System.getProperty("file.encoding","UTF-8")`.

#### **passShellEnvironment**

Determines whether shell environment variables, if there are any, shall be passed to the CGI script. Default setting: `false`.

---

## Chapter 2. Administering a load balancing cluster

Tasks for administering the Web GUI in a load balancing environment.

For instructions on how to set up a load balancing cluster, add nodes to a cluster, and remove nodes from a cluster, refer to the *IBM Tivoli Netcool/OMNIBus Installation and Deployment Guide*.

For instructions on how to set up a load balancing cluster, add nodes to a cluster, and remove nodes from a cluster see Setting up and configuring a load balancing environment.

### Related concepts

“The Web GUI in a load balancing environment” on page 93

---

### The Web GUI in a load balancing environment

Information on how the Web GUI can operate in a load balancing environment and the implications for administering and using the product.

A load balancing environment consists of a group of Web GUI servers that are linked together and operate as a single server. The name for the group of servers is a *cluster* and each of the servers is known as a *node*.

The primary benefits of a cluster are as follows:

- Load balancing – where the workload of servicing user requests is spread among the nodes. This improves the overall performance of the system.
- Availability – to maintain the availability of network monitoring even if some cluster nodes are unavailable for any reason (for example, they are shut down for maintenance).

The following sections contain more information on clusters, on administering them, and on using them:

- “Structure of a cluster” on page 93
- “Configuration data” on page 93
- “Updating configuration data” on page 94
- “Conditions necessary for changing configuration data” on page 95
- “Administering a load balancing cluster” on page 96
- “Using a load-balanced cluster” on page 97

### Related concepts

“Administering a load balancing cluster” on page 92

## Structure of a cluster

A cluster consists of a group of Web GUI servers, an HTTP server, and a DB2 database.

- The servers carry out service requests from users. In addition, each server is configured to trust the other servers in the cluster and is able to communicate with all the other cluster members. This enables them to cooperate as a single unit.
- The HTTP server distributes user HTTP sessions among the servers. It allocates requests among the servers either randomly or in a round robin fashion. The method that the HTTP server uses depends on how it was set up when installed.
- The DB2 database holds the configuration data for the cluster.

## Configuration data

Configuration data defines how a Web GUI server operates. It is held differently in a cluster than it is in a standalone server.

A standalone Web GUI server holds its configuration data in the local file system. In a cluster, the DB2 database holds the configuration data for the entire cluster. This is the master copy of the data that is shared by all the cluster nodes. A single set of configuration data means that each node is configured identically. There is no configuration data that is specific to a cluster node.

Although the database holds the master copy, each node also has a copy in its local file system. This is for fault tolerance reasons and allows the cluster to continue operation should the configuration database become unavailable during operation. When a node starts it reads a complete set of configuration data from the database into the local file system and loads it into memory to improve performance.

The configuration data that the database holds includes:

- Data sources
- Users, groups, and roles
- Page layouts, customized page information, and portlet descriptors
- Deployment descriptors
- Filters and views
- All items in the configstore:
  - AEL menus and menu configuration data
  - Metrics for gauges
  - Prompts
  - Tools
  - User preferences
- AEL preferences such as the refresh time and the number of rows to display
- Web GUI properties such as the default time zone and the timeout period
- Maps and resources, together with their properties
- Gauges and their properties
- Charts and their properties
- Predictive eventing information
- TADDM events
- Access information for the Inline frame portlet

## Updating configuration data

Changes to the configuration data need to be coordinated across the entire cluster irrespective of which node initiated the changes.

The configuration data can change in any of three ways:

- The facilities of the Web GUI itself (for example, setting a set of preferences for a portlet)
- By editing the configuration files directly (for example, setting the metrics for a gauge)
- By using WAAPI commands (for example, enabling predictive eventing)

A change could originate on any cluster node. However, this change needs to be propagated to the entire cluster to maintain the commonality of the cluster's configuration.

## Updating the database

The process for changing configuration data is as follows:

1. A user on a node changes an item in the configuration and requests that the node saves the change.
2. The node writes the new information (for example a configuration file) to the database.
3. The node notifies all the other cluster nodes that there is revised configuration information.
4. The node updates its local copy of the configuration data to reflect the change.
5. The other cluster nodes read the new information from the database and update the copies in their local file systems.
6. The cluster continues to operate with the new configuration settings.

## Detecting changes to the configuration files

It is not always necessary to restart the cluster or any of its nodes to pick up the new configuration information. Instead, revised configuration data is automatically applied when it occurs. This is achieved through:

- The Web GUI timed tasks facility
- A file that lists the files to be monitored and an associated set of monitor processes

Timed tasks determine when each node loads changed files from the database.

The file is named *webgui\_home\_dir/etc/system/stores.lst* and contains a list of all the configuration files that are kept in the database. When a node starts or joins the cluster it creates a set of processes that monitor each of the files listed in *stores.lst*. Whenever a change occurs to one of those files, the corresponding process propagates the changed file to the DB2 database and notifies other nodes of the change.

This file monitoring capability means that an individual component of the Web GUI (such as a portlet) does not need to know whether configuration information is maintained in a database or in the local file system. Instead, the component always writes changes to its configuration directly to the local file system. The monitoring processes take care of updating the database.

There are some exceptions where a restart of a node, and usually the cluster, is necessary. Changing any of the following files requires a restart of the server:

- `server.init`
- `ncwDataSourceDefinitions.xml`

## Conditions necessary for changing configuration data

To be able to operate correctly, certain conditions need to be met before the cluster can allow changes to its configuration data.

For the cluster to operate successfully, the DB2 database must be available. The database is the key coordination point of the cluster because it contains the configuration data.

If the database becomes unavailable after the cluster has started, operations continue, with each node using their local copy of the configuration data. However, each node prevents any changes occurring to the configuration data. This state continues until the DB2 database becomes available again. At this point, the cluster nodes refresh their locally stored configuration from the database and allow changes to configuration data to take place once more. The policy of allowing changes to occur only when the database is available helps to ensure that the cluster remains synchronized and that common behavior is maintained across the cluster.

When a node starts and joins the cluster, it reads the configuration data from the database, even though it may have data in its local file system. Nodes do this to ensure that they always have the latest set of configuration data. If the database is unavailable when a node starts, it cannot continue because it cannot be sure that the local copy of the configuration data is up to date.

In addition to the configuration data, each cluster node must be run the same version of the Web GUI, with the same set of features, and set up in the same way. As with the data, this is the only way to provide a common service to the users of the Web GUI.

## Administering a load balancing cluster

Administration of a load balancing cluster has two aspects that you need to be aware of:

- “Day-to-day administration” on page 96
- “Cluster administration” on page 96

### Day-to-day administration

In day-to-day administration, bear in mind that any change you make always applies to the entire cluster not just the node where you make the change. For instance, adding a user to one node adds that user to all nodes. There may be a short time delay before a change is applied to all the nodes. This depends on how often the timed tasks interval is set, and how long it is until the next execution of the timed tasks facility.

An advantage of this propagation of configuration data is that it simplifies your administration job. You need only to make each change once, and the cluster ensures that all nodes receive it. If the database is unavailable, you cannot make any change to the configuration data. When using the Web GUI itself, the system prevents you from saving any changes to the data. When editing files or using



WAAPI, the node you are using will not propagate the changed information until the database becomes available once more.

## Cluster administration

After set up, a cluster requires little administration over and above the day-to-day administration that any Web GUI installation requires. However, the Web GUI provides a comprehensive set of tools for you to administer the cluster. These tools enable you to do the following:

- Enable load balancing after installation
- Administer the timed tasks facility
- Add and remove nodes
- Resynchronize a cluster node
- Export configuration information from a test environment into production
- Maintain the list of files to be monitored and propagated to the database whenever they change

### Related tasks

“Administering timed tasks” on page 7

## Using a load-balanced cluster

To users, the Web GUI behaves almost identically in a clustered environment as a standalone server. All that many users might notice, after a move to a clustered environment, is an increased responsiveness of the product. This is due to the overall increases in performance that the cluster provides.

---

## Maintaining the list of files to monitor

On each cluster member has a copy of a file named `stores.lst` that holds a list of files to monitor for changes. When any of these files changes, the monitoring process copies it to the database.

You can add further files to the list for monitoring and saving to the database. Carry out the following procedure on each member of the cluster.

**Important:** You can add further files to store in the database only. Do not modify or remove any of the supplied entries in the file. Doing so adversely affects the operation of the cluster.

1. In a text editor, open the file `webgui_home_dir/etc/system/stores.lst`.
2. Add entries for any other directories that you want to include in the database. Specify all directories relative to the path: `webgui_home_dir/etc/configstore`.
3. Save the file and exit from the text editor.

The revised content of the file is copied to the database and propagated to all nodes in the cluster.

### Related reference

“Load balancing best practices” on page 98

---

## Cluster administration tools

Use the cluster tools to administer the members of the cluster.

### Monitoring a load balancing cluster

If synchronized data fails to be committed to a node in the cluster, that node should be removed from the cluster for corrective action. Use the diagnosis tool to identify any unsynchronized nodes in the load balancing cluster.

To determine if changes to global data are not committed to any of the nodes, use the **HATool** command script to check the synchronization of modules and repositories on the nodes in a cluster. For the HATool, you must provide the DB2 administrator's credentials.

#### Query synchronization of modules

Use this command to determine if all nodes have identical sets of modules deployed.

```
HATool.bat/sh modules username password -byNodes -showAll
```

The following parameters are optional.

- **-byNodes**

Specifies that the results of the command are ordered by the node in the cluster. This parameter is optional. The default is to list the results by module.

- **-showAll**

Specifies that all modules and nodes in the cluster should be returned. This parameter is optional. The default is to return only modules for unsynchronized nodes.

#### Query the synchronization of global repositories

Use this command to determine if all repositories are synchronized on all nodes.

```
HATool.bat/sh repositories username password -byNodes -showAll
```

The following parameters are optional.

- **-byNodes**

Specifies that the results of the command are ordered by the node in the cluster. This parameter is optional. The default is to list the results by repository.

- **-showAll**

Specifies that all modules and nodes in the cluster should be returned. This parameter is optional. The default is to return only repositories for unsynchronized nodes.

#### Release the global lock

Use this command to manually release the global lock placed on all of the console nodes when the cluster is in maintenance mode. This command is used when a node cannot commit a change during synchronization and has to be taken offline.

```
HATool.bat/sh release-lock username password
```

---

## Load balancing best practices

When administering the Web GUI in a load balancing environment, there are a number of practices you can use to avoid problems occurring in the cluster.

### Overview

Administrative items that need special attention in a load balancing environment are:

- “Timed tasks” on page 98
- “The configuration database” on page 99
- “The list of files to maintain in the database” on page 99
- “Custom web content” on page 99

### Timed tasks

Timed tasks are an essential element in the smooth running of a load balancing cluster. They ensure that all changes to files in *webgui\_home\_dir/etc/configstore/* and its subdirectories are detected and loaded into the server, without the need to restart the Web GUI server.

As a minimum, ensure that the **timedtasks.enabled** property in *theserver.init* file is set to true.

In most cases the schedules supplied for filters and views and for other components are adequate. However, you can change the schedules to suit your specific needs. If you do this in a load balancing environment, you are recommended to create identical schedules for the same set of components on all nodes in the load balancing cluster.

### The configuration database

A load balancing cluster uses a database to hold the configuration data. Individual nodes in the cluster hold only a copy of this data, primarily for performance reasons. The master copy of the configuration data is always the one in the database.

Always make sure that the database is available before making any changes to the Web GUI configuration. This is especially important if you are modifying the configuration files directly, such as defining the metrics for the gauge page. Without the database, the node where you make the change is unable to put the change into the database and then propagate that to all other nodes. The result is an inconsistent configuration in the cluster. In extreme cases this could affect the performance of the cluster.

Where ever possible, use the Web GUI itself to change the configuration. The Web GUI always checks that the database is available before allowing you to save any changes to the configuration. This avoids many of the potential data inconsistencies that could arise were you to edit the files directly.

### The list of files to maintain in the database

A cluster includes a file named *stores.lst* that holds a list of directories whose content is to be stored in the configuration database. As supplied, the file specifies

all the directories that contain information which must be synchronized across a cluster. You can further directories to this list for other files that you want synchronized on all nodes.

Periodically check the `stores.lst` to ensure it is fully up to date. The file itself is one of those synchronized across all nodes. So any changes you make to it are automatically propagated to the other nodes.

Take care when editing `stores.lst` to change only those entries you have added. Do not remove any of the supplied entries, as this can adversely affect the operation of the cluster.

## Custom web content

Place any custom web content, such as HTML file, in subdirectories of `tip_home_dir/profiles/TIPProfile/installedApps/TIPCell/isc.ear/OMNIBusWebGUI.war`. In addition, add these directories to the list of files to maintain in the database.

### Related concepts

“The Web GUI in a load balancing environment” on page 93

“Overview of timed tasks” on page 8

### Related tasks

“Administering timed tasks” on page 7

“Maintaining the list of files to monitor” on page 97

---

## Troubleshooting

Use the troubleshooting notes to help correct problems with a load balancing cluster.

## Resynchronizing a node with the cluster

In rare circumstances a fault may cause the configuration data of a node to become corrupted. Use this procedure to bring the data of a node back in synchronization with the cluster.

To resynchronize a node with the cluster restart the node.

### Related tasks

“Restarting the server” on page 1

## Recovering from a database corruption

An event such as a power failure may corrupt the content of the load-balancing database. If at least one node in the cluster still has valid configuration data, use this procedure to recover the database and the remaining nodes.

To recover the database and any nodes with corrupt configuration data, carry out the following on a node with valid configuration data:

1. Stop all nodes on the cluster.
2. Using a suitable SQL tool, access the load-balancing database, and issue the following commands:

```
DELETE FROM OMNIBUS_WEB_GUI.NODES;
DELETE FROM OMNIBUS_WEB_GUI.NODES_CONFIG_ITEMS;
DELETE FROM OMNIBUS_WEB_GUI.CONFIG_ITEMS;
```

3. Start a node that has valid configuration data. The node populates the database with its configuration data.
4. Start each of the other nodes in the cluster. Each node reads the configuration data from the database.

**Related tasks**

“Restarting the server” on page 1



---

## Chapter 3. Setting portlet preferences

You can change the settings of the portlets to customize their appearance and setup to your requirements.

### Related concepts




“The Web GUI in a load balancing environment” on page 93

---

### Setting AEL portlet preferences

To customize the appearance and setup of the AEL portlet, edit the preferences of the portlet.

To set AEL portlet preferences:

1. Open an AEL portlet.
2. Edit your portlet preferences, or, as an administrator, edit the portlet defaults:
  - Click **Personalize** .
  - To edit the portlet defaults of all users, click **Edit Options**  > **Edit shared settings**, or to edit your own preferences, click **Edit Options**  > **Personalize**.
3. In the **General Settings** and the **Appearance of the AEL** area, set the following portlet properties:

**Filter** Select a predefined filter from this list.

#### Filter SQL

Type the SQL syntax that is used to create a transient filter. When the filter string is applied to an AEL, only the rows that meet the criteria set by the filter are displayed. If any value is typed in this field, the **Filter** list is disabled.

**View** Select the view that is applied to the AEL. When a view is applied to an AEL, only the columns that are specified by the view are displayed. The view selected from the list overrides the default view that is assigned to a filter selected from the **Filter** list.

#### Transient filter name

Type the name to assign to the transient filter generated by the SQL expression in the **Filter SQL** field. This name is displayed as the title for the associated AEL. If any value is typed in this field, the **Filter** list is disabled.

#### Event list single-click action

Select the action to perform when you click an event in the AEL once. You can select default actions, such as opening the information window for the selected event, or you can select tools to be run on event data. You can create tools in the Tool Creation editor.

#### Event list double-click action

Select the action to perform when you double-click an event in the AEL. You can select default actions, such as opening the information window for the selected event, or you can select tools to be run on event data. You can create tools in the Tool Creation editor.

**Data Sources**

Select one or more data sources from which the event data is retrieved.

**Portlet Title**

Type the label to be displayed at the top of the AEL portlet.

**Use Customizer**

Select this check box to use drag and drop to control the height of the AEL applet on a page that contains multiple portlets. If you clear this check box, use the **Height** field to specify an alternative height, in pixels.

**Height**

Set the height of the portlet frame in pixels.

**Title Bar**

Select this check box to display the title bar.

**Menu Bar**

Select this check box to display the menu bar.

**Tool Bar**

Select this check box to display the tool bar.

**Filters and Views**

Select this check box to display the **Edit Filters** button and the list of available filters, and the **Edit Views** button and the list of available views on the tool bar.

**Summary Bar**

Select this check box to display the summary bar.

**Status Bar**

Select this check box to display the status bar.

4. In the **Appearance of the AEL** area, select the check box for the bars you want to be displayed in the AEL portlet, and clear the check box for the bars you do not want to be displayed.
5. In the **Bidi Settings** area, specify the settings for the display of bi-directional text:

**Component direction**

Select the arrangement of items in the portlet, left-to-right or right-to-left. The default setting uses the value defined for the page or the console. If the page and console both use the default setting, the locale of your browser determines the layout.

**Text direction**

Select the direction of text on the portlet. The default settings uses the value defined for the page or the console. If the page and console both use the default setting, the locale of your browser determines the text direction. The Contextual Input setting displays text you enter in the appropriate direction for your globalization settings.

Any change you make to these settings is effective only if you have enabled bi-directional support in your user preferences. Changes come into effect the next time you use the portlet.

6. Click **OK** to save and apply your settings to the portlet.

**Related tasks**

“Editing portlet shared settings” on page 42






---

## Setting Chart portlet preferences

To customize the appearance and setup of the Chart portlet, edit the preferences of the portlet.

To set Chart portlet preferences:

1. Open a Chart portlet.
2. Edit your portlet preferences, or, as an administrator, edit the portlet defaults:
  - Click **Personalize** .
  - To edit the portlet defaults of all users, click **Edit Options**  > **Edit shared settings**, or to edit your own preferences, click **Edit Options**  > **Personalize**.
3. In the **General Settings** area, set the following portlet properties:

### Chart Template

Select the chart template you want to use to generate the chart image. The list contains all available Web GUI chart templates.

### Description

Type the description that is used as the alternative text in the HTML image tag.

**Width** Set the width of the rendered chart image in pixels.

### Height

Set the height of the rendered chart image in pixels.

4. In the **Bidi Settings** area, specify the settings for the display of bi-directional text:

### Component direction

Select the arrangement of items in the portlet, left-to-right or right-to-left. The default setting uses the value defined for the page or the console. If the page and console both use the default setting, the locale of your browser determines the layout.

### Text direction

Select the direction of text on the portlet. The default settings uses the value defined for the page or the console. If the page and console both use the default setting, the locale of your browser determines the text direction. The Contextual Input setting displays text you enter in the appropriate direction for your globalization settings.

Any change you make to these settings is effective only if you have enabled bi-directional support in your user preferences. Changes come into effect the next time you use the portlet.

5. Click **OK** to save and apply your settings to the portlet.

### Related tasks

“Editing portlet shared settings” on page 42

---

## Setting Event Dashboard portlet preferences and defaults




To customize the appearance and setup of the Event Dashboard portlet, and the actions that can be executed from the monitor boxes, edit the preferences of the portlet.




### Before you begin


To edit the portlet preferences of the Event Dashboard, either the `ncw_dashboard_editor` role must be assigned to your user, or the `ncw_user` role and the `ncw_admin` roles must be assigned to your user.

In addition to setting portlet preferences, you can set the default preferences for all users (including your own).

To set portlet preferences and defaults for the Event Dashboard:

1. Open an Event Dashboard portlet.
2. Edit your portlet preferences, or, as an administrator, edit the portlet defaults:
  - Click **Personalize** .
  - To edit the portlet defaults of all users, click **Edit Options**  > **Edit shared settings**, or to edit your own preferences, click **Edit Options**  > **Personalize**.
3. To restore the default portlet settings, click **Reset to Defaults**.
4. To change the title of the portlet, type a new name in the **Title** field.
5. From the **Data Sources** list, select the data sources against which to run the filters associated with the monitor boxes.

You must make sure that the data sources specified in the filter and the data sources selected in the Event Dashboard contain identical fields; if this is not the case, an error message is displayed in the affected monitor boxes instead of event data.
6. To add or remove *existing* filters, and therefore monitor boxes, proceed as follows:
  - To remove all monitor boxes from the portlet, click **Remove All Monitor Boxes** .
  - To show all global filters and any user filters that you have defined on the Event Dashboard, click **Show All Filters**.
  - To remove a single monitor box, click **Remove Monitor Box**  next to the required monitor box.
  - To restore a previously-removed monitor box, or add a new monitor box, click **Add Monitor Box**  and select the monitor boxes you require from the Add Monitor Boxes window. You can select from both global filters and any user filters that you have defined.

You can also add system filters.
7. To specify the number of columns in which the monitor boxes are arranged, type a number in the **Columns** field.
8. To add a *new* filter, and therefore a new monitor box to the Event Dashboard, click **Edit Filters** .

The Filter Builder opens, in which you can specify the filter data and SQL query.

9. To specify the type of information displayed in the monitor boxes, and the format of that information:

- a. Click **Edit Preferences** .

- b. In the Preferences window, on the **Monitor Boxes** tab, complete the following fields:

**Show Number of Alerts**

Displays the number of alerts that match the filter.

**Show Highest Severity**

Displays the highest severity of the alerts that match the filter.

**Show Lowest Severity**

Displays the lowest severity of the alerts that match the filter.

**Show Highest Severity Border**

Displays a border around the monitor box in the color of the highest-severity alert that matches the filter.

**Show Metric**

Displays the selected filter metric value.

**Show Highest Color**

Applicable only if you selected the **Show Highest Severity** option:  
Displays the highest-severity alert indicator in the color of the alert, for example, in red if the highest-severity alert is critical.

**Show Lowest Color**

Applicable only if you selected the **Show Lowest Severity** option:  
Displays the lowest-severity alert indicator in the color of the alert.

**Font** Select the font and the font size for the text on the monitor boxes.

**Distribution meter**

Specify the format for the distribution meter:

- **Show Lava Lamp:** Displays the distribution meter as a series of horizontal bars.
- **Show Histogram:** Displays the distribution meter as a bar graph.
- **Show None:** Switches off the distribution meter.

- c. Optional: To specify preferences for Active Event Lists (AELs), complete the information on the other tabs.

- d. Click **Close**.

10. In the **Dashboard Layout** display area, drag the monitor boxes into the required layout.

11. To edit the filter associated with a monitor box, click **Edit Filters**  next to the filter name of a monitor box.

The Filter Builder opens, and the data and SQL query associated with the filter are loaded.

12. In the **Single Click** field, select the required action from the list in response to a single click on the distribution indicator of a monitor box:
  - **Update AEL on Same Page:** Sets the filter and view for any AEL applet that is displayed on the same page as an Event Dashboard.

- **Open AEL in New Window:** Opens a new AEL applet with the filter and default view associated with the clicked monitor box. If you select this option, you can specify what actions are executed when you click or double-click a row in the AEL.
  - **Script:** Executes a custom JavaScript when you click the monitor box.
13. Optional: If you selected the Show New AEL Window option, specify options for the behavior of the AEL:

**Event list single-click action**

Select the action to perform when you click an event in the AEL once. You can select default actions, such as opening the information window for the selected event, or you can select tools to be run on event data. You can create tools in the Tool Creation editor.

**Event list double-click action**

Select the action to perform when you double-click an event in the AEL. You can select default actions, such as opening the information window for the selected event, or you can select tools to be run on event data. You can create tools in the Tool Creation editor.

14. Optional: If you select the Script option, type the script, using JavaScript, in the **Script** field. You can use the following tokens in the script:

**\$(FILTER)**

The name of the filter associated with the monitor box that is clicked.

**\$(DATASOURCE)**

The data source of the monitor box that is clicked.

**\$(PORTLETNAMESPACE)**

The portlet namespace of the Event Dashboard portlet.

For sample scripts, see the *IBM Tivoli Netcool/OMNIBus Installation and Deployment Guide*.

15. Optional: If you selected the Update AEL on Same Page option or the Show New AEL Window option in step 12 on page 229, under **AEL Appearance**, specify the areas of the AEL that you want to be displayed when the AEL is opened after you click the distribution indicator of a monitor box.
16. In the **Bidi Settings** area, specify the settings for the display of bi-directional text:

**Component direction**

Select the arrangement of items in the portlet, left-to-right or right-to-left. The default setting uses the value defined for the page or the console. If the page and console both use the default setting, the locale of your browser determines the layout.

**Text direction**

Select the direction of text on the portlet. The default settings uses the value defined for the page or the console. If the page and console both use the default setting, the locale of your browser determines the text direction. The Contextual Input setting displays text you enter in the appropriate direction for your globalization settings.

Any change you make to these settings is effective only if you have enabled bi-directional support in your user preferences. Changes come into effect the next time you use the portlet.

17. To save and apply your settings to the portlet, click **OK**.

### Related concepts

“Filter Builder overview” on page 286

“Event management tools overview” on page 251

“Event Dashboard overview” on page 305

### Related tasks

“Creating and editing filters in basic mode” on page 287

“Creating and editing filters in advanced mode” on page 290

Chapter 4, “Customizing Active Event Lists,” on page 241

“Customizing the monitor boxes on Event Dashboards” on page 309

---

## Setting gauge portlet preferences




Change the properties of the individual gauges displayed on a Gauges page; add, remove and rearrange gauges; and customize the page itself.

### Before you begin

To be able to customize gauges, your user must have the `ncw_gauges_viewer` and the `ncw_gauges_editor` roles.

You must have the `ncw_gauges_viewer` role.

To customize gauges:

1. Open a Gauge portlet.
2. Edit your portlet preferences, or, as an administrator, edit the portlet defaults:
  - Click **Personalize** .
  - To edit the portlet defaults of all users, click **Edit Options**  > **Edit shared settings**, or to edit your own preferences, click **Edit Options**  > **Personalize**.
3. Change the content and appearance of the page as required:
  - “Changing the General settings for all gauges” on page 232.
  - “Selecting the data sources and ObjectServers” on page 232
  - “Generating HTML for mobile devices” on page 232.
  - “Changing the properties of gauges” on page 233.
  - “Adding gauges” on page 234.
  - “Removing gauges” on page 234.
  - “Rearranging gauges” on page 234.
  - “Changing the bi-directional text settings” on page 235.
4. Click **OK** to save the changes to the gauge properties. The page returns to view mode with the changes applied.

Click **Cancel** to abandon **all** of the changes you have made to the page.

**Note:** To return the page to its initial settings, as supplied, click **Reset to Defaults**.

### Related concepts

“Gauges and metrics” on page 361

“The Web GUI in a load balancing environment” on page 93

### Related tasks

“Setting the thresholds for a gauge” on page 368

“Publishing Gauges pages to mobile devices” on page 365

## Changing the General settings for all gauges

Set the title of the gauge page and the refresh rate for its gauges.

Change any combination of the preferences in the **General Settings** area:

**Title** Type a title for the Gauges page. The title is displayed on the title bar of the portlet page, not on the tab or in the navigation pane.

### Refresh rate

Type the time (in seconds) between each automatic refresh of the values on the gauges. Use a number between 10 and 99000. The default value is 10.

## Selecting the data sources and ObjectServers

Define the data sources and ObjectServers that provide data for the gauges on the page.

Select the data sources and ObjectServers that supply data for the gauges in the **Data Sources** area:

- To select an entire data source, set its check box.
- To select individual ObjectServers in a data source:
  - Clear the check box for the data source.
  - Set the check boxes for the ObjectServers you want to use.
- You can choose any combination of data sources and ObjectServers.
- Initially a page receives data from the default data source only.
- The value that each gauge shows is the total for that metric for all the selected data sources and ObjectServers.

## Generating HTML for mobile devices

Producing an HTML representation of the gauge page that you can send to mobile devices using e-mail or SMS.

1. Set the **HTML for mobile devices** check box.

The system generates an HTML representation of the Gauges page whenever you click **OK** to save changes to the preferences or defaults.

The URL for the HTML page appears in **URL for mobile devices**.

2. To preview the HTML page, click the URL.

You can now copy the URL into an e-mail or SMS for sending to mobile devices.

## Changing the properties of gauges

Modifying the properties of any gauge such as its name, appearance, and metric.

1. Click the gauge that you want to modify.
2. Change the fields in the properties box as required:

**Type** Select type of display for the gauge. For example, a speedometer or traffic lights.

The Preview area shows the selected gauge.

**Metric** Select a predefined metric from the list.

**Note:** If you change the metric for an existing gauge, always change the **Unit label**, **Unit label**, and **Description** to match the new metric.

**Label** Type a label for the gauge. This identifies the gauge on the page.

### Unit label

Type a label to describe the units for the values that the gauge displays.

### Description

Type a description for the gauge. When the Gauges page is viewed in the Web GUI, or the published URL is viewed in a Web browser or a mobile device, the text in this field is used as hover help.

**Tip:** you can include the current value of the gauge in the hover help. Add the string {0} where you want the value to appear. For example:  
The number of client connections. Current value: {0}

### Click action

Select the action that is performed when a user clicks the gauge:

**script** Execute a script.

**url** Open a URL.

### Script/URL

Type the fully-qualified URL that you want to be opened when you click a gauge, or the JavaScript code that you want to run.

**Tip:** You can use the URL of a Web GUI application page. For example, you can specify a Lightweight Event List (LEL) or a map page.

You can use a script to launch Web GUI applications and applications from other parts of Tivoli that are based on Tivoli Integrated Portal. For example, you can write a script to launch the IBM Tivoli Network Manager IP Edition.

3. Click **Apply Changes**.

## Adding gauges

Adding a new gauge to the page and choosing its properties such as name and appearance.

### Before you begin

Decide on the properties of the gauge:

- The type of the gauge.  
Example: thermometer
- The name of the metric the gauge displays.  
Example: connections
- The label for the gauge that appears on the Gauges page.  
Example: Connections
- The units that the gauge displays.  
Example: clients
- A more detailed description of the gauge.  
Example: The number of current client connections.
- The action that the portlet takes, if any, when the user clicks on the gauge.  
For URL actions, determine the fully-qualified URL that the Web GUI is to display.  
For script actions, obtain or write the corresponding JavaScript.

To add the gauge to the page:

1. Set the properties of the gauge from the information you gathered. See “Changing the properties of gauges” on page 233.
2. Click **Add Gauge**.

## Removing gauges

Removing one or more gauges that you no longer require on the page.

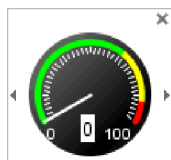
To remove a gauge from the page, click ✕.

## Rearranging gauges

Rearrange the position of the gauges on the page to suit your needs.

Do either of the following actions:

1. Use the arrows on either side of each gauge to change the order of them on the page.



- Use the left arrow ◀ to move the gauge up the list (to the left and up).
  - Use the right arrow ▶ to move the gauge down the list (to the right and down)
2. Position the mouse pointer over a gauge and hold down the left mouse button. Then drag the gauge to its new position and release the mouse button.



## Changing the bi-directional text settings

Customize the settings for displaying bi-directional text.

In the **Bidi Settings** area, specify the settings for the display of bi-directional text:

### Component direction

Select the arrangement of items in the portlet, left-to-right or right-to-left. The default setting uses the value defined for the page or the console. If the page and console both use the default setting, the locale of your browser determines the layout.

### Text direction

Select the direction of text on the portlet. The default settings uses the value defined for the page or the console. If the page and console both use the default setting, the locale of your browser determines the text direction. The Contextual Input setting displays text you enter in the appropriate direction for your globalization settings.

Any change you make to these settings is effective only if you have enabled bi-directional support in your user preferences. Changes come into effect the next time you use the portlet.




---

## Setting Inline Frame portlet preferences

To customize the appearance and setup of the Inline Frame portlet, edit the preferences of the portlet.

**Note:** The Inline Frame portlet is deprecated in the Web GUI from V7.3.1. Use the Web widget portlet instead.

To set Inline Frame portlet preferences:

1. Open an Inline Frame portlet.
2. Edit your portlet preferences, or, as an administrator, edit the portlet defaults:
  - Click **Personalize** .
  - To edit the portlet defaults of all users, click **Edit Options**  > **Edit shared settings**, or to edit your own preferences, click **Edit Options**  > **Personalize**.
3. In the **General Settings** area, set the following portlet properties:

**URL** Type the URL of the content to be included in the portal page.

**Title** Type the title to be displayed for the portlet.

**Height**  
Set the height of the portlet frame in pixels.

**iFrame Name**  
Type a custom name to be used to identify the frame when the content of the frame is loaded on demand, for example using JavaScript.
4. In the **Bidi Settings** area, specify the settings for the display of bi-directional text:

### Component direction

Select the arrangement of items in the portlet, left-to-right or right-to-left. The default setting uses the value defined for the page or

the console. If the page and console both use the default setting, the locale of your browser determines the layout.

#### Text direction

Select the direction of text on the portlet. The default settings uses the value defined for the page or the console. If the page and console both use the default setting, the locale of your browser determines the text direction. The Contextual Input setting displays text you enter in the appropriate direction for your globalization settings.

Any change you make to these settings is effective only if you have enabled bi-directional support in your user preferences. Changes come into effect the next time you use the portlet.

5. Click **OK** to save and apply your settings to the portlet.

#### Related tasks




“Editing portlet shared settings” on page 42

---

## Setting LEL portlet preferences

To customize the appearance and setup of the LEL portlet, edit the preferences of the portlet.

To set LEL portlet preferences:

1. Open an LEL portlet.
2. Edit your portlet preferences, or, as an administrator, edit the portlet defaults:
  - Click **Personalize** .
  - To edit the portlet defaults of all users, click **Edit Options**  > **Edit shared settings**, or to edit your own preferences, click **Edit Options**  > **Personalize**.
3. In the **General Settings** area, set the following portlet properties:

**Filter** Select the filter you want to apply to the event list. The default view associated with the filter is automatically applied.

**Use Customizer**  
Automatically determines the required height of the portlet frame. You can override the height by clearing this check box and typing a value in the **Height** field.

**Height**  
Set the height of the portlet frame in pixels.
4. In the **Bidi Settings** area, specify the settings for the display of bi-directional text:

#### Component direction

Select the arrangement of items in the portlet, left-to-right or right-to-left. The default setting uses the value defined for the page or the console. If the page and console both use the default setting, the locale of your browser determines the layout.

#### Text direction

Select the direction of text on the portlet. The default settings uses the value defined for the page or the console. If the page and console both use the default setting, the locale of your browser determines the text direction. The Contextual Input setting displays text you enter in the appropriate direction for your globalization settings.

Any change you make to these settings is effective only if you have enabled bi-directional support in your user preferences. Changes come into effect the next time you use the portlet.

5. Click **OK** to save and apply your settings to the portlet.

#### Related tasks




“Editing portlet shared settings” on page 42

---

## Setting preferences for the Map portlet

To customize the appearance and setup of the Map portlet, edit the preferences of the portlet.

To set Map portlet preferences:

1. Open a Map portlet.
2. Edit your portlet preferences, or, as an administrator, edit the portlet defaults:
  - Click **Personalize** .
  - To edit the portlet defaults of all users, click **Edit Options**  > **Edit shared settings**, or to edit your own preferences, click **Edit Options**  > **Personalize**.
3. In the **General Settings** area, set the following portlet properties:

#### Map Name

From the list of available maps, select the map that you want to display as an applet in the portlet.

#### Sound URL

Type the URL to the sound file that specifies the sound to play when the status of the map is updated during a refresh. The URL must be specified in the following format:

*protocol://host:port/path/filename.fileextension*

#### Refresh Rate

Type a time value in seconds that specifies the interval between refresh operations for the map.

#### Enable hover help for active objects

Select this check box to display hover help for active map objects that are associated with a filter. The hover help displays information from the filter.

#### Show status bar

Select this check box to display the status bar, which displays a countdown of the time, in seconds until the map is next refreshed.

#### Use Customizer

Uses the height setting specified in the map on the map page. If you clear this check box, you can explicitly specify the height in the **Height** field.

#### Height

If you clear the **Use Customizer** field, type an alternative height value in pixels.

4. In the **Bidi Settings** area, specify the settings for the display of bi-directional text:

### Component direction

Select the arrangement of items in the portlet, left-to-right or right-to-left. The default setting uses the value defined for the page or the console. If the page and console both use the default setting, the locale of your browser determines the layout.

### Text direction

Select the direction of text on the portlet. The default settings uses the value defined for the page or the console. If the page and console both use the default setting, the locale of your browser determines the text direction. The Contextual Input setting displays text you enter in the appropriate direction for your globalization settings.

Any change you make to these settings is effective only if you have enabled bi-directional support in your user preferences. Changes come into effect the next time you use the portlet.

5. Click **OK** to save and apply your settings to the portlet.

### Related tasks




"Editing portlet shared settings" on page 42

---

## Setting Table View portlet preferences

To customize the appearance and setup of the Table View portlet, edit the preferences of the portlet.

To set Table View portlet preferences:

1. Open a Table View portlet.
2. Edit your portlet preferences, or, as an administrator, edit the portlet defaults:
  - Click **Personalize** .
  - To edit the portlet defaults of all users, click **Edit Options**  > **Edit shared settings**, or to edit your own preferences, click **Edit Options**  > **Personalize**.
3. In the **General Settings** area, set the following portlet properties:

**Filter** Select the filter you want to apply to the event list. The default view associated with the filter is automatically applied.

### Maximum rows

Set the number of event rows you want to display. If the number you enter is smaller than the actual event count, you are notified at the bottom of the table. To display all events, set this value to -1.

### Use Customizer

Automatically determines the required height of the portlet frame. You can override the height by clearing this check box and typing a value in the **Height** field.

### Height

Set the height of the portlet frame in pixels.

4. In the **Bidi Settings** area, specify the settings for the display of bi-directional text:

### Component direction

Select the arrangement of items in the portlet, left-to-right or right-to-left. The default setting uses the value defined for the page or

the console. If the page and console both use the default setting, the locale of your browser determines the layout.

#### Text direction

Select the direction of text on the portlet. The default settings uses the value defined for the page or the console. If the page and console both use the default setting, the locale of your browser determines the text direction. The Contextual Input setting displays text you enter in the appropriate direction for your globalization settings.

Any change you make to these settings is effective only if you have enabled bi-directional support in your user preferences. Changes come into effect the next time you use the portlet.

5. Click **OK** to save and apply your settings to the portlet.

#### Related tasks




“Editing portlet shared settings” on page 42

---

## Setting Web widget portlet preferences

To customize the appearance and set up of the Web widget portlet, edit the preferences of the portlet.

To set Web widget portlet preferences:

1. Open a Web widget portlet.
2. Edit your portlet preferences, or, as an administrator, edit the portlet defaults:
  - Click **Personalize** .
  - To edit the portlet defaults of all users, click **Edit Options**  > **Edit shared settings**, or to edit your own preferences, click **Edit Options**  > **Personalize**.
3. Set the properties of the Web widget:

#### Widget title

Type a brief descriptive name for the portlet. The title appears in the navigation pane of the Tivoli Integrated Portal console.

#### Home URL

Type the Web address of the page to display in the portlet. For Web GUI items such as maps, use a URL relative to the context root. For example, for a map named MyMap use `webtop/Map/MyMap`. You can also display Web pages. In this case specify the fully qualified name of the page. For example, `http://www.mycompany.com/welcome.html`.

#### Help page

Type the URL of a custom HTML help topic to replace the default help topic for the Web widget.

#### HTML iFrame name

Type a unique iFrame name for this Web widget. This name uniquely identifies the Web widget and allows its content to be updated dynamically. Ensure that each Web widget has a unique value for this property.

#### Show a browser control toolbar

set this check box to provide users of the portlet with a Web navigation toolbar. That is a standard set of Web navigation buttons and a Web address entry field.

Users without administrative privileges may see only a subset of these properties, depending on the how the administrator configured the portlet.

4. To allow users to personalize their Web widget settings, set the relevant check boxes:

- **Widget title**
- **Home page**
- **Help page**
- **Browser control bar**

Initially, all check boxes are clear.

5. Click **Save** to save and apply your settings to the portlet.

---

## Chapter 4. Customizing Active Event Lists

You can customize the appearance and behavior of the Active Event List (AEL). You can also create tools for managing events and customize what functions are in the AEL menus.

---

### Setting the appearance and behavior of the Active Event List

You can configure the appearance and behavior of the AEL window and any monitor boxes associated with it. For example, you can specify the manner in which you are notified of changes in alert status, list refresh time, window layout, and so on.

#### Before you begin

An administrator must set the Allow preference configuration permission in your user profile in so that you can edit your user preferences.

#### Related concepts

“The Web GUI in a load balancing environment” on page 93

#### Related tasks

“Modifying the preferences of a Web GUI user” on page 70

### Changing the AEL refresh rate

You can changing the a time period in seconds after which the AEL is automatically refreshed on a regular basis by the Web GUI server.

#### Before you begin

A user can view the **Refresh** tab and change the refresh rate only if the permission for Allow refresh rate configuration has been selected in their user profile.

Setting a timed refresh forces the event list to reload data pertaining to changed events. Do not set the refresh to a low value (for example, less than 60 seconds) because this has an impact on ObjectServer performance and network traffic.

To change the refresh rate:

1. Click **Settings > User Preferences for Tivoli Netcool/OMNIBus Web GUI**.
2. From the **Available users** list, select the required user and click **Modify**.
3. Select **Allow refresh rate configuration** and type a time in the **Refresh rate (seconds):** field.
4. Click **Save**.

#### Related tasks

“Refreshing the event data” on page 377

## Turning data row caching on or off

When the AEL is refreshed, event data in its rows is loaded from the cache if the refresh interval is less than 60 seconds. To have the event list refreshed from the database, turn data row caching off.

To turn off data row caching:

1. Open the following file:  
ncwDataSourceDefinitions.xml
2. Set the **enabled** attribute of the <eventList> element to false. For example:  
`<eventList maxAge="60" enabled="false" cleantime="120" />`
3. Save and close the file.
4. Restart the server.

### Results

When the AEL is refreshed, data is drawn directly from the ObjectServer instead of the data cache.

#### Related tasks

“Restarting the server” on page 1

## Changing event severity icons

To make the icons that denote event severity more recognizable to users, replace the default icons with images of your choice.

The replacement images must be in PNG format. The file names must correspond to the integers that denote event severity in the ObjectServer. The following table describes the default file names for the images.

Table 60. Default file names for event severity icons

| File name | Corresponding event severity |
|-----------|------------------------------|
| 5.png     | Critical                     |
| 4.png     | Major                        |
| 3.png     | Minor                        |
| 2.png     | Warning                      |
| 1.png     | Indeterminate                |
| 0.png     | Clear                        |

To change the icons for event severity:

1. Rename the files for the replacement icons so that they correspond to the required event severity.
2. Change to the following directory:  
`tip_home_dir/profile/TIPProfile/installedApps/TIPCell/isc.ear/OMNIBusWebGUI.war/graphicconversions/severity`
3. Overwrite the existing PNG files with the replacements.

### Results

After users have logged out, and logged back into the Web console, the icons in the Active Event List are changed.



## Restricting the number of rows displayed in the Active Event List

You can impose a limit on the number of rows returned to the Active Event List (AEL). When operators open the AEL, they see only the specified number of rows.

To restrict the number of rows, edit the **ael.top-n.value** property in the *webgui\_home\_dir/etc/server.init* file. If you set this property to a value greater than 0, the AEL queries are modified to include a TOP keyword that restricts the number of rows returned to the AEL to the first *n* query results that match the selection criteria specified by the filter.

For example, if an AEL filter matches 8000 rows in the ObjectServer, and the **ael.top-n.value** value is set to 4000, only the top 4000 alerts are displayed. If an AEL displays events from multiple data sources, the top *n* rows per data source are displayed. For example, if the **ael.top-n.value** value is set to 50, and the AEL is configured to display events from three data sources, a maximum of 150 rows are displayed.

After you have edited the *server.init* file, restart the Tivoli Integrated Portal server.

To restrict the number of events displayed:

1. In the command-line interface, open the *webgui\_home\_dir/etc/server.init* file.
2. Uncomment the section Active Event List properties.
3. In this section, set the value of the **ael.top-n.value** property to a value greater than 0.
4. Leave the values of the other properties in the Active Event List properties section at their default values.

For more information about these properties, see the information in the *server.init* file.

5. Save and close the file.
6. Restart the server.

### Results

When operators view an AEL, the number of rows that match the filter criteria is restricted to the value of the **ael.top-n.value** property. If the AEL displays events from multiple data sources, the number is restricted to the value of **ael.top-n.value** per data source. A Top Set to message is also displayed above the distribution status bar in the AEL indicating that a TOP condition is being applied.

#### Related tasks

“Restarting the server” on page 1

#### Related reference

“Performance tuning” on page 212

## Enabling notifications and setting notification criteria

You can set the AEL to inform you about changes in alert status. You can also specify the conditions which must be met before you receive a notification.

1. Press Shift+P to open the Preferences window.
2. Click **Notifications** and complete the fields as follows:

### Enabled

Select this check box to receive notification of new, changed, or deleted alerts when the event list is minimized.

### When Iconized

Select this check box to receive notification of new, changed, or deleted alerts on iconized desktop environments. An iconized desktop environment displays an icon when the event list is minimized.

Use the **When** and **How** options to set the notification method.

### Always

Select this check box to always receive notification of new, changed, or deleted alerts.

**When** Select each check box to receive notification as follows:

**New** You receive a notification when a new alert is added to the event list.

### Change

You receive a notification when an existing alert changes in the event list.

**Delete** You receive a notification when an existing alert is deleted from the event list.

**How** Select each option to indicate how a notification should occur:

### Alert Icon

Flashes the minimized event list.

### Open Window

Opens the event list on the screen.

### Play Sound

Plays a sound on the workstation. In the text box, specify the sound file to play in the following format:

`$(SERVER)/sounds/soundfile`

Replace *soundfile* with the name of the sound file. For example: `gong.au`

### Open URL

Opens a URL

### URL Target

If you selected **Open URL**, type the URL that you want to be opened.

3. Save the settings for use in the current session, or for future sessions:
  - To use these preferences in the current session only, click **Apply**.
  - To use these preferences in future sessions, click **Save**.
4. Click one of the other tabs to make more changes or, to exit the Preferences window, click **Close**.

## Enabling flashing and setting flash speed and brightness

You can specify flashing in the AEL for alerts where the Flash field in the alerts.status table has been set to 1, or converted to Yes. Also, you can adjust the interval between each flash and the brightness of the flash in the window.

To enable flashing and adjust the speed and brightness of the flashing:

1. Press Shift+P to open the Preferences window.
2. Click **Flashing** and complete the fields as follows:

### Enable Flashing

Select this check box to enable event list flashing.

Use the **Speed** slider to indicate how quickly the event list flashes.

Use the **Brightness** slider to indicate the degree of brightness of the flashing.

3. Save the settings for use in the current session, or for future sessions:
  - To use these preferences in the current session only, click **Apply**.
  - To use these preferences in future sessions, click **Save**.
4. Click one of the other tabs to make more changes or, to exit the Preferences window, click **Close**.

## Modifying the AEL font and window settings

You can set the AEL font type, and specify the color and toolbar preferences for AELs.

To set the preferences for the font type, color and toolbar:

1. Press Shift+P to open the Preferences window.
2. Click **Event List** and complete the fields under **Event Window** it as follows:

### Show Colors

Displays each row of the event list with a background color that corresponds to the severity of the event.

### Show Distribution Summary Bar

Select this check box to display the distribution summary bar in the event list. The distribution summary bar displays the number of alerts that match each severity color.

### Show Toolbar

Select this check box to make the toolbar available on the event list.

### Font Name

Select a font for your event list from the list.

### Font Size

Select a font size for your event list from the list.

3. Save the settings for use in the current session, or for future sessions:
  - To use these preferences in the current session only, click **Apply**.
  - To use these preferences in future sessions, click **Save**.
4. Click one of the other tabs to make more changes or, to exit the Preferences window, click **Close**.

## Modifying the AEL date and time format

You can specify the date and time format used on the AEL.

1. Press Shift+P to open the Preferences window.
2. Click **Event List**.
3. Under **Date Format**, select a format for the date and time:

| Option    | Description                                                                                                                  |
|-----------|------------------------------------------------------------------------------------------------------------------------------|
| Short     | This is the default setting, and is of the format m/d/yy h:mm:ss a, for example: 12/11/00 2:15:55 PM.                        |
| Long      | This is of the format MMMM d, yyyy h:mm:ss a, for example December 11, 2000 2:15:55 PM.                                      |
| Customize | Create your own format based on available date and time formats. See "Permitted date and time formats" for more information. |

4. Specify the operator timezone by selecting an entry from the menu. You can select the location of the operator, the timezone, or the number of hours before or after Greenwich Mean Time.

**Tip:** Choose the name of a locale-based timezone (for example America/Chicago) rather than one relative to GMT (for example, etc/GMT-6).

5. Save the settings for use in the current session, or for future sessions:
  - To use these preferences in the current session only, click **Apply**.
  - To use these preferences in future sessions, click **Save**.
6. Click one of the other tabs to make more changes or, to exit the Preferences window, click **Close**.

## Permitted date and time formats

The following table describes the date and times formats that you can use for a customized date and time display in step 3.

Table 61. Date and time formats

| Symbol | Description              | Presentation     | Example             |
|--------|--------------------------|------------------|---------------------|
| G      | era designator           | Textual          | AD                  |
| y      | year                     | Numeric          | 1996                |
| M      | month in the year        | Text and Numeric | July and 07         |
| d      | day in the month         | Numeric          | 10                  |
| h      | hour in am/pm (1 - 12)   | Numeric          | 12                  |
| H      | hour in the day (0 - 23) | Numeric          | 0                   |
| m      | minute in hour           | Numeric          | 30                  |
| s      | second in minute         | Numeric          | 55                  |
| S      | millisecond              | Numeric          | 978                 |
| E      | day in week              | Textual          | Tuesday             |
| D      | day in year              | Numeric          | 189                 |
| F      | day of week in month     | Numeric          | 2 (2nd Wed in July) |
| w      | week in year             | Numeric          | 27                  |

Table 61. Date and time formats (continued)

| Symbol | Description               | Presentation | Example               |
|--------|---------------------------|--------------|-----------------------|
| W      | week in month             | Numeric      | 2                     |
| a      | am or pm marker           | Textual      | PM                    |
| k      | hour in day (1 - 24)      | Numeric      | 24                    |
| K      | hour in am or pm (0 - 11) | Numeric      | 0                     |
| z      | time zone                 | Textual      | Pacific Standard Time |

## Date and time formats in the AEL

Read about the format of the date and time presentation types, including additional information on specific data formats.

When using the customized date format in the AEL, the display presentation is determined by the total letters or numbers.

### Textual

In a textual presentation with four or more pattern symbols, the full form is used. In a textual presentation with fewer than four symbols, the short or abbreviated form is used (if one exists).

For example, if you want the date to start with the day of the week, you use the symbol E (Day in week). If you enter E fewer than four times, the day is abbreviated. If you enter E four or more times, the day is presented in the full form.

- EEE is displayed as Mon
- EEEE is displayed as Monday

### Numeric

In a numeric presentation, the minimum number of digits is used. Shorter numbers are zero-padded to this amount.

The Y (year) symbol is a special case. If the count of y is two, the value for the year is truncated to two digits. If the count of y is four, the year is shown in four digits. For example:

- yy is displayed as 03
- yyyy is displayed as 2003

### Textual and numeric

In a textual and numeric presentation, where three or more pattern letters are used, the textual form is used. Otherwise, a numeric form is used.

For example, if you use the symbol M (month in year) the presentation of month is determined by how many times you enter M. For example, for the month of February:

- M is displayed as 2
- MM is displayed as 02
- MMM is displayed as Feb

- MMM is displayed as February

### Additional information

Other points to note about letters and numbers in date formats:

- You can use non-alphabetic characters in date formats, such as colons (:), commas (,), periods (.), the number sign (#) and the at sign (@), without having to contain them in quotes.
- If you include a colon to separate hours and minutes, then the colon is displayed. For example, HH:mm a is displayed as 5:48 PM.
- A pattern containing any invalid symbol results in an error during formatting or parsing.

### Locale date and time

Each country has its own locale date and time format. The following table shows examples of date and time in the US Locale format.

Table 62. US locale date format







| Format Pattern              | Result                       |
|-----------------------------|------------------------------|
| EEE, MMM d, 'yy             | Wed, July 10, '96            |
| h:mm a                      | 12:08 PM                     |
| hh a,zzzz                   | 12 PM, Pacific Daylight Time |
| K:mm a, z                   | 0:00 PM, PST                 |
| yyyyy.MMMMM.dd GGG hh:mm aa | 1996.July.10 AD 12:08 PM     |

## Changing how event severity is depicted in the AEL

You can specify how the severity of events is displayed in the AEL: as icons, as text, or as a combination of both icons and text.

Default icons for event severity are provided. These icons are as follows.

If required, you can change these icons.

-  : Denotes critical severity (severity 5)
-  : Denotes major severity (severity 4)
-  : Denotes minor severity (severity 3)
-  : Denotes warning severity (severity 2)
-  : Denotes indeterminate severity (severity 1)
-  : Denotes clear severity (severity 0)

To change the depiction of event severity:

1. Press Shift+P to open the Preferences window.
2. Click **Event List**.
3. Under **Event List Icons**, select one of the following options:
  - **Show**: Displays an icon to denote event severity.
  - **Show With Text**: Displays an icon and text to denote event severity.

- **Don't Show:** Displays text to denote event severity.
- 4. Save the settings for use in the current session, or for future sessions:
  - To use these preferences in the current session only, click **Apply**.
  - To use these preferences in future sessions, click **Save**.
- 5. Click one of the other tabs to make more changes or, to exit the Preferences window, click **Close**.

## Changing which areas of the AEL are displayed




Depending on your preferences, you can set up the AEL so that only the areas that you want to work with are displayed, for example, the menu and title bars. As an administrator, you can control which areas are displayed for all users.

You can control whether the following areas are displayed or hidden:

- Title bar
- Menu bar
- Tool bar
- Filters and view section of the tool bar
- Summary bar
- Status bar

**Tip:** To provide the maximum space for displaying alert data, by default, the title bar and the menu bar are hidden.

To change which areas of the AEL are displayed:

1. Open an AEL.
2. Edit your portlet preferences, or, as an administrator, edit the portlet defaults:
  - Click **Personalize** .
  - To edit the portlet defaults of all users, click **Edit Options**  > **Edit shared settings**, or to edit your own preferences, click **Edit Options**  > **Personalize**.
3. To change the areas of the AEL that are displayed, use the following fields:

### Title Bar

Select this check box to display the title bar.

### Menu Bar

Select this check box to display the menu bar.

### Tool Bar

Select this check box to display the tool bar.

### Filters and Views

Select this check box to display the **Edit Filters** button and the list of available filters, and the **Edit Views** button and the list of available views on the tool bar.

### Summary Bar

Select this check box to display the summary bar.

### Status Bar

Select this check box to display the status bar.

4. Click **OK**.

## Changing the font color for an event after it has been acknowledged

You can set the color of the event font to change to after the event has been acknowledged.

To set the color of the font for an event that has been acknowledged for all new users:

1. Do one of the following actions:
  - To change the font color for all new users, edit the file `webgui_home_dir/etc/system/userdefaults.props`
  - To change the font color for an existing user, edit the file `webgui_home_dir/etc/configstore/ncwUserPreferences/username.nova`. Replace `username` with the user ID of the user.
2. Locate the property **ael\_user\_properties\_acknowledge\_font\_color** and change its value to the name of the required color.
3. Save the file.
4. If you have changed the `userdefaults.props` file, restart the server.  
If you changed the font color for an existing user, the change takes effect next time they log in.

### Related tasks

“Restarting the server” on page 1

## Adding sounds to use for notifications

You can add sound files to the Web GUI to use in AEL notifications.

Prepare the sound file and place it in the following directory::

`tip_home_dir/profiles/TIPProfile/installedApps/TIPCell/isc.ear/OMNIBusWebGUI.war/sounds`

---

## Creating event management tools

You can create and administer CGI, SQL, command line and script tools to be used in the AEL. You can also configure prompts that are displayed to users in the Active Event List (AEL) when performing actions with tools.

### Before you begin

If you want to create tools that will be run against more than one data source, note the following criteria:

- The tool must be valid against the ObjectServer from which the events originate. If you select events from multiple ObjectServers, the tool must be valid against all the ObjectServers. For example, if the tool is configured to run against fields that are not contained in one ObjectServer, the tool cannot be run against the entire selection of events.
- You must have write permission against all the ObjectServers from which the selected events originate. If you select events from multiple ObjectServers, and you do not have write permission in all the ObjectServers, the tool runs against only the ObjectServers that you are permitted to modify.

**Important:** For each tool, you must select at least one data source. From this data source, the Tool Editor obtains the fields against which the tool can be configured.



If you select multiple data sources, the Tool Editor displays only the fields common to all the selected data sources. The data sources selected in the Tool Editor are not used in the AEL, Map Editor, or Event Dashboard to retrieve event data. You select the data source or data sources from which event data is retrieved in the portlet preferences for the AEL, Map Editor, and Event Dashboard.

#### Related concepts

“The Web GUI in a load balancing environment” on page 93

## Event management tools overview

From the Tool Creation page, you can create and configure the tools used by clients connected to the AEL. Tools can be either Common Gateway Interface (CGI) tools, SQL tools, command-line tools or script tools. All of these tools are run from configurable menus in the AEL and some of them can include a prompt window or a pop-up menu for users to enter or select information.

### CGI tools

CGI tools are typically used by clients to process ObjectServer field information and return useful information about the data to the client browser.

For example, the CGI script **nco\_ping** provided with Web GUI sends specially marked packets from the local computer to a remote device to determine whether the remote computer is currently available. The script uses the IP address contained in the Node field of the selected alert (or alerts) as its target. The source file for **nco\_ping** is located in the following directory:

```
webgui_home_dir/etc/cgi-bin
```

### Query string for **nco\_ping**

The following example shows the URL for the **nco\_ping** CGI script run on an event generated from a device with the node name **zen1** and data source **NCOMS**. The GET query string is as follows.

```
selected_rows.Node=zen1&datasource=NCOMS
```

This data is passed to the CGI script.

```
protocol://server:port/ibm/console/webtop/cgi-bin/
nco_ping.cgi?%24selected_rows.Node=zen1&datasource=NCOMS
```

CGI tools do not have to run CGI scripts. CGI tools can also be used to open a local or remote Web page. Be aware that a remote script or page that contains SmartPage commands cannot be presented correctly within Web GUI. Pages that contain SmartPage commands must be hosted on the local Web GUI server.

CGI scripts can use different kinds of variables, including HTTP variables, ObjectServer fields, and browser cookie values.

**Note:** If any required resources, for example, Perl, are installed in nonstandard locations, ensure that the paths to the resources in any CGI scripts are correct.

#### Related tasks

“Setting up CGI and URL tools” on page 255

## SQL tools

SQL tools are a predefined way to run ObjectServer SQL commands on the alerts.status table and the alerts.journal data table from the Active Event List (AEL).

To run an SQL tool, you must be a read-write user authenticated with the ObjectServer against which the tool is run.

The Web GUI provides several default SQL tools. To view the tools, open the **Alerts** menu in the AEL.

In a Dual-Server Desktop environment, SQL tools run simultaneously against the master ObjectServer and the display server from which the AEL data is displayed. The same is true of journal actions.

An example of an SQL tool is **acknowledge**, which contains the following data.

```
update alerts.status set Acknowledged=1 where Serial in ($selected_rows.Serial);
Alert acknowledged by %username
```

These instructions acknowledge any selected alert in the name of the user who ran the tool. The first line applies to the data held in the alerts.status table, the second line to the alerts.journal table.

For more information about ObjectServer SQL syntax, see the *IBM Tivoli Netcool/OMNIBus Administration Guide*.

### Related tasks

“Creating SQL tools” on page 261

## Command-line tools

Command-line tools are predefined command strings that run a command-line action on a client system. When invoked from the Active Event List (AEL), the instruction typically instructs the client system to open a command prompt and pass field data to an application.

**Attention:** IBM cannot guarantee that command-line tools will not adversely affect your system. IBM does not accept responsibility for the consequences of any actions performed through the execution of a command-line tool.

A command-line tool is useful when all members of a particular user group are known to have specific (usually generic) applications on their systems. Because command-line tools are started client-side rather than server-side, the load on the Web GUI server is reduced.

When a command-line tool is started by a client, the client operating system type is automatically determined, and the appropriate command-line instruction is sent, if one is available for that operating system.

The following example shows a command-line tool instruction:

```
start cmd /k %WINDIR%\SYSTEM32\PING.EXE {@Node}
```

In this example, if the client is a Windows operating system the instruction opens a command-line and starts the Windows **ping** utility against the Node field of the selected alert. The data is then returned to the command-line window on the client system.

## Related tasks

“Creating command-line tools” on page 263

## Script tools

Script tools are a predefined way of passing contextual data from selected events in the Active Event List (AEL) to JavaScript methods. A script tool can be run from within the AEL.

Use script tools for inter-portlet activity through the Tivoli Integrated Portal Actions framework, and to customize dynamic content using JavaScript.

Script tool syntax follows JavaScript rules. The command text might also contain variables that are evaluated when the script tool is executed.

You can create, copy, modify, and delete script tools in the Tool Creation editor. Script tools can be added to menus using the Menu Configuration editor.

When an AEL is opened, all script tools that are referred to by the AEL tools are retrieved from the server.

## Access criteria for tools

You can define access criteria for any SQL, CGI, URL, script or command-line tool based on the groups that a user belongs to and the class of an event. If the access criteria are satisfied for a given tool, user, and event, the tool is displayed.

By default, no access criteria are defined for any tools. Tools that do not have access criteria defined are displayed for all users for all events. Changes in access criteria take effect when the Active Event List (AEL) is reloaded, without the need to restart the Web GUI server.

If the access criteria for a tool, user, and event are not satisfied, the tool is not displayed. If both group and class access criteria are defined, then both must be satisfied for the tool to be displayed for a given event and user. If multiple events are selected in the AEL, all access criteria must be satisfied for all selected events in order for a tool to be displayed.

## Prompt types

When you create or edit a tool, you can include a prompt to which a user must respond, for example by typing in information or selecting a value from a list.

Each prompt has a user-configurable label that is displayed above the prompt window. It informs the user of the expected input, such as the name of the server to be pinged. You can refer to prompts in tools using the **\$prompt.promptname** parameter or the **{ \$prompt.promptname }** parameter for CGI and Script tools.

The following example shows a dynamic choice prompt type:

```
<methodCall xmlns:prompt="http://www.ibm.com/tivoli/netcool/webtop/
prompts/2.2">
 <method methodName="command">
 <prompt:prompt type="DynamicChoice" name="prompt name">
 <prompt:parameters label="prompt label" order="prompt order"
errorMessage="error message" localized="true|false">
 <prompt:additionalParams>
 <prompt:param name="sqlCommand" value="sql string"/>
 </prompt:additionalParams>
```

```

 </prompt:parameters>
 </prompt:prompt>
</method>
</methodCall>

```

Prompt parameters can be modified using the WAAPI client. View the *webgui\_home\_dir/waapi/etc/samples/samlrequest\_prompt.xml* file for samples.

You can create the following types of prompts:

**String** This creates a prompt window that accepts one or more characters. If more than one prompt has been defined for a tool, all prompts will be displayed on a single panel. The **order** attribute is responsible for determining in what order prompts are displayed on this panel, from higher values to lower values in a page orientation. To ensure a prompt is always displayed last, set the value of the **order** attribute to 0.

#### Integer

This creates a prompt window that accepts an integer value.

**Float** This creates a prompt window that accepts a floating point number, which can contain a decimal point.

**Time** This creates a prompt window that accepts a time.

#### Fixed Choice

This creates a menu that is populated with options that you specify.

#### Lookup

This creates a menu or list that is populated by the values in a specified file. The **file** attribute contains an absolute path to a file in the server, where each line of text is displayed as an item in a list.

#### Password

This creates a prompt window that accepts one or more characters as a password.

#### Dynamic Choice

This creates a pop-up menu or drop-down list that is populated by the results of an ObjectServer query. The **sqlCommand** attribute contains an ObjectServer SQL SELECT statement for two columns from a table. Each row that is returned by the ObjectServer is displayed on the client as an item in a submenu or list. If a Dynamic Choice tool is run against multiple ObjectServers that have different column definitions, you can select only from the columns or column values that are common to all ObjectServers.

#### Multiline String

This creates a multiline prompt window that accepts one or more characters.

**Tip:** You can use the **order** attribute to enforce a mandatory journal entry as the last prompt to be completed by a user. (This is similar to the forced journal entry functionality in the event list). To do so, create a multiline string prompt, enter a name of Journal entry and an order of 0.

#### Formatted String

This creates a prompt window that accepts one or more characters, provided they are in the predefined format.

The **format** attribute contains a regular expression that the user must match in order for the value to be accepted.

### Real-Time Dynamic Choice

This creates a scrollable list populated by the results of an ObjectServer query in real-time, which means during tool execution. This prompt is meant to be used to display data from an ObjectServer table that is frequently changeable. As this prompt type is executed in real time, it should be used sparingly to reduce the load on the Web GUI server.

The **sqlCommand** attribute contains an ObjectServer SQL SELECT statement for two columns from a table. Each row that is returned by the ObjectServer is displayed on the client as an item in a submenu or list. If a Real-Time Dynamic Choice tool is run against multiple ObjectServers that have different column definitions, you can select only from the columns or column values that are common to all ObjectServers.

### Related tasks

“Acknowledging and deacknowledging events” on page 372

## Setting up CGI and URL tools

A CGI is a server-resident program that provides a standard way of adding dynamic content to a Web site by allowing external gateway programs to interface with information servers such as Web servers.

### Related concepts

“CGI tools” on page 251

### Creating a CGI script

Create the CGI script you want to make available as a tool for your users.

1. Create your CGI script. You can create a CGI script using any programming language.

**Attention:** Ensure the script is executable.

2. Save your CGI script on the Web GUI server in the *webgui\_home\_dir/etc/cgi-bin* directory.

### Environment variables in CGI script:

The following table is a list of all HTTP variables that can be passed from the Web GUI server to CGI scripts.

Table 63. HTTP server variables

| Variable          | Description                                                                                                                                                                                                     |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AUTH_TYPE         | The name of the authentication scheme used to protect the servlet. For example, BASIC, SSL, or null if the servlet was not protected.                                                                           |
| CONTENT_LENGTH    | The length of the request body in bytes made available by the input stream or -1 if the length is not known. For HTTP servlets, the value returned is the same as the value of the CGI variable CONTENT_LENGTH. |
| CONTENT_TYPE      | The MIME type of the body of the request, or null if the type is not known. For HTTP servlets, the value returned is the same as the value of the CGI variable CONTENT_TYPE.                                    |
| GATEWAY_INTERFACE | The revision of the CGI specification being used by the server to communicate with the script. It is "CGI/1.1".                                                                                                 |

Table 63. HTTP server variables (continued)

| Variable                 | Description                                                                                                                                                                                                                                                                                                                                     |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| HTTP_ACCEPT              | Variables with names beginning with "HTTP_" contain values from the request header, if the scheme used is HTTP. HTTP_ACCEPT specifies the content types your browser supports. For example, text/xml.                                                                                                                                           |
| HTTP_ACCEPT_CHARSET      | Character preference information. Used to indicate the client's preferred character set if any. For example, utf-8;q=0.5.                                                                                                                                                                                                                       |
| HTTP_ACCEPT_ENCODING     | Defines the type of encoding that may be carried out on content returned to the client. For example, compress;q=0.5.                                                                                                                                                                                                                            |
| HTTP_ACCEPT_LANGUAGE     | Used to define which languages you would prefer to receive content in. For example, en;q=0.5. If nothing is returned, no language preference is indicated.                                                                                                                                                                                      |
| HTTP_FORWARDED           | If the request was forwarded, shows the address and port through of the proxy server.                                                                                                                                                                                                                                                           |
| HTTP_HOST                | Specifies the Internet host and port number of the resource being requested. Required for all HTTP/1.1 requests.                                                                                                                                                                                                                                |
| HTTP_PROXY_AUTHORIZATION | Used by a client to identify itself (or its user) to a proxy which requires authentication.                                                                                                                                                                                                                                                     |
| HTTP_USER_AGENT          | The type and version of the browser the client is using to send the request. For example, Mozilla/1.5.                                                                                                                                                                                                                                          |
| PATH_INFO                | Optionally contains extra path information from the HTTP request that invoked the script, specifying a path to be interpreted by the CGI script. PATH_INFO identifies the resource or sub-resource to be returned by the CGI script, and it is derived from the portion of the URI path following the script name but preceding any query data. |
| PATH_TRANSLATED          | Maps the script's virtual path to the physical path used to call the script. This is done by taking any PATH_INFO component of the request URI and performing any virtual-to-physical translation appropriate.                                                                                                                                  |
| QUERY_STRING             | The query string that is contained in the request URL after the path.                                                                                                                                                                                                                                                                           |
| REMOTE_ADDR              | Returns the IP address of the client that sent the request. For HTTP servlets, the value returned is the same as the value of the CGI variable REMOTE_ADDR.                                                                                                                                                                                     |
| REMOTE_HOST              | The fully-qualified name of the client that sent the request, or the IP address of the client if the name cannot be determined. For HTTP servlets, the value returned is the same as the value of the CGI variable REMOTE_HOST.                                                                                                                 |
| REMOTE_USER              | Returns the login of the user making this request if the user has been authenticated, or null if the user has not been authenticated.                                                                                                                                                                                                           |
| REQUEST_METHOD           | Returns the name of the HTTP method with which this request was made. For example, GET, POST, or PUT.                                                                                                                                                                                                                                           |
| SCRIPT_NAME              | Returns the part of the URL from the protocol name up to the query string in the first line of the HTTP request.                                                                                                                                                                                                                                |

Table 63. HTTP server variables (continued)

| Variable        | Description                                                                                                                                                                                                                                         |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SERVER_NAME     | Returns the host name of the server that received the request. For HTTP servlets, it is the same as the value of the CGI variable SERVER_NAME.                                                                                                      |
| SERVER_PORT     | Returns the port number on which this request was received. For HTTP servlets, the value returned is the same as the value of the CGI variable SERVER_PORT.                                                                                         |
| SERVER_PROTOCOL | Returns the name and version of the protocol the request uses in the following form: protocol/majorVersion.minorVersion. For example, HTTP/1.1. For HTTP servlets, the value returned is the same as the value of the CGI variable SERVER_PROTOCOL. |
| SERVER_SOFTWARE | Returns the name and version of the servlet container on which the servlet is running.                                                                                                                                                              |
| HTTP_COOKIE     | HTTP Cookie String.                                                                                                                                                                                                                                 |
| WEBTOP_USER     | The user name of the user who is logged in.                                                                                                                                                                                                         |
| NCHOME          | The NCHOME environment variable.                                                                                                                                                                                                                    |

## Registering CGI scripts

After you have installed a CGI script on the server, for security purposes, the script must be registered. CGI scripts cannot be used as tools until they are registered.

### Before you begin

**UNIX** **Linux** Ensure that the CGI script has the appropriate file permissions.

The registration process authorizes execution, ensures that any SmartPage tags present in the source are processed correctly, and attaches a group to the script in order to minimize the possibility of misuse.

To register a CGI script:

1. Save the CGI script (for example, **nco\_ping**) in the *webgui\_home\_dir/etc/cgi-bin* directory.
2. Click **Administration > Event Management Tools > CGI Registry**.
3. Click **Register**.
4. In the Register CGI window, use the following fields and buttons to register the CGI script:

**Name** Type a name for the CGI script. This does not have to be the same as the file name of the CGI script.

By default, the following characters may not be used in names:

\$ ! £ % ^ & \* ( ) + = ~ ` ~ # @ ' : ; < > { } [ ] ? / \ \ | , "

By default, the following characters may not be used as the initial character of names:

/ \ \ \* ? " < > | & .

These invalid characters are defined in the following file:

*webgui\_home\_dir/etc/illegalChar.prop*



### Use Smartpage commands

Select this check box if the output of the CGI script is HTML data that contains SmartPage commands.

### File name

Type the file name of the script in the text field. It is not necessary to include the path.

### Groups for this CGI

If you want to restrict access to the CGI script, click **Groups** and select the groups to which you want to allow access. The default group is \*, where all users have access.

5. Click **Save**.

## Results

The CGI script is now registered for use in the Web GUI, and is displayed as an entry in the **Available CGIs** registration list of the **CGI Registry**.

## What to do next

To change an entry in the list, select the entry and click **Modify**. To remove an entry, select the name and click **Unregister**.

## Creating CGI tools

Create a CGI tool that runs a CGI script from the AEL to process and return ObjectServer field information.

## Before you begin

Before creating a CGI tool, you must create a CGI script and register the script for use in the Web GUI.

You can use an existing CGI tool as a template.

1. In the navigation, click **Administration > Event Management Tool > Tool Creation**
2. In the Tool Creation, click **Create Tool**.
3. Select **CGI/URL** from the **Type** list.
4. Optional: To copy an existing tool, select it from the list of tools displayed and click **Copy Tool**.
5. Type a name for the tool in the **Name** field. Do not use spaces or special characters in the name.

By default, the following characters cannot be used in tool names:

\$ ! % ^ & \* ( ) + = ~ ` ~ # @ ' : ; < > { } [ ] ? / \ \ | , "

By default, the following characters cannot be used as the initial character of tool names:

/ \ \ \* ? " < > | & .

These invalid characters are defined in the following file:

*webgui\_home\_dir/etc/illegalChar.prop*

6. Click **Show Data Sources** to display a list of available data sources and select the data sources that you require.
7. Complete the following tool configuration fields:

**URL** Type the location of the CGI script. By default this field contains the correct path for the cgi-bin directory on the local Web GUI server.



The \$(SERVER) keyword is resolved at runtime to *protocol://host:port/ibm/console/webtop*. Append the path with the file name of the script that you want to associate with the tool.

To pass field data to a script on a remote server, replace \$(SERVER) with an external URL address. For example *http://www.ibm.com*.

The \$(NGFSERVER) keyword resolves at runtime to *protocol://host:port*.

**Fields** Click **Show** and select the ObjectServer columns that you want to pass as arguments to the tool from the **Available** column. If you select more than one data source from the **Data Sources** list, the **Fields** list contains only columns that are common to all data sources.

To pass a full list of all selected rows to a tool, select **Serial**, select the **Execute for each selected row** check box, and clear the **Window for each selected row** check box. At runtime, the **Serial** field resolves to the **\$selected\_rows.Serial** parameter. If you select more than one data source from the **Data Sources** list, add the **\$selected\_rows.datasourcesource** parameter to distinguish between identical serial numbers that originate in different data sources.

#### Method

Specify the method for submitting field data to the CGI script:

- **GET**: Appends the name-value pairs to the URL, and is therefore useful if you want to bookmark the page containing the output.
- **POST**: Encodes the name-value pairs inside the body of the HTTP request. Note that firewalls can be configured to intercept and destroy this data stream, causing the form to be interpreted as empty.

The CGI script receives the data via a **QUERY\_STRING** environment variable regardless of the method chosen. This differs from the CGI convention where, if a method of **POST** is used, the script receives data via **stdin**.

#### Open in

Select either the **New window** or **Specific window** radio button. If you select **Specific window**, type a name for the window in the adjacent text field.

#### Execute for each selected row

Select this checkbox if you want the tool to run against all selected rows individually within the AEL. Clear the check box if you want the tool to run against only the first row in the selection.

#### Window for each selected row

Select this check box to open a separate window for each selected row in the AEL.

8. Define access for tools based on the groups that a user belongs to and the class of an event against which the tool is deployed:

**Group** Select the group that you want to access the tool and click **>**. To give all groups access to the selected tool, click **>>**. Users must be members of a selected group to use the tool. If you selected multiple data sources, this list displays all groups from all selected data sources.

**Class** Select the class that you want to access the tool and click **>**. To give all classes access to the selected tool, click **>>**.

**Tip:** Each event in the ObjectServer has an associated Class field. The value of this field is set by the event source. The Class field typically describes what kind of device an event comes from. If the class of an event matches any of the classes you select, the tool is available when the user selects the event in the AEL.

If you selected multiple data sources, this list displays classes from all selected data sources.

9. Click **Save**.

The tool appears as an entry in the list of available tools and is now available for use in the Active Event List (AEL).

10. Select either the **Open in: New window** or **Open in: Specific window** radio button.

| Option          | Description                                                                                      |
|-----------------|--------------------------------------------------------------------------------------------------|
| New Window      | Displays the output of the tool in a new browser window.                                         |
| Specific Window | Opens the tool output in a window of your choice. Enter a name for the window in the text field. |

## Example

### Example 1: Using values from cookies

CGI scripts can also use values from cookies that originate from the same domain as the Web GUI server.

**Attention:** Using browser cookie values in CGI scripts is an advanced task. You are responsible for setting browser cookies and using cookie values correctly.

To reference a value from a cookie, use the following syntax:

```
{%cookie.cookieName}
```

In this syntax, *cookieName* is the name of the cookie. For example, the full syntax of referencing a value from a cookie is:

```
$(SERVER)/cgi-bin/scriptName.cgi?parameterName={%cookie.cookieName}
```

### Example 2: Referencing a value from an ObjectServer field

To reference a value from an ObjectServer field in a query string, use the following syntax:

```
{@fieldName}
```

In this syntax, *fieldName* is the name of the ObjectServer field.

### Example 3: Showing a query string

The following example shows a query string using the Node and Summary fields from the ObjectServer:

```
protocol://server:port/ibm/console/webtop/cgi-bin/
scriptName.cgi?node={@Node}&abstract={@Summary}
```

## What to do next

For users to have access to a new tool in the AEL, you must create a menu entry for it.

### Related tasks

“Adding tools to a menu” on page 277

“Modifying tools” on page 267

### Related reference

“Tool runtime parameters” on page 274

Appendix F, “URLs for opening Web GUI pages,” on page 415

## Creating SQL tools

Create SQL tools that contain SQL instructions for modifying the event data stored in the ObjectServer alerts.status and alerts.journal data tables. The SQL tools can be run from within the Active Event List (AEL).

Any SQL tool that modifies ObjectServer data must be run by a Web GUI user that also exists in the ObjectServer as a write user.

To create an SQL tool:

1. In the navigation, click **Administration > Event Management Tool > Tool Creation**.
2. On the Tool Creation page, click **Create Tool**.
3. Select **SQL** from the **Type** list.
4. Optional: To use an existing tool as a template, select it from the list of tools displayed and click **Copy Tool**.
5. Click **Data Source** to select the data sources against which you want to run the SQL instructions. The data source selection also specifies the user groups or classes that are used to define the access criteria.
6. Enter a name for the tool in the **Name** field. Do not use spaces or special characters in the name.

By default, the following characters cannot be used in tool names:

\$ ! £ % ^ & \* ( ) + = ~ ` ~ # @ ' : ; < > { } [ ] ? / \ \ | , "

By default, the following characters cannot be used as the initial character of tool names:

/ \ \ \* ? " < > | & .

These invalid characters are defined in the following file:

`webgui_home_dir/etc/illegalChar.prop`

7. On the **SQL** table, complete the following fields:

### SQL Commands

Type the SQL commands that you want to use to update the alerts.status table in the ObjectServer.

### Execute for each selected row

Select this checkbox if you want the tool to run against all selected rows individually within the AEL.

Clear the check box if you want the tool to run against only the first row in the selection.

**Important:** Do not select this checkbox if the SQL instructions explicitly state that the command must run against all rows.

8. Optional: To update the alerts.journal table of the selected data source or data sources, click **SQL** and complete the following fields:

**Journal Entry**

Type the SQL commands that you want to use to update the alerts.journal table in the ObjectServer.

If you leave the **Journal Entry** field empty, no journal entry is recorded when the tool is run. If you want a blank journal entry to be recorded when the tool is run, enter a space or type another character in the field.

**Tip:** You can use a Multiline String prompt for forced journal entry.

**Execute for each selected row**

Select this checkbox if you want the tool to run against all selected rows individually within the AEL, and modify all corresponding journal entries. Clear the check box if you want the tool to run against only the first row in the selection, and modify the corresponding journal entry.

9. Define access for tools based on the groups that a user belongs to and the class of an event against which the tool is deployed:

**Group** Select the group that you want to access the tool and click >. To give all groups access to the selected tool, click >>. Users must be members of a selected group to use the tool. If you selected multiple data sources, this list displays all groups from all selected data sources.

**Class** Select the class that you want to access the tool and click >. To give all classes access to the selected tool, click >>.

**Tip:** Each event in the ObjectServer has an associated Class field. The value of this field is set by the event source. The Class field typically describes what kind of device an event comes from. If the class of an event matches any of the classes you select, the tool is available when the user selects the event in the AEL.

If you selected multiple data sources, this list displays classes from all selected data sources.

If no group or class is selected, users of any group can execute the tool, and the tool can be executed against events of any class.

10. Click **Save**.

## Results

The tool appears as an entry in the list of available tools and is now available for use in the Active Event List (AEL).

## Examples

The following SQL command updates the alerts.status table in the ObjectServer by setting the OwnerGID of the selected rows to the value that is chosen from the \$prompt menu. The \$prompt menu is a system-managed sub-menu of the groupassign tool that contains all the user groups.

```
update alerts.status set OwnerGID=$prompt.groupassign
where Serial in ($selected_rows.Serial);
```

The following SQL command updates the alerts.journal table in the ObjectServer:

Alert assigned to group `CONVERSION($prompt.groupassign)` by `%username`.

## What to do next

For users to have access to a new tool in the AEL, you must create a menu entry for it.

### Related concepts

“SQL tools” on page 252

### Related tasks

“Creating command-line tools”

“Adding tools to a menu” on page 277

“Modifying tools” on page 267

## Creating command-line tools

Create command strings that run a command-line action on a client system. With the command string available as a tool, users can instruct a client system to open a command prompt and pass field data to an application.

To create a command-line tool:

1. In the navigation, click **Administration > Event Management Tool > Tool Creation**
2. On the Tool Creation, click **Create Tool**.
3. Select **Command** from the **Type** list.
4. Optional: To use an existing tool as a template, select it from the list of tools displayed and click **Copy Tool**.
5. Click **Data Source** to select the data sources against which you want to run the SQL instructions. The data source selection also specifies the user groups or classes that are used to define the access criteria.
6. Type a name for the tool in the **Name** field above the Tool Configuration dialog. Do not use spaces or special characters in the name.

By default, the following characters cannot be used in tool names:

`$ ! £ % ^ & * ( ) + = ~ ` ~ # @ ' : ; < > { } [ ] ? / \ \ | , "`

By default, the following characters cannot be used as the initial character of tool names:

`/ \ \ * ? " < > | & .`

These invalid characters are defined in the following file:

`webgui_home_dir/etc/illegalChar.prop`

7. Complete the following tool configuration fields:

### Platform

Select this checkbox to specify which client operating system types can access this tool from the AEL.

### Command

For each selected client operating system, modify the default entry and type the command to launch the target application. Include the full path to the command.

**Tip:** On the Windows operating system use the following construct to ensure the DOS console closes when the tool completes:

```
start /b cmd /k
```

In addition you can launch multiple applications from one tool by inserting && between each command. For example, to launch two Internet Explorer windows use the following command: **start /b cmd /k cmd iexplore.exe && start /b cmd /k iexplore.exe**

#### **Execute for each selected row**

Select this checkbox if you want the tool to run against all selected rows individually within the AEL. Clear the check box if you want the tool to run against only the first row in the selection.

8. Define access for tools based on the groups that a user belongs to and the class of an event against which the tool is deployed:

**Group** Select the group that you want to access the tool and click >. To give all groups access to the selected tool, click >>. Users must be members of a selected group to use the tool. If you selected multiple data sources, this list displays all groups from all selected data sources.

**Class** Select the class that you want to access the tool and click >. To give all classes access to the selected tool, click >>.

**Tip:** Each event in the ObjectServer has an associated Class field. The value of this field is set by the event source. The Class field typically describes what kind of device an event comes from. If the class of an event matches any of the classes you select, the tool is available when the user selects the event in the AEL.

If you selected multiple data sources, this list displays classes from all selected data sources.

If no group or class is selected, users of any group can execute the tool, and the tool can be executed against events of any class.

9. Click **Save**.

## **Results**

The tool appears as an entry in the list of available tools and is now available for use in the Active Event List (AEL).

## **What to do next**

For users to have access to a new tool in the AEL, you must create a menu entry for it.

### **Related concepts**

“Command-line tools” on page 252

### **Related tasks**

“Creating command-line tools” on page 263

“Creating SQL tools” on page 261

“Creating script tools” on page 265

“Modifying tools” on page 267

“Adding tools to a menu” on page 277

### **Related reference**

“Script tool examples” on page 266

## Creating script tools

Create script tools to pass contextual data from selected events in the Active Event List (AEL) to separate portlet instances or to dynamically customized page content.

### Before you begin

Script tool syntax follows JavaScript rules. The command text might also contain variables that are evaluated when the script tool is executed.

To create a script tool:

1. In the navigation, click **Administration > Event Management Tool > Tool Creation**.
2. In the Tool Creation page, click **Create Tool**.
3. Select **Script** from the **Type** list.
4. Optional: To use an existing tool as a template, select it from the list of tools displayed and click **Copy Tool**.
5. Click **Data Source** to select the data sources against which you want to run the SQL instructions. The data source selection also specifies the user groups or classes that are used to define the access criteria.

6. Type a name for the tool in the **Name** field above the Tool Configuration dialog. Do not use spaces or special characters in the name.

By default, the following characters cannot be used in tool names:

\$ ! £ % ^ & \* ( ) + = ~ ` ~ # @ ' : ; < > { } [ ] ? / \ \ | , "

By default, the following characters cannot be used as the initial character of tool names:

/ \ \ \* ? " < > | & .

These invalid characters are defined in the following file:

*webgui\_home\_dir/etc/illegalChar.prop*

7. Complete the following tool configuration fields:

#### Script Commands

Type the script that you want to use to configure the new tool.

#### Execute for each selected row

Select this checkbox if you want the tool to run against all selected rows individually within the AEL. Clear the check box if you want the tool to run against only the first row in the selection.

8. Define access for the tool based on the groups that a user belongs to and the class of an event against which the tool is deployed:

**Group** Select the group that you want to access the tool and click >. To give all groups access to the selected tool, click >>. Users must be members of a selected group to use the tool. If you selected multiple data sources, this list displays all groups from all selected data sources.

**Class** Select the class that you want to access the tool and click >. To give all classes access to the selected tool, click >>.

**Tip:** Each event in the ObjectServer has an associated **Class** field. The value of this field is set by the event source. The **Class** field typically describes what kind of device an event comes from. If the class of an event matches any of the classes you select, the tool is available when the user selects the event in the AEL.

If you selected multiple data sources, this list displays classes from all selected data sources.

If no group or class is selected, users of any group can execute the tool, and the tool can be executed against events of any class.

9. Click **Save**.

## Results

The tool appears as an entry in the list of available tools and is now available for use in the Active Event List (AEL).

## What to do next

For users to have access to a new tool in the AEL, you must create a menu entry for it first.

### Related tasks

“Creating command-line tools” on page 263

## Script tool examples

Script tool syntax follows JavaScript rules, and you can create many different script tools to extract data from events, as illustrated by these examples.

### Sample: itnmBroadcast Event

This script tool invokes a JavaScript method from the AEL portlet page and is included in Network Manager IP Edition as default functionality to enable intra-portlet communication.

The script commands are as follows:

```
{ $appletparam.iscPortletNamespace } sendPortletEvent({ 'name':
'http://ibm.com/TIP#BroadcastEvent', 'client':
'NW', 'entityId': '{@NmosEntityId}' });
```

The method invoked is uniquely signed by the use of the **iscPortletNamespace** applet parameter, which is a unique signature for each instance of the AEL portlet. The `sendPortletEvent()` method takes a **itnmBroadcastEvent** as a parameter.

In the following example, the **itnmBroadcastEvent** has been configured with the following attributes:

- name: `http://ibm.com/TIP#BroadcastEvent`
- client: `NW`
- entityId: `{@NmosEntityId}`

At runtime, `{@NmosEntityId}` is resolved into the `NmosEntityId` field value of the selected event.

### Sample: Event “scratchpad”

This JavaScript code sample creates a simple “scratchpad” tool and illustrates the potential of script tools.

After you execute the scratchpad tool against a selected event, below the AEL a new area is displayed containing event data, such as a summary of the event, the time of occurrence, the node where the event occurred, and its severity. The event data is added to the page using the Document Object Model (DOM).

The script commands are as follows:



```

var str = 'Event \''{@Serial}\'' at Node \''{@Node}\'' has Summary \''{@Summary}\'',
Severity \''{@Severity}\'' and last occurred on ' + new Date({@LastOccurrence}*1000);

var scratchpad = document.getElementById("scratchpad_{$appletparam.EntityName}");

if (scratchpad == null)
{
 scratchpad = document.createElement("div");
 scratchpad.setAttribute("id", "scratchpad_{$appletparam.EntityName}");
 scratchpad.setAttribute("style", "overflow:auto;height:200px;border-width:1px;
border-style: solid;margin:5px;padding:5px;font-family:Verdana,Arial,Helvetica,
sans-serif;font-size:0.7em;");

 var header= document.createElement("h2");
 header.appendChild(document.createTextNode("Scratch Pad for entity
{$appletparam.EntityName}")); scratchpad.appendChild(header);
 document.body.appendChild(scratchpad);
}

var div = document.createElement("div");
var dl = document.createElement("dl");

var dt = document.createElement("dt");
dt.appendChild(document.createTextNode(new Date()));
dl.appendChild(dt);

var dd = document.createElement("dd");
dd.appendChild(document.createTextNode(str));
dl.appendChild(dd);

div.appendChild(dl);

scratchpad.appendChild(div);

```

### Related tasks

“Creating command-line tools” on page 263

## Modifying tools

You can modify the settings of existing CGI, SQL, command-line or script tools.

To modify a tool:

1. In the navigation, click **Administration > Event Management Tool > Tool Creation**.
2. In the Tool Creation page, select the tool you want to modify from the list of tools displayed.
3. Select a tool type from the **Type** list.
4. For SQL tool types: To modify the tool, use the following fields:

### SQL Commands

Type the SQL commands that you want to use to update the alerts.status table in the ObjectServer.

### Execute for each selected row

Select this checkbox if you want the tool to run against all selected rows individually within the AEL.

Clear the check box if you want the tool to run against only the first row in the selection.

**Important:** Do not select this checkbox if the SQL instructions explicitly state that the command must run against all rows.

### Journal Entry

Type the SQL commands that you want to use to update the alerts.journal table in the ObjectServer.

If you leave the **Journal Entry** field empty, no journal entry is recorded when the tool is run. If you want a blank journal entry to be recorded when the tool is run, enter a space or type another character in the field.

**Tip:** You can use a Multiline String prompt for forced journal entry.

### Execute for each selected row

Select this checkbox if you want the tool to run against all selected rows individually within the AEL, and modify all corresponding journal entries. Clear the check box if you want the tool to run against only the first row in the selection, and modify the corresponding journal entry.

5. For CGI tool types: To modify the tool, use the following fields:

**URL** Type the location of the CGI script. By default this field contains the correct path for the cgi-bin directory on the local Web GUI server. The \$(SERVER) keyword is resolved at runtime to *protocol://host:port/ibm/console/webtop*. Append the path with the file name of the script that you want to associate with the tool.

To pass field data to a script on a remote server, replace \$(SERVER) with an external URL address. For example *http://www.ibm.com*.

The \$(NGFSERVER) keyword resolves at runtime to *protocol://host:port*.

**Fields** Click **Show** and select the ObjectServer columns that you want to pass as arguments to the tool from the **Available** column. If you select more than one data source from the **Data Sources** list, the **Fields** list contains only columns that are common to all data sources.

To pass a full list of all selected rows to a tool, select **Serial**, select the **Execute for each selected row** check box, and clear the **Window for each selected row** check box. At runtime, the **Serial** field resolves to the **\$selected\_rows.Serial** parameter. If you select more than one data source from the **Data Sources** list, add the **\$selected\_rows.datasource** parameter to distinguish between identical serial numbers that originate in different data sources.

### Method

Specify the method for submitting field data to the CGI script:

- **GET:** Appends the name-value pairs to the URL, and is therefore useful if you want to bookmark the page containing the output.
- **POST:** Encodes the name-value pairs inside the body of the HTTP request. Note that firewalls can be configured to intercept and destroy this data stream, causing the form to be interpreted as empty.

The CGI script receives the data via a QUERY\_STRING environment variable regardless of the method chosen. This differs from the CGI convention where, if a method of POST is used, the script receives data via stdin.

**Open in**

Select either the **New window** or **Specific window** radio button. If you select **Specific window**, type a name for the window in the adjacent text field.

**Execute for each selected row**

Select this checkbox if you want the tool to run against all selected rows individually within the AEL. Clear the check box if you want the tool to run against only the first row in the selection.

**Window for each selected row**

Select this check box to open a separate window for each selected row in the AEL.

6. For command-line tool types: To modify the tool, use the following fields:

**Platform**

Select this checkbox to specify which client operating system types can access this tool from the AEL.

**Command**

For each selected client operating system, modify the default entry and type the command to launch the target application. Include the full path to the command.

**Tip:** On the Windows operating system use the following construct to ensure the DOS console closes when the tool completes:

```
start /b cmd /k
```

In addition you can launch multiple applications from one tool by inserting && between each command. For example, to launch two Internet Explorer windows use the following command: **start /b cmd /k cmd iexplore.exe && start /b cmd /k iexplore.exe**

**Execute for each selected row**

Select this checkbox if you want the tool to run against all selected rows individually within the AEL. Clear the check box if you want the tool to run against only the first row in the selection.

7. For script tool types: To modify the tool, use the following fields:

**Script Commands**

Type the script that you want to use to configure the new tool.

**Execute for each selected row**

Select this checkbox if you want the tool to run against all selected rows individually within the AEL. Clear the check box if you want the tool to run against only the first row in the selection.

8. Define access for tools based on the groups that a user belongs to and the class of an event against which the tool is deployed:

**Group** Select the group that you want to access the tool and click >. To give all groups access to the selected tool, click >>. Users must be members of a selected group to use the tool. If you selected multiple data sources, this list displays all groups from all selected data sources.

**Class** Select the class that you want to access the tool and click >. To give all classes access to the selected tool, click >>.

**Tip:** Each event in the ObjectServer has an associated Class field. The value of this field is set by the event source. The Class field typically

describes what kind of device an event comes from. If the class of an event matches any of the classes you select, the tool is available when the user selects the event in the AEL.

If you selected multiple data sources, this list displays classes from all selected data sources.

9. Click **Save**.

## Results

The tool appears as an entry in the list of available tools and is now available for use in the AEL.

### Related tasks

“Creating command-line tools” on page 263

“Creating CGI tools” on page 258

“Creating SQL tools” on page 261

“Copying tools” on page 271

## Renaming tools

You can change the name of any of the existing CGI, SQL, script or command-line tools.

If you rename a tool, the names of menu items that link to the tool might also be renamed. If a menu item has the same name as the tool, when you rename the tool the menu item is renamed automatically. If a menu item has a different name than the tool, when you rename the tool the menu item label does not change. In both cases, the link from the menu item to the tool is retained.

To rename a tool

1. In the navigation, click **Administration > Event Management Tool > Tool Creation**.
2. In the Tool Creation page, select the tool you want to rename.
3. Enter a new name for the tool in the **Name** field. Do not use spaces or special characters in the name.

By default, the following characters cannot be used in tool names:

`$ ! £ % ^ & * ( ) + = ~ ` ~ # @ ' : ; < > { } [ ] ? / \ \ | , "`

By default, the following characters cannot be used as the initial character of tool names:

`/ \ \ * ? " < > | & .`

These invalid characters are defined in the following file:

`webgui_home_dir/etc/illegalChar.prop`

4. Click **Save**.

The tool appears as an entry in the list of available tools and is now available for use in the Active Event List (AEL).

### Related tasks

“Renaming menu items” on page 280

## Copying tools

You can copy any of the existing CGI, SQL, script or command-line tools to have a copy of the tool, or to modify the configuration of the tool and save it as a new tool. This helps in using an existing tool configuration as a template for new tools.

To copy a tool:

1. Click **Administration > Event Management > Tool > Tool Creation**.
2. In the Tool Creation page, select the tool you want to copy from the list of tools displayed, and click **Copy Tool**.
3. In the **Name** field, type a new name for the tool. The tool name cannot contain spaces.

By default, the following characters cannot be used in tool names:

`$ ! £ % ^ & * ( ) + = ~ ` ~ # @ ' : ; < > { } [ ] ? / \ \ | , "`

By default, the following characters cannot be used as the initial characters of tool names:

`/ \ \ * ? " < > | & .`

These invalid characters are defined in the following file:

`.webgui_home_dir/etc/illegalChar.prop`

4. For SQL tool types: To modify the tool, use the following fields:

### SQL Commands

Type the SQL commands that you want to use to update the alerts.status table in the ObjectServer.

### Execute for each selected row

Select this checkbox if you want the tool to run against all selected rows individually within the AEL.

Clear the check box if you want the tool to run against only the first row in the selection.

**Important:** Do not select this checkbox if the SQL instructions explicitly state that the command must run against all rows.

### Journal Entry

Type the SQL commands that you want to use to update the alerts.journal table in the ObjectServer.

If you leave the **Journal Entry** field empty, no journal entry is recorded when the tool is run. If you want a blank journal entry to be recorded when the tool is run, enter a space or type another character in the field.

**Tip:** You can use a Multiline String prompt for forced journal entry.

### Execute for each selected row

Select this checkbox if you want the tool to run against all selected rows individually within the AEL, and modify all corresponding journal entries. Clear the check box if you want the tool to run against only the first row in the selection, and modify the corresponding journal entry.

5. For CGI tool types: To modify the tool, use the following fields:

**URL** Type the location of the CGI script. By default this field contains the correct path for the cgi-bin directory on the local Web GUI server. The `$(SERVER)` keyword is resolved at runtime to `protocol://host:port/`

ibm/console/webtop. Append the path with the file name of the script that you want to associate with the tool.

To pass field data to a script on a remote server, replace \$(SERVER) with an external URL address. For example `http://www.ibm.com`.

The \$(NGFSERVER) keyword resolves at runtime to `protocol://host:port`.

**Fields** Click **Show** and select the ObjectServer columns that you want to pass as arguments to the tool from the **Available** column. If you select more than one data source from the **Data Sources** list, the **Fields** list contains only columns that are common to all data sources.

To pass a full list of all selected rows to a tool, select **Serial**, select the **Execute for each selected row** check box, and clear the **Window for each selected row** check box. At runtime, the **Serial** field resolves to the `$selected_rows.Serial` parameter. If you select more than one data source from the **Data Sources** list, add the `$selected_rows.datasource` parameter to distinguish between identical serial numbers that originate in different data sources.

#### Method

Specify the method for submitting field data to the CGI script:

- **GET**: Appends the name-value pairs to the URL, and is therefore useful if you want to bookmark the page containing the output.
- **POST**: Encodes the name-value pairs inside the body of the HTTP request. Note that firewalls can be configured to intercept and destroy this data stream, causing the form to be interpreted as empty.

The CGI script receives the data via a `QUERY_STRING` environment variable regardless of the method chosen. This differs from the CGI convention where, if a method of **POST** is used, the script receives data via `stdin`.

#### Open in

Select either the **New window** or **Specific window** radio button. If you select **Specific window**, type a name for the window in the adjacent text field.

#### Execute for each selected row

Select this checkbox if you want the tool to run against all selected rows individually within the AEL. Clear the check box if you want the tool to run against only the first row in the selection.

#### Window for each selected row

Select this check box to open a separate window for each selected row in the AEL.

6. For command-line tool types: To modify the tool, use the following fields:

#### Platform

Select this checkbox to specify which client operating system types can access this tool from the AEL.

#### Command

For each selected client operating system, modify the default entry and type the command to launch the target application. Include the full path to the command.

**Tip:** On the Windows operating system use the following construct to ensure the DOS console closes when the tool completes:

```
start /b cmd /k
```

In addition you can launch multiple applications from one tool by inserting && between each command. For example, to launch two Internet Explorer windows use the following command: **start /b cmd /k cmd iexplore.exe && start /b cmd /k iexplore.exe**

#### **Execute for each selected row**

Select this checkbox if you want the tool to run against all selected rows individually within the AEL. Clear the check box if you want the tool to run against only the first row in the selection.

7. For script tool types: To modify the tool, use the following fields:

#### **Script Commands**

Type the script that you want to use to configure the new tool.

#### **Execute for each selected row**

Select this checkbox if you want the tool to run against all selected rows individually within the AEL. Clear the check box if you want the tool to run against only the first row in the selection.

8. Define access for the tool based on the groups that a user belongs to and the class of an event against which the tool is deployed:

**Group** Select the group that you want to access the tool and click >. To give all groups access to the selected tool, click >>. Users must be members of a selected group to use the tool. If you selected multiple data sources, this list displays all groups from all selected data sources.

**Class** Select the class that you want to access the tool and click >. To give all classes access to the selected tool, click >>.

**Tip:** Each event in the ObjectServer has an associated Class field. The value of this field is set by the event source. The Class field typically describes what kind of device an event comes from. If the class of an event matches any of the classes you select, the tool is available when the user selects the event in the AEL.

If you selected multiple data sources, this list displays classes from all selected data sources.

9. Click **Save**.

## **Results**

The tool appears as an entry in the list of available tools and is now available for use in the AEL.

## **What to do next**

For users to have access to a new tool in the AEL, you must create a menu entry for it first.

#### **Related tasks**

“Modifying tools” on page 267



## Deleting tools

You can delete any of the existing CGI, SQL, script or command-line tools to remove it from the available tools in the AEL.

1. Click **Administration > Event Management Tool > Tool Creation**. The Tool Creation page is displayed.
2. Select the specific tool you want to delete from the list of tools displayed, and click **Delete Tool**.
3. Click **OK** when prompted for confirmation. A message is displayed confirming that the tool has been deleted, and the tool is removed from the list of tools.

## Tool runtime parameters

Tools can contain one or more runtime parameters; the values of these parameters are determined when the tool is executed.

### Runtime parameters

**@*fieldname***

*fieldname* is the name of a field in the alerts.status table in an ObjectServer. When the tool is executed, this variable is resolved to the value of the *fieldname* of the selected row.

**\$selected\_rows.*fieldname***

*fieldname* is the name of a field in the alerts.status table in an ObjectServer. When the tool is executed, this variable is resolved to a comma-separated string of the field values of all the selected rows.

If your environment uses multiple data sources, use the **\$selected\_rows.serial** parameter in combination with the **\$selected\_rows.datasource** parameter. This combination ensures that no duplicate serials are returned by tools because a unique combination of serial and data source is returned.

**\$prompt.*promptname***

*promptname* is the name of a defined Web GUI prompt. When the tool is executed, this variable is resolved to the prompt value. This might be a comma-separated list in the case where a prompt contains multiple values.

**%uid** *uid* is resolved to the user ID of the ObjectServer user. This might be an empty string if not available.

**%username**

*username* is resolved to the user name of the user. If the user is not an ObjectServer user, the string *#unknownUser* is substituted.

**%gid** *gid* is resolved to the group ID of the ObjectServer user. This might be an empty string if not available.

**%datasource**

*datasource* is resolved to the name of the data source for which the AEL is displaying events.

**%cookie.*cookieName***

*cookieName* is resolved to the value of the cookie named *cookieName*. This might be an empty string if not available.

**\$appletparam.*parameterName***

*parameterName* is the name of an applet parameter. When the tool is executed, this variable is resolved to the value of the applet parameter. This might be an empty string if not available.



### **\$selected\_rows.datasource**

Identifies the data source in which the row originates. Note that this parameter does not resolve to a field in an ObjectServer. Use this parameter for tools that are to be run against events from multiple data sources, to distinguish between identical data from different data sources.

### **CONVERSION(@fieldname)**

*fieldname* is the name of a field in the alerts.conversion table in an ObjectServer. When the tool is executed, this variable is resolved to the conversion value of the field value for the selected row.

Example: Alert prioritized from CONVERSION(@Severity)

### **CONVERSION(\$selected\_rows.fieldname)**

*fieldname* is the name of a field in the alerts.status table in an ObjectServer. When the tool is executed, this variable is resolved to a comma-separated string of the conversion values of the field values for all selected rows.

### **CONVERSION(\$prompt.promptname)**

*promptname* is the name of a prompt. When the tool is executed, this variable is resolved to the prompt label (as opposed to the prompt value).

Example: Alert assigned to CONVERSION(\$prompt.userassign)

**Important:** If \$selected\_rows.fieldname is used (with or without CONVERSION), clear the tool's **Execute for each selected row** check box. Setting this check box can result in ObjectServer performance degradation.

## **Specifying parameters**

For CGI/URL, Command and Script tools, enclose these parameters (where supported) in braces. For example:

```
start cmd /k %WINDIR%\SYSTEM32\PING.EXE {$prompt.hostname}
```

For conversions the CONVERSION keyword is enclosed in the braces.

Braces are not necessary for parameters in SQL tools (including in the Journal component).

## **Parameters supported by each type of tool**

The following table shows the parameters that each type of tool supports. Refer to the notes at the end of the table for more specific information on the support for certain parameters.

Table 64. Parameters supported by each type of tool

| Parameter                 | SQL<br>(including<br>Journal) | CGI/URL        | Command | Script |
|---------------------------|-------------------------------|----------------|---------|--------|
| @fieldname                | Y                             | Y              | Y       | Y      |
| \$selected_rows.fieldname | Y                             | Y <sup>1</sup> | Y       | Y      |
| \$prompt.promptname       | Y                             | Y              | Y       | Y      |
| %uid                      | Y                             | Y              | Y       | Y      |
| %username                 | Y                             | Y              | Y       | Y      |
| %gid                      | Y                             | Y              | Y       | Y      |

Table 64. Parameters supported by each type of tool (continued)

| Parameter                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | SQL<br>(including<br>Journal) | CGI/URL        | Command | Script |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------|----------------|---------|--------|
| %datasource                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Y                             | Y <sup>2</sup> | Y       | Y      |
| %cookie.cookieName                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | N                             | Y              | Y       | Y      |
| \$appletparam.parameterName                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | N                             | N              | N       | Y      |
| \$selected_rows.datasource                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Y                             | Y              | Y       | Y      |
| CONVERSION(@fieldName)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Y                             | Y              | Y       | Y      |
| CONVERSION(\$selected_rows.fieldName)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Y                             | Y <sup>3</sup> | Y       | Y      |
| CONVERSION(\$prompt.promptName)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Y                             | Y              | Y       | Y      |
| <b>Notes:</b> <ol style="list-style-type: none"> <li>1. This parameter can also be implicitly inserted through the selection of fields during CGI/URL tool creation. At runtime the field resolves to the \$selected_rows.fieldName parameter.</li> <li>2. At runtime the CGI/URL tool inserts the datasource parameter even if this parameter is not explicitly set during tool creation.</li> <li>3. This parameter can also be implicitly inserted through the selection of fields during CGI/URL tool creation. At runtime the field resolves to the CONVERSION.\$selected_rows.fieldName parameter.</li> </ol> |                               |                |         |        |

## Customizing AEL Menus

You can change the content of the menus of the Active Event List (AEL). You can add tool entries to the menus, create new sub-menus, and modify or delete menu items. A tool can include a prompt window or pop-up menu for the user to enter information, and you can edit these prompts or create new prompts.

### The Alerts and Tools menus

The **Alerts** and **Tools** menus are configurable menus that can be accessed from the AEL. The **Alerts** menu contains a number of SQL tools you can use to interact with alert data and manipulate the data. By default, the **Tools** menu contains CGI tools and local (command-line) tools.

The **Alerts** menu can be accessed both from the AEL toolbar and by right-clicking an event in the AEL. The **Tools** menu can be accessed from the AEL toolbar.

You can add your own tools and sub-menus to either the **Alerts** menu or the **Tools** menu. To do this you must use the Tools Editor to create the tool you want to add, and then add the tools to the menus using the Menus Editor.

Tools in the **Tools** menu can be configured with access criteria that apply to users and events. Tools are visible only if the access criteria applied to them are met, or when no criteria are set because no groups or classes have been defined.

If multiple events are selected in the AEL, all access criteria must be satisfied for all selected events for a tool to be displayed.

By default, no access criteria are defined for any tools. Tools that have no access criteria defined are displayed for all users for all events. Changes in access criteria take effect when the AEL is reloaded, without the need to restart the Web GUI server.

The default content of the AEL menu bar cannot be changed. You cannot create new top-level menus.

#### Related tasks

“Creating event management tools” on page 250

## Adding tools to a menu

If you have created a new tool, you need to add the tool as an entry to an AEL menu in order to use the tool. You can also add other existing tools to menus.

To add tools to a menu:

1. Click **Administration > Event Management Tools**, and click **Menu Configuration**.
2. From the **Available menus** list, select a menu to which you want to add a tool and click **Modify**. The **Menus Editor** window is displayed.
3. Select **tool** from the **Available items** list to view a list of available tools.
4. Add an existing tool to the selected menu or create a new tool and then add the tool to the selected menu:
  - a. Select a tool from the list and click **Add selected item**. The tool is added to the list in the **Current items** pane on the left side of the page.
  - b. Select **<new tool>** from the list and click **Add selected item**. The **Tool Configuration** window opens. Define the new tool using the **Tool Configuration** and **Access Criteria** dialogs and click **Save**. Close the **Save Confirmation** window to return to the **Menus Editor** window. The tool is added to the **Current items** list and **Available items** lists.

**Tip:** If you have several tools in one menu, you can make the menu easier to read by adding a separator. Select **<separator>** from the **Available items** list and click **Add selected item**.

5. Click **Save**. The menu is now updated in the AEL, and is added as an entry in **Available menus**.

#### Related tasks

“Creating event management tools” on page 250

## Modifying existing tools

You can edit the tools available in the menus.

To edit a tool menu entry:

1. Click **Administration > Event Management Tools**, and click **Menu Configuration**.
2. Select the menu that contains the tool you want to modify and click **Modify**. The **Menus Editor** window is displayed.
3. Select the tool you want to edit from the **Current items** list and click the **Edit** button. The **Tool Configuration** window is displayed.
4. Edit the tool using the **Tool Configuration** and **Access Criteria** dialogs.
5. Click **Save** and close the **Save Confirmation** window to return to the **Menus Editor** window. The tool is saved with the changed settings.

## Related tasks

"Modifying tools" on page 267

## Creating submenus

You can create submenus of the **Alerts** and **Tools** AEL menus. Each submenu can contain tool entries, separator bars, and other submenus. A submenu can be used in different menus.

To create a new submenu:

1. Click **Administration > Event Management Tools**, and click **Menu Configuration**.
2. Click **New**. The **Menu Editor** window is displayed, with the **Name** and **Label** fields and **Current items** list blank.
3. Enter the properties for the new submenu using the following fields and buttons:

**Name** Type a name for the new submenu. Do not use spaces or special characters in the name.

**Label** Type a label for the submenu. This is the text that is displayed in the AEL.

### Mnemonic

Select a key entry if you want users to be able to display this menu in the AEL using the keyboard shortcut of Alt and selected character.

### Available items

Select **tool** to list all available tools you can then add to the menu, or select **menu** to list all available menus you can add to the menu. Also, you can select **<separator>** to add a line between menus and tools. For example, if you have several tools in one menu, you can make the menu easier to read by adding a separator.

Select the tool or menu you want to make available in the menu and click **Add selected item**.

### Current items

Lists all the tools and menus added to the menu. Select the tool or menu you want to remove from the menu and click **Remove selected item**. Use the arrow buttons to the right of this list to change the display order of the tools and menus within the menu.

### Rename

Select a tool in the **Current items** list and click this button to change the name of the tool as it appears in the AEL.

**Label** Type a label for the tool. This is the text that appears in the AEL.

### Mnemonic

Select a key entry if you want users to be able to display this menu in the AEL using the keyboard shortcut of Alt and selected character.

### ShortCut

Type a shortcut character if you want users to be able to display this menu in the AEL using the keyboard shortcut of Ctrl and the character provided here.

**Restriction:** The rename options are available only for tools.

**Edit** Select a tool or menu in the **Current items** list and click **Edit** to modify the tool or menu settings. If you have a tool selected, the Tool Configuration window opens. If you have a menu selected, you are asked to save changes and then redirected to the **Menus Editor** where you can modify the setup of the menu.

4. Click **Save** and **Ok**. The new submenu is displayed in the list of available menus.

#### Related tasks

“Adding submenus to a menu”

## Adding submenus to a menu

You can customize what submenus are available in the **Alerts** and **Tools** AEL menus.

To add a submenu to a menu:

1. Click **Administration > Event Management Tools**, and click **Menu Configuration**.
2. Select the menu to which you want to add the sub-menu and click **Modify**.
3. In the **Available items** list, select **menu**. A list of available menus is displayed.
4. You can add an existing menu as a sub-menu or create a new menu and add the menu as a sub-menu:
  - a. Select the menu you want to add and click **Add selected item**. The menu is added to the list in the **Current items** list. Optionally, configure the sub-menu using the **Edit** button.
  - b. Select **<new menu>** from the list and click **Add selected item**. You are asked whether you want to save the changes to the menu that you are modifying. Click **Cancel** if you do not want to add a new menu, click **Yes** to save changes, or **No** to continue without saving changes. The Menu Editor window is displayed for the new menu that you are adding. Define the new menu.

**Note:** Only menu entries that are logically permitted by the nesting rules are displayed in the list. If the entry you are modifying is a sub-menu itself, the menu or menus above it are not shown.

5. Click **Save** and **Ok**. The new menu is added as a sub-menu to the menu that you chose. The new menu is visible in the list of available menus.

#### Related tasks

“Creating submenus” on page 278

## Deleting submenus

You can remove any submenu created by users. You cannot delete the **Alerts** or **Tools** menus.

To delete a menu:

1. Click **Administration > Event Management Tools**, and click **Menu Configuration**.
2. Select the menu that you want to delete.
3. Click **Delete** and confirm the deletion.

## Removing items from a menu

You can remove submenus, tools, and separators from an existing menu. Removing a tool or submenu from a menu does not delete the tool or submenu.

To remove items from a menu:

1. Click **Administration > Event Management Tools**, and click **Menu Configuration**.
2. Select the menu from which you want to remove items.
3. Click **Modify**. The Menus Editor window is displayed.
4. Select the item that you want to remove from the **Current items** list and click **Remove selected item**.

**Tip:** If you select an entry in the list, an icon displayed above the list to the right indicates whether that item is a tool, a menu, or a separator.

5. Click **Save** to save your changes.

### Related tasks

"Deleting tools" on page 274

"Deleting submenus" on page 279

## Changing the order of items in a menu

You can move items up and down within a menu to create a customized order of all items.

To change the order in which items are displayed in a menu:

1. Click **Administration > Event Management Tools**, and click **Menu Configuration**.
2. Select the menu that you want to modify.
3. Click **Modify**. The Menus Editor window is displayed.
4. Select the item that you want to move from the **Current items** list.
  - Click **Top** to move the item to the top of the menu.
  - Click **Bottom** to move the item to the bottom of the menu.
  - Click **Up** to move the item one place up in the menu.
  - Click **Down** to move the item one place down in the menu.

**Tip:** If you select an entry in the list, an icon displayed above the list to the right indicates whether that item is a tool, a menu, or a separator.

5. Click **Save** to save your changes.

## Renaming menu items

You can change the label of a tool that is displayed in the AEL menu. To rename the tool itself, edit the name of the tool in the Tools Editor.

To change the label for a tool menu item:

1. Click **Administration > Event Management Tools**, and click **Menu Configuration**.
2. Select the menu that you want to modify.
3. Click **Modify**. The Menus Editor window opens.
4. Select the tool menu item you want to modify from the **Current items** list and click **Rename**. A separate window is displayed.

**Tip:** If you select a submenu or other item that you cannot rename in this way, the **Rename** button is grayed out.

5. Edit the following fields:

**Label** Type a label for the tool. This is the text that appears in the AEL.

**Mnemonic**

Select a key entry if you want users to be able to display this menu in the AEL using the keyboard shortcut of Alt and selected character.

**ShortCut**

Type a shortcut character if you want users to be able to display this menu in the AEL using the keyboard shortcut of Ctrl and the character provided here.

**Restriction:** The rename options are available only for tools.

6. Click **Save** to save your changes.

**Related tasks**

“Renaming tools” on page 270





---

## Chapter 5. Filtering event information

Network events typically create many alerts that are not of immediate importance to the personnel monitoring the system. Use filters and views to control the event information that is displayed to users.

---

### Filters

Filters constrain the rows returned by a data source by applying SQL correlation conditions to the field data in the data source. Filters can be applied to the following event displays in the Web GUI: the Active Event List (AEL), Lightweight Event List (LEL), Table View, and monitor boxes on an Event Dashboard. To create and edit filters, you use an HTML utility called the Filter Builder.

Filters can be created by Web GUI administrators who have the `ncw_admin` role, and by users who have the `ncw_user` role and the `netcool_rw` role.

**Important:** For each filter, you must select at least one data source. From this data source, the Filter Builder obtains the fields that you can use in the SQL query. If you select multiple data sources, the Filter Builder displays only the fields that are common to all those data sources. The data sources selected in the Filter Builder are not used in the AEL, Map Editor, or Event Dashboard to retrieve event data. You select the data sources for event data retrieval in the portlet preferences for the AEL, Map Editor, and Event Dashboard.

When a filter is applied to an event list, only the alerts that meet the criteria are displayed. A view may be assigned to a filter. If you do not assign a view to a filter, a default view is assigned. The following example shows a filter:

```
Node like '[a-zA-Z].*' and Severity > 3
```

This statement matches all alerts where the node starts with an alphabetic character and the severity of this data is greater than minor, that is, major or critical events.

### Filter categories

Filter categories control user access to filters, and are used for data migrated from IBM Tivoli Netcool/Webtop.

#### Global filters

Global filters are accessible to all users. As a read-write user, or a read-only user, you can copy global filters to your user profile, where you can modify the filters.

You can modify global filters and copy them to the user profiles of other users. You can also create new global filters.

#### User filters

User filters are stored in your user profile; only you and the administrator can access these profiles. In the Filter Builder, user filters are contained in a list called **My Filters**.

You can access the filters that all users have in their user profiles, and create filters in the profiles of users. In the **Available Filters** list, these filters are classified as **username Filters**. To assign filters to multiple users,

you can make copies of existing filters and assign them to a user group. The filters remain associated with the users in the group, not with the group itself.

### **System filters**

In the Filter Builder, access to system filters, and the ability to create, edit, and delete system filters is restricted to administrators. In the AEL, only administrators can select system filters from the **Filters** list. If the AEL was launched with a system filter already applied, non-administrators can view and select this system filter from the **Filters** list. If your Web GUI installation was upgraded or migrated from Netcool/Webtop, system filters are present after upgrade or migration. The system filters contain entity data migrated from Netcool/Webtop.

## **Filter collections**

Filter collections are logical groupings of filters. They are also used for migrated data from Netcool/Webtop. If your Web GUI installation was upgraded or migrated from Netcool/Webtop, filter collections contain the data migrated from entity groups.

Filter collections can contain only global filters or system filters. To edit a filter collection, your user must have the ncw\_admin role. You can create or delete filter collections only by using the Web GUI Administration Application Program Interface (WAAP) client.

## **Dependent filters**

Dependent filters concatenate the SQL statements from multiple filters by using the SQL OR operator. Dependent filters do not have their own SQL filter statements; they use only the statements from other filters.

Before you delete a filter, you must make sure that the filter is not used in any dependent filters. Otherwise, the dependent filter might return incorrect event data.

## **Transient filters**

Transient filters are filters that persist only for the duration of your current session.

### **Related concepts**

“Filter Builder overview” on page 286

“The Web GUI in a load balancing environment” on page 93

### **Related tasks**

“Modifying the preferences of a Web GUI user” on page 70

---

## Views

Views constrain the columns displayed in an Active Event List (AEL), Lightweight Event List (LEL), or Table View. You can control the order in which columns are displayed, lock columns in the display, and control the sorting of information in the columns.

When you create a view, you must select at least one data source. The data source is used to control the columns that can be included in the view (the columns represent the fields in the data source). If you select multiple data sources, you can select fields from all data sources.

### View categories

Views are either accessible to all users, or are assigned to a user profile. Access to views is controlled as follows:

#### Global views

These views are accessible to all users. As a read-write user, or a read-only user, you can copy global views to your user profile, where you can modify the views.

You can modify global views and copy them to the user profiles of other users. You can also create new global views.

#### User views

These views are stored in your user profile; only you and the administrator can access these profiles. User views are contained in a list called **My Views**.

You can access the views that all users have in their user profiles, and create views in the profiles of users. In the **Available Views** list, these views are classified as **username Views**. To assign views to multiple users, you can make copies of existing views and assign them to a user group. The views remain associated with the users in the group, not with the group itself.

### System views

In the View Builder, access to system views, and the ability to create, edit, and delete system views is restricted to administrators. In the AEL, only administrators can select system views from the **Views** list. If the AEL was launched with a system view already applied, non-administrators can view and select this system view from the **Views** list. If your Web GUI installation was upgraded or migrated from Netcool/Webtop, system views are present after upgrade or migration. The system views contain entity view data migrated from Netcool/Webtop.

#### Related concepts

“The Web GUI in a load balancing environment” on page 93

#### Related tasks

“Modifying the preferences of a Web GUI user” on page 70

---

## Setting up filters for event data


Use the Filter Builder to apply filters to an event list or Event Dashboard portlet. Administrators and read-write users can create and edit filters.

To open the Filter Builder, go to one of the following locations:

**AEL** Click **Edit > Filters**.

To open the default AEL applet, click **Availability > Events > Active Event List (AEL)**.

**Event Dashboard**

In either the tool bar or a monitor box, click **Edit Filters** . If you open the Filter Builder from a monitor box, the properties of the filter associated with the monitor box are loaded.

To open the default Event Dashboard portlet, click **Availability > Events > Event Dashboard**.

**Navigation**

Click **Administration > Event Management Tools > Filters**.

**Web page**

Use the following URL to add a link to the Filter Builder from a Web page:  
`protocol://server:port/ibm/console/webtop/startFB.do`

**Related concepts**

“The Web GUI in a load balancing environment” on page 93

“Event Dashboard overview” on page 305

## Filter Builder overview

The Filter Builder is an HTML utility that you use to construct filters that are dynamically applied to event data.

**Important:** For each filter, you must select at least one data source. From this data source, the Filter Builder obtains the fields that you can use in the SQL query. If you select multiple data sources, the Filter Builder displays only the fields that are common to all those data sources. The data sources selected in the Filter Builder are not used in the AEL, Map Editor, or Event Dashboard to retrieve event data. You select the data sources for event data retrieval in the portlet preferences for the AEL, Map Editor, and Event Dashboard.

You can use the following modes to create filters; the Filter Builder displays a tab for each mode.

**Basic** Provides a set of lists and text fields that you use to specify the filter conditions. To build the conditions, select a field from the specified data source or data sources, select a comparator, and type a numeric data type or string data type value. The data type value is used as the filtering criteria used against the field. If you use basic mode to construct your filter, you can view the resulting SQL in the text field on the **Advanced** tab.

**Advanced**

Provides a text field into which you can enter ObjectServer SQL syntax.

If you create a filter in advanced mode, it might not be possible to express the SQL syntax in the fields on the **Basic** tab. After you have saved a filter created in advanced mode, the **Basic** tab is removed for that filter.

For more information about ObjectServer SQL syntax, see the *IBM Tivoli Netcool/OMNIBus Administration Guide*.

#### **Dependent.**

This tab is displayed only for dependent filters. On this tab, use the **Search** fields to identify the filters that you want to use for the dependencies. After you have identified the required filters, use the buttons to move the filters from the **Available filters** list to the **Selected dependencies** list. In a dependent filter, the SQL WHERE statements of each filter are concatenated by using OR statements.

### **Filter Builder metrics**

A metric is an aggregate statistic that can be derived from the alerts that match a filter to display a useful figure, for example, an average, count, or sum of all field values. If a filter is displayed using a monitor box linked to an AEL, the metric information obtained from the set of alerts that match this filter is used for this display.

#### **Related concepts**

"Filters" on page 283

"Filter Builder overview" on page 286

## **Creating and editing filters**

Use the Filter Builder to create and edit filters for event data.

#### **Related concepts**

"The Web GUI in a load balancing environment" on page 93

### **Creating and editing filters in basic mode**

To create event filter conditions using lists that constrain your selection of values from the data source and the available comparators, use the Filter Builder in basic mode.

To create and edit filters in basic mode:

1. Open the Filter Builder.
2. To create a new filter, click **New Filter**.
3. To edit an existing filter.
  - a. Select the list that contains the required filter.
  - b. After the list has refreshed, click the filter.

If you are editing an existing filter, omit step 4.

4. Select the users you want to grant access to the filter and click **OK**.
  - **Public:** To make the filter accessible by all users, select **global**. To add a system filter, select **system**. All Web GUI users can select and copy global filters.
  - **Users:** Select the check boxes for the users who you want to grant access to this filter. If the check box corresponding to a user is selected, the selected filter is copied directly to the **My Filters** list for that user.

The list of users who can be granted access to the filter is based on the value of the **users.reload.mode** property in the *webgui\_home\_dir/etc/*

server.init file. If this property is set to 0, all system users are available. If set to 1, only users with either ncw\_user or ncw\_admin roles are available.

5. Specify the general properties for the filter:

#### Filter Name

Type a name for the filter. The filter name cannot contain spaces.

By default, the following characters cannot be used in filter names:

\$ ! % ^ & \* ( ) + = ~ ` ~ # @ ' : ; < > { } [ ] ? / \ \ | , "

These invalid characters are defined in the following file:

*webgui\_home\_dir/etc/illegalChar.prop.*

#### Default View

Select the view with which you want to associate the filter, or select the view that is associated with the filter. The default view is applied when you launch an AEL with the filter but do not specify a view. The default view is also applied when you launch an AEL from an Event Dashboard by clicking the monitor box that is associated with the filter.

#### Collection

For global filters and system filters only: Select the filter collection or collections to which you want to add the filter.

#### Description

Type a description that helps identify the purpose of the filter.

#### Data Source

Select the data source or data sources that contain the fields against which you want to run queries. Click **Show Data Sources** to display a list of available data sources.

**Important:** For each filter, you must select at least one data source. From this data source, the Filter Builder obtains the fields that you can use in the SQL query. If you select multiple data sources, the Filter Builder displays only the fields that are common to all those data sources. The data sources selected in the Filter Builder are not used in the AEL, Map Editor, or Event Dashboard to retrieve event data. You select the data sources for event data retrieval in the portlet preferences for the AEL, Map Editor, and Event Dashboard.

If you want to add a data source to an existing filter, make sure that the alerts.status table of the new data source contains all the fields that the filter specifies. If you add a data source that does not contain all the specified fields, the filter might return erroneous results.

The default data source corresponds to the default ObjectServer specified in the ncwDataSourceDefinitions.xml configuration file.

6. Click the **Basic** tab and, in the first row, create a filter condition as follows:
  - a. From the **Field** list, select a field from the specified data source.
  - b. From the **Comparator** list, select a comparator.
  - c. In the **Value field**, type a numeric data type value, or a string data type value. The data types must correspond to those in the ObjectServer field. String data type entries in the **Value** field must be contained in single quotes.
  - d. Optional: Use the LIKE and NOT LIKE comparators for regular expression pattern-matching metacharacters against the entry in the **Value** field.

**Restriction:** Do not use the `getdate` expression in the **Value** field. If you want to apply the `getdate` expression, use advanced mode instead.

7. To add additional filter conditions, click **New Condition**. You can add as many filter conditions as required.

8. Use the radio buttons under **Match** to specify how the filter conditions combine in aggregate:

- Select **And** to trigger the filter only if all the conditions are met.
- Select **Or** to trigger the filter if any of the conditions are met.

See “Sample SQL statements” for sample SQL statements that are generated based on the radio button selected.

9. Optional: To preview the literal SQL WHERE clause output, click **Advanced**.

10. Click **Metric** and use the following fields to set the metric value:

**Label** Type a title for the metric.

**Function**

Select a function to perform on the field data. The functions that can be performed on the field data are as follows:

- **Average:** Returns the average value of the selected field for all records that match the filter.
- **Count:** Returns a count of all the records that match the filter. The selected field is not used for this calculation.
- **Sum:** Returns the sum of the selected field for all records that match the filter.
- **Minimum:** Returns the lowest value of the selected field in records that match the filter.
- **Maximum:** Returns the highest value of the selected field in records that match the filter.

**Field** Select a field on which to perform the chosen function. The list contains all available fields in the `alerts.status` table of the data source.

11. Click **Save** to save the filter and continue working in the Filter Builder, or click **Save and Close** to save the filter and close the Filter Builder.

## Sample SQL statements

In step 8, when the **And** radio button is selected, the filter rows create the following SQL statement:

```
SELECT * from alerts.status where Node = 'node1' and Severity > 4 and
Summary like 'alert on .*'
```

When the **Or** radio button is selected, the filter rows create the following SQL statement

```
SELECT * from alerts.status where Node = 'node1' or Severity > 4 or
Summary like 'alert on .*'
```

## Related concepts

“Filter Builder overview” on page 286

## Related reference

“Filter Builder comparison operators” on page 294

“Pattern-matching meta characters” on page 295



## Creating and editing filters in advanced mode

Specify event data filtering conditions in ObjectServer SQL syntax.

For more information about ObjectServer SQL syntax, see the *IBM Tivoli Netcool/OMNIBus Administration Guide*.

To create or edit a filter using advanced mode:

1. Open the Filter Builder.
2. To create a new filter, click **New Filter**.
3. To edit an existing filter.
  - a. Select the list that contains the required filter.
  - b. After the list has refreshed, click the filter.

If you are editing an existing filter, omit step 4.

4. Select the users you want to grant access to the filter and click **OK**.
  - **Public:** To make the filter accessible by all users, select **global**. To add a system filter, select **system**. All Web GUI users can select and copy global filters.
  - **Users:** Select the check boxes for the users who you want to grant access to this filter. If the check box corresponding to a user is selected, the selected filter is copied directly to the **My Filters** list for that user.

The list of users who can be granted access to the filter is based on the value of the **users.reload.mode** property in the `webgui_home_dir/etc/server.init` file. If this property is set to 0, all system users are available. If set to 1, only users with either `ncw_user` or `ncw_admin` roles are available.

5. Use the following fields and buttons to set the general properties for the filter:

### Filter Name

Type a name for the filter. The filter name cannot contain spaces.

By default, the following characters cannot be used in filter names:

\$ ! % ^ & \* ( ) + = ~ ` ~ # @ ' : ; < > { } [ ] ? / \ \ | , "

These invalid characters are defined in the following file:

`webgui_home_dir/etc/illegalChar.prop`.

### Default View

Select the view with which you want to associate the filter, or select the view that is associated with the filter. The default view is applied when you launch an AEL with the filter but do not specify a view. The default view is also applied when you launch an AEL from an Event Dashboard by clicking the monitor box that is associated with the filter.

### Collection

For global filters and system filters only: Select the filter collection or collections to which you want to add the filter.

### Description

Type a description that helps identify the purpose of the filter.

### Data Source

Select the data source or data sources that contain the fields against which you want to run queries. Click **Show Data Sources** to display a list of available data sources.

**Important:** For each filter, you must select at least one data source. From this data source, the Filter Builder obtains the fields that you can



use in the SQL query. If you select multiple data sources, the Filter Builder displays only the fields that are common to all those data sources. The data sources selected in the Filter Builder are not used in the AEL, Map Editor, or Event Dashboard to retrieve event data. You select the data sources for event data retrieval in the portlet preferences for the AEL, Map Editor, and Event Dashboard.

If you want to add a data source to an existing filter, make sure that the alerts.status table of the new data source contains all the fields that the filter specifies. If you add a data source that does not contain all the specified fields, the filter might return erroneous results.

The default data source corresponds to the default ObjectServer specified in the ncwDataSourceDefinitions.xml configuration file.

6. Click **Advanced**.

7. In the **SQL Where clause** text field, enter any valid ObjectServer SQL WHERE clause.

For brevity, the SELECT part of the statement is omitted and you supply only the WHERE clause; the view associated with the filter determines the columns in the alerts.status table that are selected..

8. Click **Metric** and use the following fields to set the metric value:

**Label** Type a title for the metric.

**Function**

Select a function to perform on the field data. The functions that can be performed on the field data are as follows:

- **Average:** Returns the average value of the selected field for all records that match the filter.
- **Count:** Returns a count of all the records that match the filter. The selected field is not used for this calculation.
- **Sum:** Returns the sum of the selected field for all records that match the filter.
- **Minimum:** Returns the lowest value of the selected field in records that match the filter.
- **Maximum:** Returns the highest value of the selected field in records that match the filter.

**Field** Select a field on which to perform the chosen function. The list contains all available fields in the alerts.status table of the data source.

9. Click **Save** to save the filter and continue working in the Filter Builder, or click **Save and Close** to save the filter and close the Filter Builder.

**Related concepts**

"Filter Builder overview" on page 286

**Related reference**

"Filter Builder comparison operators" on page 294

"Pattern-matching meta characters" on page 295

## Creating and editing dependent filters

To aggregate the SQL WHERE clauses of multiple filters by using the OR operator, create a dependent filter.

**Tip:** All the filters that you use in a dependent filter must contain an SQL WHERE statement.

To create or edit a dependent filter:

1. Open the Filter Builder.
2. Create or edit a filter:
  - To create a new filter, click **New Filter**.
  - To edit an existing filter select the list that contains the required filter, and then select the filter from the refreshed list. You can omit step 3.
3. Select the users you want to grant access to the filter and click **OK**.
  - **Public:** To make the filter accessible by all users, select **global**. To add a system filter, select **system**. All Web GUI users can select and copy global filters.
  - **Users:** Select the check boxes for the users who you want to grant access to this filter. If the check box corresponding to a user is selected, the selected filter is copied directly to the **My Filters** list for that user.

The list of users who can be granted access to the filter is based on the value of the **users.reload.mode** property in the *webgui\_home\_dir/etc/server.init* file. If this property is set to 0, all system users are available. If set to 1, only users with either ncw\_user or ncw\_admin roles are available.
4. Use the following fields to set the general properties for the filter:

### Filter Name

Type a name for the filter. The filter name cannot contain spaces.

By default, the following characters cannot be used in filter names:

\$ ! % ^ & \* ( ) + = ~ ` ~ # @ ' : ; < > { } [ ] ? / \ | , "

These invalid characters are defined in the following file:

*webgui\_home\_dir/etc/illegalChar.prop*.

### Default View

Select the view with which you want to associate the filter, or select the view that is associated with the filter. The default view is applied when you launch an AEL with the filter but do not specify a view. The default view is also applied when you launch an AEL from an Event Dashboard by clicking the monitor box that is associated with the filter.

### Collection

For global filters and system filters only: Select the filter collection or collections to which you want to add the filter.

### Description

Type a description that helps identify the purpose of the filter.

### Data Source

Select the data source or data sources that contain the fields against which you want to run queries. Click **Show Data Sources** to display a list of available data sources.

**Important:** For each filter, you must select at least one data source. From this data source, the Filter Builder obtains the fields that you can use in the SQL query. If you select multiple data sources, the Filter

Builder displays only the fields that are common to all those data sources. The data sources selected in the Filter Builder are not used in the AEL, Map Editor, or Event Dashboard to retrieve event data. You select the data sources for event data retrieval in the portlet preferences for the AEL, Map Editor, and Event Dashboard.

If you want to add a data source to an existing filter, make sure that the alerts.status table of the new data source contains all the fields that the filter specifies. If you add a data source that does not contain all the specified fields, the filter might return erroneous results.

The default data source corresponds to the default ObjectServer specified in the ncwDataSourceDefinitions.xml configuration file.

5. On the **Dependent** tab, use the **Search** fields to identify the filters that you want to use for the dependencies.

**Tip:** To search through large numbers of filters, select the filter category, and type the filter name in the **Search** field. If a matching filter is found in the selected filter category, the filter is selected in the **Available filters** list.

6. Add filters from the **Available filters** list to the **Selected dependencies** list as required.
7. To set a metric for the filter, click **Metric** and use the following fields to specify the metric value:

**Label** Type a title for the metric.

**Function**

Select a function to perform on the field data. The functions that can be performed on the field data are as follows:

- **Average:** Returns the average value of the selected field for all records that match the filter.
- **Count:** Returns a count of all the records that match the filter. The selected field is not used for this calculation.
- **Sum:** Returns the sum of the selected field for all records that match the filter.
- **Minimum:** Returns the lowest value of the selected field in records that match the filter.
- **Maximum:** Returns the highest value of the selected field in records that match the filter.

**Field** Select a field on which to perform the chosen function. The list contains all available fields in the alerts.status table of the data source.

8. Click **Save** to save the filter and continue working in the Filter Builder, or click **Save and Close** to save the filter and close the Filter Builder.

## Editing filter collections

You can add filters to filter collections and remove filters.

### Before you begin

To edit filter collections, your user must have the ncw\_admin role.

You can add only global and system filters to a filter collection.

You can use the Filter Builder only to add filters to collections or remove filters from collections. To create or delete filters collections, or to modify collections (for

example by changing the name of a filter collection), you must use the WAAPI client. For sample WAAPI commands for filter collections, see the following file: `webgui_home_dir/waapi/etc/samples/samplerrequest_filtercollection.xml`

To edit filter collections:

- To add a filter to a collection:
  1. Open the Filter Builder.
  2. From the **Available Filters** list, select **Category**.
  3. Select the collection that contains the required filter, and then select the filter. The properties of the selected filter are loaded.
  4. In the **Collections** list, select the filter collection or collections to which you want to add the filter.
  5. Click **Save and Close** to save the assignment of the filter to the collection or collections, and close the Filter Builder.
- To remove a filter from a collection:
  1. Open the Filter Builder.
  2. To identify which filters are assigned to a collection, select **Collection** from the **Available Filters** list and then select the required collection. The list of filters assigned to the collection is sorted into global filters and system filters.
  3. Select the filter that you want to remove from the collection. The properties of the selected filter are loaded. In the **Collections** list, the collections to which the filter is assigned are highlighted.
  4. Press CTRL and click the filter collection from which you want to remove the filter.
  5. Click **Save and Close** to save the assignment of the filter to the collection or collections, and close the Filter Builder.

#### Related tasks

“Remotely administering the Web GUI server” on page 100

### Filter Builder comparison operators

When you use the Filter Builder in basic mode, the **Comparator** field contains these comparison operators.

The operators are described in the following table.

*Table 65. Comparison operators in the basic mode of the Filter Builder*

| Operator | Description                                                                                                                                                                                                                                                     |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| =        | Tests for equality.                                                                                                                                                                                                                                             |
| <>       | Tests for inequality.                                                                                                                                                                                                                                           |
| !=       |                                                                                                                                                                                                                                                                 |
| <        | Tests for greater than (>), less than (<), greater than or equal to (>=) or less than or equal to (<=). These operators perform case-sensitive string comparisons. In standard ASCII case-sensitive comparisons, upper case letters precede lower case letters. |
| >        |                                                                                                                                                                                                                                                                 |
| <=       |                                                                                                                                                                                                                                                                 |
| >=       |                                                                                                                                                                                                                                                                 |

Table 65. Comparison operators in the basic mode of the Filter Builder (continued)

| Operator   | Description                                                                                                                                                                                                                                                                                                                                   |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [not] like | The like operator performs string comparisons. The string following the like operator, which can be the result of a regular expression, is the pattern to which the column expression is compared. A regular expression can include regular expression pattern-matching metacharacters. The not keyword inverts the result of the comparison. |

### Related tasks

“Creating and editing filters in basic mode” on page 287

“Creating and editing filters in advanced mode” on page 290

## Pattern-matching meta characters

Regular expressions are made up of normal characters and metacharacters. Normal characters include upper and lower case letters and numbers

Regular expression pattern-matching can be performed with either a single character or a pattern of one or more characters in parentheses, called a character pattern. Metacharacters have special meanings, as described in the following table.

Table 66. Pattern-matching metacharacters

| Pattern-matching metacharacter | Description                                                                                                                              | Example                                                                                   |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| *                              | This metacharacter matches zero or more instances of the preceding character or character pattern.                                       | The pattern goo* matches my godness, my goodness, and my goodness, but not my gdness.     |
| +                              | This metacharacter matches one or more instances of the preceding character or character pattern.                                        | The pattern goo+ matches my goodness and my goodness, but not my godness.                 |
| ?                              | This metacharacter matches zero or one instance of the preceding character or character pattern.                                         | The pattern goo? matches my godness and my goodness, but not my goodness or my gdness.    |
| \$                             | This metacharacter matches the end of the string.                                                                                        | The pattern end\$ matches the end, but not the ending.                                    |
| ^                              | This metacharacter matches the beginning of the string.                                                                                  | The pattern ^severity matches severity level 5, but not The severity is 5.                |
| .                              | This metacharacter matches any single character.                                                                                         | The pattern b.at matches baat, bBat, and b4at, but not bat or bB4at.                      |
| [abcd]                         | This metacharacter matches any characters in the square brackets or in the range of characters separated by a hyphen (-), such as [0-9]. | ^[A-Za-z]+\$ matches any string that contains only upper or lower case letter characters. |

Table 66. Pattern-matching metacharacters (continued)

| Pattern-matching metacharacter | Description                                                                                                                                                                                                                  | Example                                                                                                                                                       |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [^abcd]                        | Matches any character except those in the square brackets or in the range of characters separated by a hyphen (-), such as [0-9].                                                                                            | [^0-9] matches any string that does not contain any numeric characters.                                                                                       |
| ()                             | This metacharacter indicates that the characters in the parenthesis should be treated as a character pattern.                                                                                                                | A(boo)+Z matches AbooZ, AboobooZ, and AboobooZ, but not AboZ or AboooZ.                                                                                       |
|                                | This metacharacter matches one of the characters or character patterns on either side of the vertical bar.                                                                                                                   | A(B C)D matches ABD and ACD, but not AD, ABCD, ABBD, or ACCD.                                                                                                 |
| \                              | The backslash escape character indicates that the metacharacter following should be treated as a regular character. The metacharacters in this table require a backslash before them if they appear in a regular expression. | To match an opening square bracket, followed by any digits or spaces, followed by a closed bracket, use the following regular expression:<br><br>\[[0-9 ]*\]. |

## Copying a filter from another filter group

Read-write users can copy global filters and system filters to the **My Filters** group. Administrators can copy filters from any location to any destination.

You can deploy filters directly to individual read-write users, or copy filters to and from the public **Global Filters** group, or the **My Filters** group. In addition, filters can be allocated to user groups.

To copy filters from the **Global** filter group to the **My Filters** group:

1. Open the Filter Builder.
2. Choose **Global Filters** from the **Available Filters** list.
3. From the filter list, select the filter you want to copy. The page updates with the filter properties.
4. Click **Copy Filter**.
5. Select the following check boxes to set access for the new filter:

**Public** To make the filter accessible by all users, select **global**. To add a system filter, select **system**. All Web GUI users can select and copy global filters.

**Users** Select the check boxes for the users who you want to grant access to this filter. If the check box corresponding to a user is selected, the selected filter is copied directly to the **My Filters** list for that user.

The list of users who can be granted access to the filter is based on the value of the **users.reload.mode** property in the *webgui\_home\_dir/etc/server.init* file. If this property is set to 0, all system users are available. If set to 1, only users with either *ncw\_user* or *ncw\_admin* roles are available.

### Groups

For copying filters only: Select the check boxes for the groups to which want to grant access to this filter. If the check box corresponding to a group is selected, the selected filter is assigned to the user profiles of all users who are a member of that group. The users can access the filter under **My Filters**.

6. Click **OK**. The page updates and you are presented with a copy of the chosen filter.
7. Make any modifications necessary to the filter configuration, and click **Save**. The filter is saved to the **My Filters** group, and is now available for use in the **Filters** list of the AEL.

### Results

For example, to copy a filter from the administrator's personal filter group to the public group:

1. Choose **My Filters** from the **Available Filters** list.
2. From the filter list, select the filter you want to copy. The page updates with the filter properties.
3. Click **Copy Filter**. The next page contains a grouped list of the possible filter recipients.
4. Select the **global** check box and click **OK**. The page updates and you are presented with a copy of the chosen filter.
5. Make any modifications necessary to the filter configuration, and click **Save**. The filter is saved to the **global** group. Users can now copy the filter from this group to their personal **My Filters** group.

## Deleting filters

Delete filters that are no longer required.

### Before you begin

Remove the filter that you want to delete from any dependent filters. If you delete a filter that is still specified in a dependent filter, the dependent filter might return incorrect or misleading event information.

As a read-write user, you can delete only filters in the **My Filters** list. These filters are assigned to your user.

You can delete filters assigned to your user, filters that are assigned to other users, and global and system filters.

1. Open the Filter Builder.
2. From the **Available Filters** list, select the group that contains the filter to be deleted.
3. Select the filter and click **Delete Filters**.

---

## Setting up views for event lists

Use the View Builder, which is an HTML utility, to build views for applying to an Active Event List (AEL). Administrators and read-write users can create and edit views.

Open the View Builder from the following locations:

**AEL** Click **Edit > Views**.

### Navigation

Click **Administration > Event Management Tools > Views**.

### Related concepts

"The Web GUI in a load balancing environment" on page 93

## Creating views

Create views that are dynamically applied to Active Event List data. The views determine what information is displayed from the available event data.

To create a view:

1. Open the View Builder.
2. Click **New View**.
3. Select the users you want to grant access to the view being created and click **OK**.

**Public** To make the view accessible by all users, select **global**. To add a system view, select **system**. All Web GUI users can select and copy global views.

**Users** Select the users whom you want to grant access to this view. The view is copied to the **My Views** categories of all selected users.

The list of users who can be granted access to the view is based on the setting of the **users.reload.mode** property in the *webgui\_home\_dir/etc/server.init* file. If set to 0, all users are available. If set to 1, only users with either *ncw\_user* or *ncw\_admin* roles are available.

4. Use the following fields and buttons to set the general properties for the view:

**Name** Type a name for the view. The view name may contain no spaces.

By default, the following characters may not be used in view names: \$ ! £ % ^ & \* ( ) + = ~ ` ~ # @ ' : ; < > { } [ ] ? / \ \ | , "

By default, the following characters may not be used as the initial character of view names: / \ \ \* ? " < > | & .

These invalid characters are defined in the following file:  
*webgui\_home\_dir/etc/illegalChar.prop*

### Data Source

Select the data source or data sources that contain the fields that you want to be displayed in the view. Click **Show Data Sources** to display a list of available data sources.

**Important:** For each view you must select at least one data source. From this data source, the View Builder obtains the fields that you can use in the view. If you select multiple data sources, the View Builder displays a union of all the fields contained in all data sources. The data sources selected in the View Builder are not used in the AEL to retrieve



event data. You select the data source or data sources from which event data is retrieved in the portlet preferences for the AEL. If a view specifies fields that do not exist in a data source, when the view and the data source are applied to an AEL, the columns that represent the non-existent fields are blank.

If you want to add a data source to an existing view, make sure that the `alerts.status` table of the new data source contains all the fields that the views specifies.

The default data source corresponds to the default `ObjectServer` specified in the `ncwDataSourceDefinitions.xml` configuration file.

5. Click **Save**.

## What to do next

You can now configure the view properties, or click **Save and Close** to save the view and close the View Builder.

### Related tasks

“Adding and removing columns”

“Configuring column titles and width” on page 300

“Changing the column order” on page 300

“Changing the sorting precedence” on page 302

“Changing the sort order in a column” on page 303

## Adding and removing columns

Set what event information is displayed from the available event data. The selected event data is added in columns to the AEL.

To add new event data to, or remove event data from the view:

1. Open the View Builder.
2. Select a view from the **Available Views** list. The page updates with the view properties.
3. Use the following fields and buttons to add or remove event data from the view:

### Available fields

Lists all available event data fields. The fields available are derived from the `alerts.status` table in the `ObjectServer`. If you selected multiple `ObjectServers` from the **Data Source** list, all fields from all `ObjectServers` are displayed. Select the field of the event data you want to appear as a new column in the event list. Click **Add selected field** to add the selected event data field to the event list as a column. Click **Add all fields** to add all fields to the event list as columns.

### Event list view

Lists all the fields selected and displayed as columns in the event list. Select the fields you want to remove from the event list. Click **Remove selected fields** to remove the selected event data field from the event list columns. Click **Remove all fields** to remove all fields from the event list columns. Use the arrow buttons to the right of this list to change the display order of the columns in the event list.

### Lock column

Locks the selected column at the far left of the event list in the

view, so that the column is always displayed when you scroll horizontally. Use the arrow buttons to the right of the list to change the display order of the locked columns in the event list. You cannot change the display order to mix locked columns and unlocked columns.

4. Click **Save** to save the view and continue customizing the view, or click **Save and Close** to save the view and close the View Builder.

If you click **Save** wait for your changes to be applied to the AEL before continuing the customize the view.

## Configuring column titles and width

Customize the columns that display event data.

To configure a column used in the view:

1. Open the View Builder.
2. Select a view from the **Available Views** list.
3. In the **Event list view** list, select a field.  
The **Column title** field updates with the corresponding column title for the selected field.
4. In the **Column title** field, type a new title for the column or keep the default.
5. Select the position of the title text in the column heading. Select an entry from the **Justify title** drop-down menu to the right of the **Column title** field.
6. Select the position of the event data in the columns. Select an entry from the **Justify data** drop-down menu to the right of the **Column width** field.
7. To change the width of a column, type a numeric value for the column width in the **Column width** field.
8. Click **Save** to save the view and continue customizing the view, or click **Save and Close** to save the view and close the View Builder.

If you click **Save** wait for your changes to be applied to the AEL before continuing the customize the view.

## Results

After a field is moved between the **Available fields** list and the **Event list view** list, the field retains any configurations you apply.





## Changing the column order

Set the horizontal left-to-right order of the event data columns in the view.

To change the order of columns within the view:

1. Open the View Builder.
2. Select a view from the **Available views** list. The page updates with the view properties.
3. In the **Event list view** list, select a field.
4. Use the arrow buttons to the right of the list to change the display order of the columns in the view. The buttons are described in the following table.

Table 67. View editor column order buttons

| Button                                                                            | Action                                                                                                                                                                                                    |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Click <b>Top</b> to move the field to the top of the list. In the view, the field is displayed as the column furthest to the left.                                                                        |
|  | Click <b>Up</b> to move the selected field up one position in the list. In the view, the field is displayed as the column to the left of the one listed below it in the <b>Event List View</b> list.      |
|  | Click <b>Down</b> to move the selected field down one position in the list. In the view, the field is displayed as the column to the right of the one listed above it in the <b>Event List View</b> list. |
|  | Click <b>Bottom</b> to move the selected field to the bottom of the list. In the view, the field is displayed as the column furthest to the right in the view.                                            |

Locked columns are displayed at the top of the **Event list view** list. Unlocked columns are always displayed beneath locked columns. You cannot change the display order to mix locked columns and unlocked columns.

- Click **Save** to save the view and continue customizing the view, or click **Save and Close** to save the view and close the View Builder.

If you click **Save** wait for your changes to be applied to the AEL before continuing the customize the view.

#### Related tasks

“Locking columns”





## Locking columns

Lock columns that are important to you, so that the columns are always displayed when you scroll horizontally Active Event List (AEL) to which the view is applied.

Locked columns are displayed at the left side of the view.

To lock columns:

- Open the View Builder.
- Select a view from the **Available views** list. The page updates with the view properties.
- In the **Event list view** area, select the field that represents the required column and select the **Lock column** check box. The field moves to the top of the list and the name changes to *field\_name* [locked].
- Lock all of the required columns.
- Optional: If you have more than one locked column, change the order in which the columns are displayed in the AEL. The top locked column is displayed as the leftmost column of the view, and the bottom locked column is displayed to the left of the first unlocked column.

| Button                                                                              | Action                                                                 |
|-------------------------------------------------------------------------------------|------------------------------------------------------------------------|
|  | Moves the selected column to the top of the list of locked columns.    |
|  | Moves the selected column up the list of locked columns.               |
|  | Moves the selected column down the list of locked columns.             |
|  | Moves the selected column to the bottom of the list of locked columns. |

6. Click **Save** to save the view and continue customizing the view, or click **Save and Close** to save the view and close the View Builder.

If you click **Save** wait for your changes to be applied to the AEL before continuing the customize the view.

## Changing the sorting precedence

Set the precedence of sorting principles.

The field at the top of the **Sorted By** list has the highest precedence when sorting. If you add a second field to the list, and the first field contains a number of identical entries, the second field is used to sort within those entries. To change the sort order of columns in the view:

1. Open the View Builder.
2. Select a view from the **Available views** list. The page updates with the view properties.
3. Use the following fields and buttons to add or remove fields to be used in setting the sorting precedence:

### Available sort fields





Lists all available event data fields that can be used to arrange which data has a priority when the events are displayed in the event list. The fields available are derived from the alerts.status table in the ObjectServer. Select the field of the event data you want to display first in the event list. Click **Add selected field** to add the selected event data field to the event list as a priority. Click **Add all fields** to add all fields to the event list. Then you can set the sorting precedence in the **Sorted by** list.

### Sorted by

Lists all the fields selected to be part of the sorting hierarchy. The field at the top of the list has the highest precedence when sorting. If you add a second field to the list, and the first field contains a number of identical entries, the second field is used to sort within those entries. Click **Remove selected fields** to remove the selected event data field from the sorting hierarchy. Click **Remove all fields** to remove all fields from the sorting hierarchy. Use the arrow buttons to the right of this list to change the sorting hierarchy. The fields at the top of the list have a higher priority when event data is displayed in the event list.

With an entry in the **Sorted by** list highlighted, click one of the direction buttons to the right of the list. The buttons are summarized in the following table.

Table 68. View editor sort order buttons

| Button                                                                              | Action                                                                    |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------|
|  | Click <b>Top</b> to move the field to the top of the list.                |
|  | Click <b>Up</b> to move the selected field up one position.               |
|  | Click <b>Down</b> to move the selected field down one position.           |
|  | Click <b>Bottom</b> to move the selected field to the bottom of the list. |

4. Click **Save** to save the view and continue customizing the view, or click **Save and Close** to save the view and close the View Builder.

If you click **Save** wait for your changes to be applied to the AEL before continuing the customize the view.

## Changing the sort order in a column

Set the sorting order of the event data within a column to ascending or descending.

In the **Sorted by** list, a suffix is placed next to each entry to indicate the sort order of data within that field. By default, this entry is [desc], so the field is sorted in descending alphabetical or alphanumerical order. The suffix [asc] indicates that the field is sorted in ascending alphabetical, or alphanumerical order.

To change the direction of sorting within a field:

1. Open the View Builder.
2. Select a view from the **Available views** list. The page updates with the view properties.
3. Select a field in the **Sorted by** list within the **Sort Columns** area.
4. Set the sort direction by selecting **Ascending** or **Descending** from the **Sort** list.
5. Click **Save** to save the view and continue customizing the view, or click **Save and Close** to save the view and close the View Builder.

If you click **Save** wait for your changes to be applied to the AEL before continuing the customize the view.

## Copying global views

As a read write user or an administrator, you can copy global views and system views to the **My Views** group.

To copy views from the **Global** view list to the **My Views** group:

1. Open the View Builder.
2. From the **Available Views** list, select either **Global Views** or **System Views**.
3. From the view list, select the view you want to copy.

The page updates with the view properties.

4. Click **Copy View**.
5. From the **Users** list, select your user name.
6. Click **OK**.

The copy of the view is displayed. By default, the view name is set to "CopyOfviewname."

7. Edit the view as required.
8. Click **Save** to save the view and continue customizing the view, or click **Save and Close** to save the view and close the View Builder.

If you click **Save** wait for your changes to be applied to the AEL before continuing the customize the view.

## Results

The view is saved in your user profile. In the Active Event List (AEL), when you select the view from the **Views** list, the view is applied to the AEL.



---

## Chapter 6. Visualizing high-level event information

In environments that deal with very high numbers of alerts, use the event visualization components to give you an overview of the available data.

While the Active Event List, Lightweight Event List, and the Table View give detailed and interactive representations of alert status within a network, use the following event visualization components for a comparative and high-level view of the alerts:

**Charts** Use charts to represent event information against scales that indicate the values of the information.

### Event Dashboards

Use Event Dashboards to represent multiple SQL queries against the alerts tables of the ObjectServer.

### Gauges

Use gauges to display the values of metrics.

**Maps** Use maps to obtain an interactive representation of a network.

---

## Visualizing event information on Event Dashboards

Use the Event Dashboard portlet to maintain a high-level overview of events that match SQL queries. From the Event Dashboard dashboard, you can open an Active Event List (AEL) to investigate events in depth.

To open the default Event Dashboard, click **Availability > Events > Event Dashboard**.

### Event Dashboard overview


Use this window to view one or more categories of alert information. Each alert category is depicted by a *monitor box*, which represents a filter.

#### About monitor boxes

A monitor box contains the following buttons and fields:

##### Filter name

Displays the name of the filter that is associated with this monitor box.

To edit the filter, click **Edit Filters**  next to the filter name. The Filter Builder is opened, and the data and SQL query associated with the filter are loaded.

**Total** This value represents the total number of events that match the filter.

##### Highest

This value represents the highest severity among the alerts that match the filter.

##### Lowest

This value represents the lowest severity among the alerts that match the filter.

**Metric** This value represents a calculation that is applied to the alerts that match the filter. This derived value shows the average, sum, lowest, or highest value of a selected column in the alerts database table that is being queried by the filter.

#### **Distribution meter**

This area shows the severity distribution of the alerts that match the filter. The colors used in the bars are preconfigured to identify the different severity levels.

In the portlet preferences, you can configure the Event Dashboard to perform actions when you click the distribution meter. In the Preferences window, you can specify how the distribution meter is displayed: as a histogram or in a lava lamp format. You can also switch off the distribution meter.

You must make sure that the data sources specified in the filter and the data sources selected in the Event Dashboard contain identical fields; if this is not the case, an error message is displayed in the affected monitor boxes instead of event data.

#### **Related concepts**

"The Web GUI in a load balancing environment" on page 93

#### **Related tasks**

"Setting up filters for event data" on page 286

"Setting Event Dashboard portlet preferences and defaults" on page 228

"Adding monitor boxes to Event Dashboard portlets"

"Deleting monitor boxes from Event Dashboards" on page 307

"Customizing monitor box information" on page 308

"Changing the event information displayed on monitor boxes" on page 308

"Customizing the monitor boxes on Event Dashboards" on page 309

"Freezing and unfreezing Event Dashboards" on page 310

"Monitoring events in the AEL" on page 371


## **Adding monitor boxes to Event Dashboard portlets**

To add a new monitor box to an Event Dashboard portlet, use the Filter Builder to add a new filter.

As a read-write user you can add filters to the My Filters list. The filters in this list are assigned to your user profile.

You can add filters to the My Filters list, and to the user profiles of read-write users. You can also add global filters.

To add monitor boxes to an Event Dashboard portlet:

1. Open an Event Dashboard portlet.
2. To open the Filter Builder, click **Edit Filters** .
3. From the **Available Filters** list, select **My Filters**.
4. Select the level of access for the filter:
  - To make the filter available to all users, select **Global Filters**.
  - To assign the filter to a specific user, select the required user name.
5. Click **Add Filter**.



6. Use the Filter Builder to create the required filter.
7. Click **Save** to save the filter, or click **Save and Close** to save the filter and close the Filter Builder.

## Results

A new monitor box that displays the events matching the filter is added to the Event Dashboard portlet.

### Related concepts

“Event Dashboard overview” on page 305

“Filter Builder overview” on page 286

“The Web GUI in a load balancing environment” on page 93

### Related tasks

“Creating and editing filters in basic mode” on page 287

“Creating and editing filters in advanced mode” on page 290

“Creating and editing dependent filters” on page 292

## Deleting monitor boxes from Event Dashboards


To delete a monitor box from an Event Dashboard portlet, delete the filter that corresponds to the monitor box.

### Before you begin

As a read-write user, you can delete filters that are assigned to your user. In the Filter Builder, these filters are contained in the **My Filters** list.

You can delete filters assigned to your user, filters that are assigned to other users, and global and system filters.

To delete monitor boxes from an Event Dashboard portlet:

1. Open an Event Dashboard portlet and note the name of the filter that you want to delete.
2. To open the Filter Builder, on the tool bar, click **Edit Filters** .
3. In the Filter Builder, from the **Available Filters** list, select the list that contains the required filter:
  - To delete a filter associated with your user, select **My Filters**.
  - Select **Global Filters**, **System Filters**, or the user name associated with the required filter.
4. From the list, select the filter that you want to delete and click **Delete**.
5. Click **OK**.

## Results

The monitor box that corresponded to the deleted filter is no longer displayed on the Event Dashboard portlet.

If you delete a global filter, the monitor box is immediately removed from the Event Dashboard.

### Related concepts

“Event Dashboard overview” on page 305


“The Web GUI in a load balancing environment” on page 93

## Customizing monitor box information

To change the information that is displayed on a monitor box on an Event Dashboard portlet, edit the filter that controls the monitor box.

As a read-write user, you can edit only filters that are assigned to your user. In the Filter Builder, these filters are contained in the **My Filters** list.

You can delete filters assigned to your user, filters that are assigned to other users, and global and system filters.

1. On an Event Dashboard portlet, next to the filter name of the required monitor box, click **Edit Filters** .

The Filter Builder opens, and the selected filter is loaded.

2. Edit the general properties of the filter.
3. Edit the SQL query of the filter by using either Basic mode or Advanced mode.
4. If the filter is a dependent filter, edit the dependencies.
5. To save your changes, click **Save and Close**.

### Related concepts

“Event Dashboard overview” on page 305

“Filter Builder overview” on page 286

“The Web GUI in a load balancing environment” on page 93

### Related tasks

“Creating and editing filters in basic mode” on page 287


“Creating and editing filters in advanced mode” on page 290

“Creating and editing dependent filters” on page 292

## Changing the event information displayed on monitor boxes

To specify the type of information displayed in the monitor boxes, and the format of that information, use the Event Dashboard preferences.

To change the information on the monitor boxes:

1. On an Event Dashboard portlet, click **Edit Preferences** .
2. On the Preferences window, click **Monitor Boxes** and complete the fields as follows:

#### Show Number of Alerts

Displays the number of alerts that match the filter.

#### Show Highest Severity

Displays the highest severity of the alerts that match the filter.

#### Show Lowest Severity

Displays the lowest severity of the alerts that match the filter.

#### Show Highest Severity Border

Displays a border around the monitor box in the color of the highest-severity alert that matches the filter.

**Show Metric**

Displays the selected filter metric value.

**Show Highest Color**

Applicable only if you selected the **Show Highest Severity** option:

Displays the highest-severity alert indicator in the color of the alert, for example, in red if the highest-severity alert is critical.

**Show Lowest Color**

Applicable only if you selected the **Show Lowest Severity** option:

Displays the lowest-severity alert indicator in the color of the alert.

**Font** Select the font and the font size for the text on the monitor boxes.

**Distribution meter**

Specify the format for the distribution meter:

- **Show Lava Lamp:** Displays the distribution meter as a series of horizontal bars.
- **Show Histogram:** Displays the distribution meter as a bar graph.
- **Show None:** Switches off the distribution meter.

3. Save the settings for use in the current session, or for future sessions:

- To use these preferences in the current session only, click **Apply**.
- To use these preferences in future sessions, click **Save**.

4. Optional: If the Event Dashboard is configured to open an Active Event List (AEL), click the other tabs and change the AEL settings.

5. To exit the Preferences window, click **Close**.

**Related concepts**

“Event Dashboard overview” on page 305

“The Web GUI in a load balancing environment” on page 93

**Related tasks**

“Changing the AEL refresh rate” on page 241

“Enabling notifications and setting notification criteria” on page 244

“Enabling flashing and setting flash speed and brightness” on page 245

“Modifying the AEL font and window settings” on page 245

“Modifying the AEL date and time format” on page 246

“Changing how event severity is depicted in the AEL” on page 248

“Customizing the monitor boxes on Event Dashboards”


## Customizing the monitor boxes on Event Dashboards


Use the portlet preferences of the Event Dashboard portlet to control how the monitor boxes are arranged, and which monitor boxes are displayed.

### Before you begin

To change the portlet preferences of an Event Dashboard portlet, your user must be assigned the `ncw_dashboard_editor` role.

You control the layout of the monitor boxes in the **Dashboard Layout** area of the Edit Event Dashboard Portlet Preferences window.


1. On an Event Dashboard portlet, click **Edit** .
2. To restore the default settings for the portlet, as specified by the administrator, click **Reset to Defaults**.

3. To remove monitor boxes from the layout, use the following options:
  - To remove a single monitor box, on the monitor box, click **Remove Monitor Box** .

- To remove all monitor boxes, click **Remove All Monitor Boxes** .


4. To add a new filter, and a new monitor box, click **Edit Filters**  and use the Filter Builder to specify the properties of the filter and the SQL query.

**Tip:** If you want all new filters to be automatically added as monitor boxes to the Event Dashboard, select **Show All Filters**.

5. To restore a previously-removed monitor box, or add a new filter as a monitor box, click **Add Monitor Box**  and, from the list, select the required monitor box. The monitor boxes are listed by filter name.
6. To show all filters on the layout (that is, all global filters and all filters assigned to your user in the **My Filters** list) select **Show All Filters**.

**Tip:** To display only a subset of the available filters:

- a. Select **Show All Filters** to display all the available filters.
- b. Clear **Show All Filters**.
- c. Remove the monitor boxes that you do not require.

7. To change the appearance of the Event Dashboard, click **Edit Preferences**  and change the settings under **Monitor Boxes**.
8. To change the arrangement of the monitor boxes on the Event Dashboard, proceed as follows:
  - a. To change the number of columns in which the monitor boxes are arranged, in the **Columns** list, select or type the required number.
  - b. Drag the monitor boxes into the required arrangement.
9. To save and apply your settings to the portlet, click **OK**.

#### Related concepts

"Event Dashboard overview" on page 305

"Filter Builder overview" on page 286

#### Related tasks



"Changing the event information displayed on monitor boxes" on page 308

"Creating and editing filters in basic mode" on page 287

"Creating and editing filters in advanced mode" on page 290

## Freezing and unfreezing Event Dashboards

To take a snapshot of the alert information before it is changed by updates from the ObjectServer, you can freeze all the monitor boxes on the Event Dashboard.

- To freeze the monitor boxes, click **Freeze/Unfreeze** .  
The updates from the ObjectServer are suppressed. On the status bar of the portlet, the **Auto refresh in** countdown is paused.
- To unfreeze the monitor boxes and obtain the updates from the ObjectServer, click **Freeze/Unfreeze** .  
On the status bar of the portlet, the **Auto refresh in** countdown resumes.

- Optional: To force a refresh of the monitor boxes independently of the refresh rate, click **Refresh** .

#### Related concepts


“Event Dashboard overview” on page 305

## Customizing Active Event List actions on Event Dashboards

To customize the actions that you can execute from an Event Dashboard portlet, and from an Active Event List (AEL) that is opened from the Event Dashboard, use the portlet preferences.

### Before you begin

To change the portlet preferences of an Event Dashboard portlet, your user must be assigned the `ncw_dashboard_editor` role.

1. On an Event Dashboard portlet, click **Edit** .
2. In the **Single-click** list, select the action that you want to be performed when you click the distribution meter.
  - **Update AEL on Same Page:** Sets the filter and view for any AEL applet that is displayed on the same page as an Event Dashboard.
  - **Open New AEL Window:** Opens a new AEL applet with the filter and default view associated with the clicked monitor box. If you select this option, you can specify what actions are executed when you click or double-click a row in the AEL.
  - **Script:** Executes a custom JavaScript when you click the monitor box.
3. Optional: If you selected the **Script** option, type a Java script in the **Script** field. You can use the following tokens in the script:

#### **\$(FILTER)**

The name of the filter associated with the monitor box that is clicked.

#### **\$(DATASOURCE)**

The data source of the monitor box that is clicked.

#### **\$(PORTLETNAMESPACE)**

The portlet namespace of the Event Dashboard portlet.

For sample scripts, see the *IBM Tivoli Netcool/OMNIBus Installation and Deployment Guide*.

4. Optional: If you selected the **Open AEL in New Window** option, select the actions that you want to be performed when you click or double-click a row in the AEL.
  - **Event list single-click action:** Select the action to perform when you click an event in the AEL once. You can select default actions, such as opening the information window for the selected event, or you can select tools to be run on event data. You can create tools in the Tool Creation editor.
  - **Event list double-click action:** Select the action to perform when you double-click an event in the AEL. You can select default actions, such as opening the information window for the selected event, or you can select tools to be run on event data. You can create tools in the Tool Creation editor.
5. Under **AEL Appearance**, specify the areas of the AEL that you want to be displayed. Make these settings only if you selected **Open AEL in New Window** from the **Single-Click Action** list.
6. To save and apply your settings to the portlet, click **OK**.

## Example

The following example shows the script that is executed if the Update AEL option is selected, and you click the distribution indicator of a monitor box.

```
var ev = { "name" : "http://ibm.com/TIP#itmBroadcastEvent" ,
 "item_entityName" : $(FILTER),
 "item_dataSource" : $(DATASOURCE)
};
$(PORTLETNAMESPACE).sendPortletEvent(ev);
```

For more information about creating scripts that run from the Event Dashboard, see the *IBM Tivoli Netcool/OMNIBus Web GUI Administration and User's Guide*.

### Related concepts


"The Web GUI in a load balancing environment" on page 93

## Creating a page with an Event Dashboard and Active Event List

To obtain a high-level overview of events and be able to jump to an in-depth view of specific events, create a page that contains the Event Dashboard portlet and the Active Event List (AEL) applet, and customize the Event Dashboard to refresh the AEL when you click a monitor box.

### Before you begin

To create pages in Tivoli Integrated Portal, your user requires the `isc_admins` role.

1. From the navigation, click **Settings > Page Management**.
2. On the Page Management page, click **New Page**.
3. On the Choose a Portlet page, select **Event Dashboard** and click **OK**.
4. Click **Horizontal Split** .
5. On the Choose a Portlet page that is displayed below the Event Dashboard portlet, select **Active Event List (AEL)** and click **OK**.
6. Click **Save**.
7. On the Save Page window, edit the general properties of the page:
  - a. Type a name for the page.
  - b. Specify the location of the page in the navigation, either by accepting the default or by clicking **Location** and navigating to the required location.
  - c. From the **Page Persistence** list, select **Client Side**.
8. To specify user access to the page:
  - a. Click **Roles with Access to this Page** and click **Add**.
  - b. From the **Available Roles** list, select the required roles and click **Add**.
9. Optional: If you have defined Tivoli Integrated Portal views, add the page to a view:
  - a. Click **View Membership** and click **Add**.
  - b. From the **Available Views** list, select the required views and click **Add**.
10. Click **Save**.
11. To customize the Event Dashboard to refresh the AEL:
  - a. On the Event Dashboard portlet toolbar, click **Edit options > Edit Preferences**.

**Tip:** To set the portlet preferences for all users, click **Edit options > Edit Defaults**.

- b. From the **Single-Click Action** list, select **Update AEL on Same Page**.
  - c. Click **OK**.
12. To test the interaction between the Event Dashboard portlet and the AEL applet, in the Event Dashboard, click the distribution indicator of one of the monitor boxes.

## Results

On the page, the AEL is updated and now contains only the events that are captured by the filter specified for the monitor box.

### Related concepts

“The Web GUI in a load balancing environment” on page 93

### Related tasks

“Setting Event Dashboard portlet preferences and defaults” on page 228

---

## Visualizing event information on maps

You can use maps to graphically represent the status of a network.

### Related concepts

“About maps”

## About maps

Maps are Web applets that you use to create a view of a network, for example of a network topology or a geographical overview. You can overlay a background image with interactive widgets, called *map objects*, that can be configured to display alert data, and to open event lists for a detailed view of the network.

## Map resources

Map resources are the background graphic for the map or any icons that you want to use as map objects. The background typically represents a network topology or a geography, although you do not have to specify a background image. Default graphics and icons are provided; alternatively, you can upload additional background images and icons as map resources. Map resources can be .gif files, .jpeg files, or .png files.

## Map objects

Map objects are items that you can put on a map, to represent part of a network topology, for example. Map objects include buttons, lines, or icons.

After you have put an object on a map, you can customize the appearance and behavior of the map object, which are determined by the type of map object. The types of map object are as follows:

### Active objects

Map objects that can display alert severity information. You associate an active object with a filter; when the map is displayed, the color of an active object changes to show the highest severity status alert that is captured by the filter. You can also turn on hover help for the object, which displays information from the filter.



**Inactive objects**

Map objects that do not display alert severity information, and can be associated only with the URL of another Web page, for example, a page that contains another map.

**Monitor boxes**

Use monitor boxes to display detailed alert severity information on a map. You associate a monitor box with a filter; when the map is displayed, the distribution meter of the monitor box displays the range of events captured by the filter. You can customize the distribution meter to be a histogram or lavalamp; you can also customize the appearance of the monitor box.

**Text** Use a text object to write text directly onto a map.

Active map objects, monitor boxes and text objects can be associated with the following applets and pages:

- Active Event List (AEL)
- Lightweight Event List (LEL)
- Table View
- URL of a Web page, for example, a page that contains another map

**Map preferences**

Each map has a number of preferences, that determine its overall appearance. You can adjust these preferences to suit your needs and the use that the map is put to.

**Map name**

The identity of the displayed map. You can choose from a number in a drop-down list.

**Sound URL**

The URL of a sound file to play when the map is updated due to a refresh operation.

**Refresh rate**

the time (in seconds) between refresh operation on the map.

**Hover help for active objects**

Determines whether to display hover help for active objects on the map that are associated with filters. This can help users determine what action is associated with an object.

**Status bar**

Determines whether to display a status bar with the map. The bar includes a countdown timer until the next refresh operation.

**Use Customiser**

Determines whether to use the height specified in the map or the value specified in the **Height** preference.

**Height**

Specifies the height to use when displaying the map. This value works in conjunction with the **Use Customiser** preference. When in use this value overrides the height specified in the map definition.

**Map editors**

To add map objects to a map, and customize the map objects, use the Java Map Editor, which provides a graphical interface for customizing maps, or the HTML



Map Editor, in which you customize the map by writing HTML code in a text field.

## Publishing maps

To make maps visible to other users, add the maps to the navigation in Tivoli Integrated Portal.

First, you create a page in the Tivoli Integrated Portal. Then, you add either of the following portlets to the page:

- Map
- Web widget

Finally, you reference the map by editing the portlet preferences.

### Related tasks

Chapter 5, “Filtering event information,” on page 283

“Visualizing event information on maps” on page 313

## Previewing maps

As an administrator, after Tivoli Netcool/OMNIBus is installed, to familiarize yourself with maps, you can view sample maps that are deployed within the Web GUI.

The Web GUI provides three default maps that communicate with a simnet probe. The simnet probe sends simulated events to the ObjectServer; the probe is provided with Tivoli Netcool/OMNIBus. After you start the simnet probe, the maps are populated with events.

To preview maps:

1. To run the simnet probe, from the command-line interface, enter the following command:  
`$NCHOME/omnibus/probes/nco_p_simnet -server NETCOOLPRI`
2. In Tivoli Integrated Portal, click **Administration > Event Management Tools > Example Maps**.
  - To view a sample customer map that uses active buttons to open Active Event Lists, click **Customers**
  - To view a sample eCommerce map that uses active icons, represented by the default graphics, click **eCommerce**
  - To view a sample network on a geographical map that uses active buttons to open monitor boxes, click **Geographic**

## Creating maps

After you have added the resources that you want to use in your map, you can create the map itself.

### Related concepts

“The Web GUI in a load balancing environment” on page 93

## Creating maps using the Java Map Editor

Use the Java Map Editor to create and edit maps.

To create a map using the Java Map Editor:

1. Click **Administration > Event Management Tools > Maps > Map Creation**.
2. To open the Java Map Editor, click **Java > New**.
3. In the Java Map Editor applet, type a name for your map in the **Input** field.  
The map name must be alphanumeric and cannot contain spaces.
4. Click **OK**.

### What to do next

You can now customize the map properties using the Java Map Editor.

#### Related tasks

“Customizing maps using the Java Map Editor” on page 317

## Creating maps using the HTML Map Editor

Use the HTML Map Editor to create and edit maps.

To create a map by using the HTML Map Editor:

1. Click **Administration > Event Management Tools > Maps > Map Creation**.
2. Click **HTML** and then click **New**.
3. In the HTML Map Editor, use the following fields and buttons to provide the map settings:

#### Map name

Type a name for the map. The map name must be alphanumeric and cannot contain spaces.

#### Group name

Select the group for which you want to grant access to the map. The \* group grants access to all users including any new users you create.

#### Background image

Type the name of a background image, for example `background.gif`.  
The image must be in either GIF or JPEG format and must be uploaded as a map resource.

#### Background color

Type the color for the background.

#### Height

Type the height of the page in pixels.

**Width** Type width of the page in pixels.

4. Click **Submit**.

### What to do next

You can now specify the details of the new map by using HTML map editing commands in a text editor.

#### Related tasks

“Customizing maps in HTML” on page 328

## Customizing maps

After you have created your map, you can customize it by adding map objects to the map, and editing the properties of the map objects.

### Related concepts

“The Web GUI in a load balancing environment” on page 93

### Customizing maps using the Java Map Editor

After you have created a map, use the Java Map Editor to add active or inactive objects to the map, and configure the properties of the objects. Also customize the properties of the map, such as the access permissions, size, and background image.

To customize a map using the Java Map Editor:

1. Click **Administration > Event Management Tools > Maps > Map Creation**.
2. Select the map that you want to customize.
3. Click **Java** and then click **Modify**.

### Results

The Java Map Editor is launched with the properties of the selected map loaded.

### What to do next

Now use the Java Map Editor to customize the map properties and the map objects.

#### Setting access to a map:

You can define which user group has access to a map.

To restrict access to a map to a specific user group:

1. Open the Java Map Editor with the map you want to modify.
2. In the **Group Access** select the user group that has access to the map.  
To specify unrestricted access to a map, choose the entry \*.

**Note:** Users with the mcw\_admin role can edit any map on the system. However, they can view only unrestricted maps or maps available to groups that the users are members of.

3. Click **File > Save** to save the settings of the map.

#### Setting map size and background:

Use the Java Map Editor to set the background image for your map. Use the Color Picker to set the background color for your map.

To set the size and background of the map:

1. Open the Java Map Editor with the map you want to modify.
2. Use the following fields and buttons to set the size and the background properties of the map:

##### Map Size

Set the width and height of the map in pixels.

## Background

Set the background color for the map using the **Open a color picker** button.

Select a background image for the map from the list. The image overlays the background color. When you select a background image, the size of the map in pixels is automatically resized to the size of the background image.

**Tip:** Images that have been saved as resources for the current map appear in this list, and a number of generic images are provided by default.

3. Select **Server > Resync** to update the available images for the map. After adding a new resource, you must always update the Java Map Editor to include the new resource in the list of available items to be used in maps.
4. Click **File > Save** to save the settings of the map.

## Related concepts

"Color picker" on page 327

## Related tasks

"Previewing map resources" on page 337

## Setting the map grid for assistance:

Map items can be moved within the map by using either the mouse or the cursor keys. Use the grid feature to assist you in positioning and aligning these items.

To turn on the grid feature:

1. Open the Java Map Editor with the map you want to modify.
2. Click **View** and select the **Show Grid** check box. Alternatively, you can use Ctrl+G to switch on (and off) the map grid.
3. If it is not already selected, select the **Snap to Grid** check box in the **View** menu. When this feature is turned on, it causes all items on the map to snap to the nearest adjacent top and left-side grid lines.

If the **Snap to Grid** check box is not selected, you can also select the **Autolayout** option from the **View** menu to initiate an automatic adjustment of all objects on the map to the nearest adjacent top and left-side grid lines.

4. Resize the cell spacing of the map grid by selecting **Grid Size** from the **View** menu, and entering the desired cell space (in pixels).

## Results

The placement of map items is now controlled by their proximity to the adjacent top and left-side grid lines.

## Adding map objects and resources:









Use the Java Map Editor to add map resources, such as the background graphic for the map and graphics that you want to use as map objects, and also to add the map objects.

Map resources can be .gif files, .jpeg files, or .png files.

To add map resources and map objects:

1. Open the Java Map Editor with the map you want to modify.
2. To add map resources:
  - a. Click **File > Import Image**.
  - b. Select the required image and click **Open**.
  - c. Click **OK**.
  - d. Repeat until you have added all the required map resources.
3. To set the background for the map, select an option from the **Background** area:
  - To add a background color, click the Color Picker, or type a color in the field.
  - To use a map resource for the background image, for example, a geographical map, select a resource from the list.

If you select a map resource, the size of the map snaps to the size of the resource.

4. Add the required map objects:
  - To add a active button, click **Active rectangle button** . Optionally, specify the shape of the button; Rectangle, rounded rectangle or circle.
  - To add an inactive button, click **Rectangle button** . Optionally, specify the shape of the button; Rectangle, rounded rectangle or circle.
  - To add an active icon, click **New active icon** . Icons are added with a default placeholder graphic; you specify the graphic when you edit the properties of the map object.
  - To add an inactive icon, click **New icon** . Icons are added with a default placeholder graphic; you specify the graphic when you edit the properties of the map object.
  - To add an active line, click **New active line** .
  - To add an inactive line, click **New line** .
  - To add a monitor box, click **Lavalamp Monitor** . Optionally, specify the format of the distribution indicator: Lavalamp or histogram.
  - To add a text object, click **New text** .
5. Position the mouse pointer where you want to add the object on the map and click the map.

The new object is placed on the map.
6. To add all the required map objects, repeat steps 4 and 5.

## What to do next

You can now edit the properties of the map objects.

### Setting map object properties:

The properties of a map object control the appearance of the object and the information associated with it; the properties also include the name and the label. Depending on the type of object, the properties that you can set are displayed automatically.

### Before you begin

In the case of image objects, if you want to add a new image to an object, you must have added the image as a map resource. After you have added a new resource, you must refresh the Java Map Editor window.

In the case of monitor box objects, if you select extra information to be displayed on a monitor box (step 9b on page 321), you must make sure that the dimensions of the monitor box are large enough to accommodate the information. If the dimensions are too small, none of these details can be displayed and the monitor box might resemble a button on the map.

To configure the properties of a map object:

1. Open the Java Map Editor with the map you want to modify.
2. To open the Properties window, double-click the required object on the map.
3. Click **Properties**.
4. In the **Name** field, type the name of the object. Each object on a map must have a unique name.
5. In the **Label** field, type the label that you want to be displayed in the status bar of the Web browser when the mouse pointer is hovered over the object.
6. In the **Translucency** field, type a percentage value to control the level of translucency. 100% means that the object is completely translucent.

**Tip:** For active map objects: To use the glowing background effect for event severity, set a low translucency value.

7. To enable a shadow for the map object, select **Show Shadow**.
8. If the object is a button, set the following additional properties:
  - a. Under **Type**, select the shape of the button.

**Tip:** If you selected a rounded rectangle or elliptical button, to specify the corner radius of the button, click **Size & Position** and type a value in the **Arc Diameter** field.

- b. From the **Legend** list, select the text or value that you want displayed on the button.

**None** No text or value is displayed on the button.

**Label** Displays the text entered in the **Label** field.

**Count** For active buttons only: Displays the total number of alerts for the filter.

**Metric** For active buttons only: Displays the metric measurement.

9. If the object is a monitor box, set the following additional properties

- a. Under **Type**, specify how you want the distribution indicator to be displayed.
- b. Specify which information about alert distribution you want to be displayed on the monitor box:

**Show Label**

Displays the label of monitor box, as entered in the **Label** field

**Show Total**

Displays the total number of alerts in the distribution.

**Show Highest**

Displays the highest severity recorded for the alert distribution.

**Show Lowest**

Displays the lowest severity recorded for the alert distribution.

**Show Metric**

Displays the metric measurement.

**Show Severity Border**

Displays a color border around the monitor box border showing the highest alert severity.

**Tip:** Select this option if the alert distribution is large and the total number of alerts at the highest severity level is too small to appear in the distribution indicator.

10. If the object is an icon, set the following additional properties:

- a. Active icons only: Under **Type**, select the shape of the highlight bar that displays the event severity color.

**Tip:** If you selected a rounded rectangle or elliptical button, to specify the corner radius of the button, click **Size & Position** and type a value in the **Arc Diameter** field.

- b. Active icons only: From the **Legend** list, select the text or value that you want to be displayed on the icon.

**None** No text or value is displayed on the icon.

**Count** Displays the total number of alerts for the filter.

**Metric** Displays the metric measurement.

- c. From the **Image** list, select the image that you want to be displayed on the icon.

11. To save the settings, click **OK**.

12. To resynchronize the Java Map Editor with the Web GUI server, click **Server > Resync**.

## Linking map objects with URLs:

You can associate a map object with a URL. In the map, when the object is clicked, the URL is opened. The URL can point to a SmartPage template or other resource, such as a CGI script.

### Before you begin

CGI scripts must be registered before you can link a map object to the script.

If you link an active map object with a URL, you do not have to assign a filter.

**Restriction:** You cannot associate inactive icons or inactive lines with a URL.

To link a active map object to a URL:

1. Open the Java Map Editor with the map you want to modify.
2. To open the Properties window, double-click the required object on the map.
3. Click **Associations**.
4. From the **Action** list, select **Open URL**.
5. To link the object with a resource on the Web GUI server, in the **URL** field, type the URL of the resource.

To open an Active Event List (AEL), type a URL as shown in the following example:

```
$(SERVER)/AELView?filtertype=type&filtername=filtername&view=viewname&datasource=datasourcename
```

Where the parameters are as follows:

#### **\$(SERVER)**

Resolves to *protocol://server:port/context-root/webtop*. This enables maps to be transported to other Web GUI servers. The value *context-root* refers to the context root of the Web GUI. This can be configured during installation and the default value is *ibm/console*.

#### **datasource**

A data source defined in the *ncwDataSourceDefinitions.xml* data source configuration file. If you do not specify this parameter, the default data source is used. For more information about the default data source, see the *IBM Tivoli Netcool/OMNIbus Installation and Deployment Guide*.

**Tip:** If you have defined multiple data sources, you can specify them in the string by using a comma-separated list, as shown in the following example:

```
datasource=datasource1,datasource2
```

If you specify multiple data sources, make sure that the filter, either a predefined filter or a transient filter, queries fields that are contained in all data sources. If you select a data source that is not defined in the filter, an error message is displayed in the AEL instead of event data.

#### **filtername**

The name of the filter that you want to apply. If you do not specify this parameter, the default filter is used.

If you do not specify a **filtertype** parameter, all filter types are searched to identify a matching filter.



**filtertype**


The type of filter that you want to apply. Use this parameter together with the **filtername** parameter. Possible values are as follows:

- global
- system
- user
- user\_transient

**viewname**

A view that overrides the default view associated with the filter. If you do not specify this parameter, the default view associated with the filter is used.

6. To link the object to a SmartPage template:

- a. Click **Open the Template builder dialog**  .
- b. In the URL Build Tool window, select a SmartPage template type from the list.
- c. Enter the value of any variables contained in the template SmartPage tag source code.  
For example, in the `table.html` template, you must provide a map name that corresponds to the `Map_Name` variable, and an entity name that corresponds to the `Filter_Name` and `Filter_Type` variable.
- d. Click **OK**.

In the **URL** field, the template address and SmartPage tag variable details are displayed as a query string, for example:

```
$(SERVER)/Template/table.html?Map_Name=Summary&Filter_Name=
AllEvents&FilterType=Global
```

7. In the **Target** field, specify how you want the resource to be displayed:
  - To display the resource in a named IFrame portlet, type the name of the IFrame
  - To display the resource in same frame as the map, replacing the map, select **\_self**.
  - To display the resource in new Web browser window, select **\_blank**.
  - To display the resource in the parent frame set containing the source link, select **\_parent**.
  - To display the resource in the frame containing the source link, select **\_top**.
8. From the **Data Source** list, select a data source.  
The default is the default data source specified in the `ncwDataSourceDefinitions.xml` data source configuration file.
9. Active icons only: From the **Feedback** list, specify how event information is displayed in the icon:
  - **Fill Background**: The icon background changes color to denote event severity. If you select this option, the icon must have transparent areas for the feedback color to be visible in the map.
  - **Highlight Bar**: A bar is displayed below the map object. The bar changes color to denote event severity. To select the shape of the highlight bar, change the Type setting on the **Properties** tab.
  - **Glow Background**: The background of the icon changes color to denote event severity. If you select this option, change the Translucency setting on the **Properties** tab so that you can see the severity color behind the icon.

10. To save the settings, click **OK**.

#### **Related tasks**

Chapter 5, "Filtering event information," on page 283

"Registering CGI scripts" on page 257

#### **Related reference**

"Template overview" on page 409

Appendix F, "URLs for opening Web GUI pages," on page 415

#### **Linking map objects with event lists:**

You can associate an active map object with an Active Event List (AEL), Lightweight Event List (LEL), or Table View. When you click the object, the specified event list is opened.

To link an active map object with an event list, you must assign a filter to the object. If you do not assign a filter to an active button or an active line, the button or line is not displayed correctly on the published map. In the Java Map Editor, active buttons or lines to which no filters are assigned are yellow.

**Restriction:** You can assign only global filters and system filters to an active map object. You cannot assign user filters.

To link an active map object with an event list:

1. Open the Java Map Editor with the map you want to modify.
2. To open the Properties window, double-click the required object on the map.
3. Click **Associations**.
4. From the **Actions** list, select the event list that you want to be opened when you click the map object:
  - To open an AEL, click **Active Event List (AEL)**
  - To open an LEL, click **Lightweight Event List (LEL)**
  - To open a Table View, click **Event Table**
5. In the **Target** field, specify how the resource will be displayed:
  - To display the event list in a named IFrame, type the name of the IFrame
  - To display the event list in the full current browser window, replacing the map, select **\_self**
  - To display the event list in a new browser window, select **\_blank**
  - To display the event list in the parent frame set containing the source link, select **\_parent**
  - To display the event list in the frame containing the source link, select **\_top**
6. From the **Filters** list, select the required filter.
7. From the **Data Source** list, select one or more data sources.

The default corresponds to the default data source specified in the `ncwDataSourceDefinitions.xml` data source configuration file.

If you select a data source that is not defined in the selected filter, or if the data source contains fields that are not defined in the filter, an error message is displayed. You must select a filter in which the required data source is defined.
8. Active icons only: From the **Feedback** list, specify how event information is displayed in the icon:

- **Fill Background:** The icon background changes color to denote event severity. If you select this option, the icon must have transparent areas for the feedback color to be visible in the map.
  - **Highlight Bar:** A bar is displayed below the map object. The bar changes color to denote event severity. To select the shape of the highlight bar, change the Type setting on the **Properties** tab.
  - **Glow Background:** The background of the icon changes color to denote event severity. If you select this option, change the Translucency setting on the **Properties** tab so that you can see the severity color behind the icon.
9. To save the settings, click **OK**.

### Setting the size and positions of map objects:

You can set the size of a map object, and you can specify the position of a map object within your map.

To configure the size and position of a map object:

1. Open the Java Map Editor with the map you want to modify.
2. Open the map object Properties window using one of the following methods:
  - Double-click the object on the map.
  - Select the map object and click **Edit > Properties**.
  - Right-click an object and selecting **Properties** from the context menu.
3. Click the **Size & Position** tab.
4. Enter the distance in pixels from the left of the object to the left of the page in the **X** field. If the object is a line, enter a line-end coordinate in the **X2** field.
5. Enter the distance in pixels from the top of the object to the top of the page in the **Y** field. If the object is a line, enter a line-end coordinate in the **Y2** field.
6. Enter the width of the object in pixels in the **Width** field (buttons, monitor boxes, and active icons only).
7. Enter the height of the object in pixels in the **Height** field (buttons, monitor boxes, and active icons only).
8. If the map object is a **Rounded Rectangle** button or icon, you can change the arc of the corners by entering the diameter, in pixels, in the **Arc Diameter** field.

**Tip:** If **Feedback** is set to **Fill Background** in the **Associations** tab of an active icon, and the shape setting on the **Properties** tab is set to **Rounded rectangle**, you can modify the shape of the background color swatch by entering the diameter in the **Arc Diameter** field.

9. If the map object is a line, enter the thickness of the line in pixels in the **Thickness** field. The default thickness is 1.
10. Click **OK** to save the settings.

### Setting the color and font of map objects:

You can set the color and font of active map objects.

To specify color, you have the following options:

- Click **Color picker** to select a color.
- Type a permitted color name. Permitted entries are as follows:
  - black
  - blue
  - cyan
  - darkGray
  - green
  - lightGray
  - magenta
  - orange
  - pink
  - red
  - white
  - yellow
- Type the RGB hexadecimal value of a color (for example #FFFFCC).

**Restriction:** You cannot set the color of the following objects:

- Active buttons
- Active icons
- Inactive icons
- Active lines

You cannot set fonts for the following objects:

- Inactive icons
- Active lines
- Inactive lines

To configure the color and font of a map object:

1. Open the Java Map Editor with the map you want to modify.
2. On the map, right-click the required object and click **Properties**.
3. Click **Colors & Font**.
4. For inactive buttons:
  - a. In the **Color** field, specify the required color.
  - b. To make the button translucent, select the **Translucency** check box. For example, you might want to display only the text on the button, and no background.

**Tip:** If you create a translucent button that is the full size of the map and then place objects on top of it, the background behind the objects becomes clickable. You can make part of a .gif file clickable by drawing a translucent rectangle button over the required part of the .gif file.

- c. To change the appearance of the legend, select a font, a font size, and font color.

5. For active buttons: To change the appearance of the legend, select a font, a font size, font color, and font style.
6. For monitor boxes:
  - a. In the **Foreground** field and **Background** field, specify the required color.
  - b. To change the appearance of the label, select a font, a font size, font color, and font style.
7. For active icons: To change the appearance of the label, select a font, a font size, font color, and font style.
8. For inactive lines: In the color field, specify the required color.
9. For text:
  - a. Select a font, a font size, font style, justification, and font color.
  - b. To rotate the text, in the **Rotation** field, type the required value in degrees. The text rotates counter clockwise by the specified angle.
10. To save the settings, click **OK**.
11. To resynchronize the Java Map Editor with the Web GUI server, click **Server > Resync**.

#### *Color picker:*

The color picker is used to select the color of the map background, inactive buttons, lines, and text.

- When used to change the map background color, the **Color Picker** button is located to the right of the **Background** drop-down menu. The button displays the current background color.
- When used to change the color of inactive buttons, lines, or text, the **Color Picker** button is located on the **Color & Font** tab in the Properties window for the selected map object. The button displays the current color of the button, line, or text object.

When you click the **Color Picker** button, the *Select a Color* window is displayed.

The Select a Color window contains the following tabs:

- The **Swatches** tab shows a selection of predefined colors. Click a color to select it. The selected color is displayed in the sample color box. Click **OK** to apply the color to the background or to an item on the map.
- From the **HSB** tab you can choose the hue, saturation, and brightness of the color. Select the color from the chart, or enter the numerical values for hue, saturation, and brightness. The selected color appears in the sample color box. Click **OK** to apply the color to the background or to an item on the map.
- From the **RGB** tab you can specify the red, green, and blue color scales either by using the slider controls or by entering the numerical values. The selected color appears in the sample color box. Click **OK** to apply the color to the background or to an item on the map.
- The **Severity** tab shows the colors used for severity in the event list. The available options are:
  - Clear (green)
  - Indeterminate (purple)
  - Warning (light blue)
  - Minor (yellow)
  - Major (orange)
  - Critical (red)

Click a color to select it. The selected color appears in the sample color box. Click **OK** to apply the color to the background or to an item on the map.

- From the **Gray Scale** tab you can select any shade of gray between black and white by moving the slider control. Click **OK** to apply the color to the background or to an item on the map.

#### Related tasks

“Setting map size and background” on page 317

## Customizing maps in HTML

As an alternative to the Java interface, you can use the HTML Map Editor to customize maps. Enter the HTML map editing commands in the map display text field to configure maps.

To configure a map using the HTML Map Editor:

1. Click **Administration > Event Management Tools**.
2. Click **Maps > Map Creation**.
3. Select **HTML** and click **Modify**.
4. To specify the general properties of the map, use the following fields:

#### Map name

Type a name for the map. The map name must be alphanumeric and cannot contain spaces.

#### Group name

Select the group for which you want to grant access to the map. The \* group grants access to all users including any new users you create.

#### Background image

Type the name of a background image, for example `background.gif`. The image must be in either GIF or JPEG format and must be uploaded as a map resource.

#### Background color

Type the color for the background.

#### Height

Type the height of the page in pixels.

**Width** Type width of the page in pixels.

5. In the text entry field, type the map properties in HTML code.

**Tip:** For orientation, open an existing map in the HTML Map Editor.

6. Click **Save**.

#### Adding and configuring buttons:

Use the map editing commands of the HTML Map Editor to add buttons to your maps.

You can add two types of buttons to a map:

#### Inactive buttons

Inactive buttons can only be used to link to a URL.

#### Active buttons

Active buttons can also be used to show status color, and to open a Table View.

To add buttons to a map:

1. Open the HTML Map Editor with the map you want to modify.
2. In the text editor area, add the following button instruction:

```
button(name="button",label="example",x=12,y=34,w=56,h=78)
```

In these instructions, *button* is the unique name of the button, *example* is the label for the button, and the numeric values for *x*, *y*, *w* and *h* are in pixels. All button instructions start using this format, and each button has the following common properties:

- A name, which must be unique.
- A label, which appears in the status bar of the Web browser when the mouse pointer is over it.
- A position, which is defined by the coordinates of the top left corner of the button relative to the top left corner of the page.
- A button size, which is defined by the height and width.

**Note:** The variable *BSI* (Button Start Instruction) is used to represent this initial part of the instruction.

3. Click **Save**.

## Results

*Linking an inactive button to a URL:*

Use the map editing commands of the HTML Map Editor to link an inactive button on your map to a URL.

To link a button to a URL:

1. Open the HTML Map Editor with the map you want to modify.
2. In the text editor area, add the following button instruction:

```
BSI,action="go",url="$(SERVER)/newpage")
```

In this instruction, *BSI* is the first part of the button instruction and *newpage* is the destination HTML page.

The button appears gray.

3. Click **Save**.

## Related tasks

“Specifying a target” on page 333

*Linking an active button to a URL and displaying status:*

Use the map editing commands of the HTML Map Editor to link an active button on your map to a URL, and display the event status as the button color.

To link a button to a URL and display the status as the button color:

1. Open the HTML Map Editor with the map you want to modify.
2. In the text editor area, add the following button instruction:

```
BSI,filter="example",filtertype="type",action="go",url="$(SERVER)/newpage.html")
```

*BSI* is the first part of the button instruction, *example* is the name of a filter, *type* denotes the type of filter, which can be “system” or “global”, and *newpage* is the destination HTML page. The color of the highest-severity event from the alerts.status table, as captured by the filter, is used as the color of the button.

3. Click **Save**.

## Related tasks

"Specifying a target" on page 333

*Associating a Table View with an active button and displaying status:*

Use the map editing commands of the HTML Map Editor to associate an active button with a Table View.

To associate a Table View with a button and display the status as the button color

1. Open the HTML Map Editor with the map you want to modify.
2. In the text editor area, add the following button instruction:

```
BSI,filter="example",filtertype="type",action="table")
```

*BSI* is the first part of the button instruction, *example* is the name of a filter, *type* denotes the type of filter, which can be "system" or "global", and *table* defines that a Table View is opened when the button is clicked.

The color of the highest-severity event from the alerts.status table, as captured by the filter, is used as the color of the button.

The default view associated with the filter defines the columns in the Table View that are displayed.

3. Click **Save**.

## Adding and configuring icons:

Use the map editing commands of the HTML Map Editor to add icons to your maps.

You can add two types of icons to a map:

### Inactive icons

Inactive icons have no function on the map.

### Active icons

Active icons can link to a URL.

All icon instructions start using the format described in step two. Each icon has the following common properties:

- A name, which must be unique.
- A label, which appears in the status bar of the Web browser when the mouse pointer is over the icon.
- A position, which is defined by the coordinates of the top left corner of the icon relative to the top left corner of the page.
- An icon size, which is defined by the height and width.
- All icons must be in either GIF or JPEG format, and must be entered as resources on the map.

To add icons to a map:

1. Open the HTML Map Editor with the map you want to modify.
2. In the text editor area, add the following icon instruction:

```
icon(name="icon",label="example",x=12,y=34,image="image"
```

In these instructions, *icon* is the unique name of the icon, *example* is the label for the icon, the numeric values for *x* and *y* are in pixels, and *image* is the image file name.



**Note:** The variable *ISI* (Icon Start Instruction) is used to represent this initial part of the instruction.

3. To add an inactive icon, complete the instruction by adding a close parenthesis. This instruction adds an *inactive* icon.
4. Click **Save**.

*Linking an active icon to a URL:*

Use the HTML Map Editor and the HTML map editing commands to link an active icon on your map to a URL.

To link an icon to a URL:

1. Open the HTML Map Editor with the map you want to modify.
2. In the text editor area, add the following icon instruction:  
`ISI,action="go",url="$(SERVER)/newpage")`  
*ISI* is the first part of the icon instruction, and *newpage* is the destination HTML page.
3. Click **Save**.

#### **Related tasks**

“Specifying a target” on page 333

#### **Adding and configuring lines:**

Use the map editing commands of the HTML Map Editor to add lines to your maps.

You can add two types of lines to a map:

##### **Inactive lines**

Inactive lines can only be used to link to a URL.

##### **Active lines**

Active lines can also be used to show status color, and to open a Table View.

All line instructions start using the format described in step two, and each line has the following common properties:

- A name, which must be unique.
- A label, which appears in the status bar of the Web browser when the mouse pointer is over the line.
- A position for each end of the line, which is defined by the coordinates relative to the top left corner of the page.
- A thickness can also be applied to the line. The default line thickness is 1 pixel.

To add lines to a map:

1. Open the HTML Map Editor with the map you want to modify.
2. In the text editor area, add the following line instruction:  
`line(name="line",label="example",x=12,y=34,x2=56,y2=78,thickness=9`  
*line* is the unique name of the line, *example* is the label for the line, the numeric values for *x*, *y*, *x2*, *y2*, and *thickness* are in pixels.

**Note:** The variable *LSI* (Line Start Instruction) is used to represent this initial part of the instruction.

### 3. Click **Save**.

*Setting the color for an inactive line:*

Use the map editing commands of the HTML Map Editor to set the color of an inactive line.

To complete the line instruction for an inactive line and set the color of the line:

1. Open the HTML Map Editor with the map you want to modify.
2. In the text editor area, add the following line instruction:

```
LSI,color=black)
```

*LSI* is the first part of the line instruction.

The value of color can be the name of a system color (for example, black) or its hexadecimal equivalent (for example, #Ff1dC4).

### 3. Click **Save**.

*Linking an inactive line to a URL:*

Use the HTML Map Editor and the HTML map editing commands to link an inactive line on your map to a URL.

To link a line to a URL:

1. Open the HTML Map Editor with the map you want to modify.
2. In the text editor area, add the following line instruction:

```
LSI,action="go",url="$(SERVER)/newpage")
```

*LSI* is the first part of the line instruction and newpage is the destination HTML page.

The line appears gray.

### 3. Click **Save**.

#### **Related tasks**

"Specifying a target" on page 333

*Associating a Table View with an active line and displaying status:*

Use the HTML Map Editor and the HTML map editing commands to associate an active line with a Table View.

To associate a Table View with a line and display the status as the line color

1. Open the HTML Map Editor with the map you want to modify.
2. In the text editor area, add the following line instruction:

```
LSI,filter="example",filtertype="type",action="table")
```

*LSI* is the first part of the line instruction, *example* is the name of a filter, *type* denotes the type of filter, which can be "system" or "global", and *table* defines that a Table View is opened when the line is clicked.

The color of the highest-severity event from the alerts.status table, as captured by the filter, is used as the color of the line.

The view associated with the filter defines the columns in the Table View that are displayed.

### 3. Click **Save**.

### Adding text to a map:

Use the map editing commands of the HTML Map Editor to add text to the map background image.

The text is unable to perform any special function, such as link to a URL.

All text instructions use the format described in step two, and have the following common properties:

- A name, which must be unique.
- A label, which appears as the required text.
- A position, which is defined relative to the top left corner of the page.
- A font (default helvetica).
- A size (default 10 point).
- A style (default plain).

To add text to a map:

1. Open the HTML Map Editor with the map you want to modify.
2. In the text editor area, add the following text instruction:  

```
text(name="text",label="example ",x=12,y=34,font="font",size=12,
style="style", color="color")
```

text is the unique name for the text, example is the text you want to enter, the numeric values for x and y are in pixels, font is the new font, the numeric value for size is in points, style can be plain, bold or italic, and the value of color can be either the name of the color (such as black), or the RGB numeric value of the color (for example #ffffcc).
3. Click **Save**.

### Specifying a target:

For all map entries that have a link to a URL, you can also specify the target. When the target is omitted, the URL replaces the map in the current Web browser window.

The target option is in the format:

target="string"

In this example, string is the destination where the URL appears.

Where frames have been defined in the HTML page, you can specify the target as the name of the frame. For example UpperFrame or LowerFrame.

### Linking maps:

You can transfer maps from one Web GUI server to another.

- To link from one map page to another, use a URL, for example:  
*protocol://server:port/webtop/mappage*

Where the parameters are as follows:

#### **server**

The name of the host on which the Web GUI server is located.

**port**

The port number of the Web GUI server.

**mappage**

The destination HTML map page.

- To export map pages to a Web GUI server on another computer, construct the URL as follows:

`$(SERVER)/mappage`

Where the parameters are as follows:

**\$(SERVER)**

Resolves to *protocol://server:port/webtop*.

**mappage**

The destination HTML map page.

## Changing the color of map elements that have no associated events

To denote active elements on maps that have no associated events, assign a color that is applied to these elements. When users display a map, such active elements are displayed in the specified color.

By default, the active elements that have no associated events are displayed in the same color that is used for events with severity 0 (clear).

**Tip:** After you have edited the Web GUI server.init file, you must restart the Tivoli Integrated Portal server.

1. Open the *install\_dir/profiles/TIPProfile/etc/webtop/server.init* file.
2. Uncomment the **maplet.noeventcolor** parameter.
3. Specify a hexadecimal color value. The **maplet.noeventcolor** has no default, so you must specify a value. For example:
  - To specify gray, type: 0xDDDDDD
  - To specify white, type: 0xFFFFF
4. Save and close the file.
5. Restart the server.

**Related tasks**

"Restarting the server" on page 1

## Publishing maps

After your map is completed, you can publish it by adding it to the navigation in Tivoli Integrated Portal. You have two ways to publish a map.

**Related concepts**


"The Web GUI in a load balancing environment" on page 93

## Publishing maps on a Map portlet

To make a map available to Web GUI users, create a new page in Tivoli Integrated Portal, add the Map portlet to the page, and select the map that you want to display on the portlet.

To add a map to a page:

1. Click **Settings > Page Management**.
2. On the Page Management page, click **New Page**.
3. On the Page Settings window, edit the general properties of the page:
  - a. Type a name for the page.
  - b. Specify the location of the page in the navigation, either by accepting the default or by clicking **Location** and navigating to the required location.
4. To specify user access to the page:
  - a. Click **Optional setting** and click **Add**.
  - b. From the **Available Roles** list, select the required roles and click **Add**.
5. Click **OK**.
6. On the Choose a Portlet page, select **Map** and click **OK**.

The new page is displayed with the map portlet; currently the map portlet is blank, except for a message telling you how to add a map to the page.
7. Add a map to the page:
  - a. On the title bar of the newly created page, click **Edit Options**  > **Edit shared settings**.
  - b. Select the map from the list and set its characteristics as required.
  - c. Click **OK**.
8. Click **Save**.
9. Optional: To add further portlets to the page, proceed as follows:
  - To split the page vertically, click **Vertical Split**.
  - To split the page horizontally, click **Horizontal Split**.


Then, select the portlet that you require for the new section of the page, and customize the portlet as appropriate.
10. Optional: If you have defined Tivoli Integrated Portal views, to add the page to a view:
  - a. Click **View Membership** and click **Add**.
  - b. From the **Available Views** list, select the required views and click **Add**.

## Publishing maps on a Web widget portlet

To make a map available to Web GUI users, create a new page in Tivoli Integrated Portal, add the Web widget portlet to the page, and specify the URL of the map that you want to display on the portlet.

To add a map to a Web widget portlet:

1. Click **Settings > Page Management**.
2. On the Page Management page, click **New Page**.
3. On the Page Settings window, edit the general properties of the page:
  - a. Type a name for the page.
  - b. Specify the location of the page in the navigation, either by accepting the default or by clicking **Location** and navigating to the required location.
4. To specify user access to the page:

- a. Click **Optional setting** and click **Add**.
  - b. From the **Available Roles** list, select the required roles and click **Add**.
5. Click **OK**.
6. On the Choose a Portlet page, select **Web Widget** and click **OK**.
7. Add the map to the page:
  - a. On the title bar of the newly created page, click **Edit Options**  > **Edit shared settings**.
  - b. Set the properties of the Web widget:
 

**Widget title**  
Type a title for the map page.

**Home page**  
Type the location of the map relative to the root context of the Web GUI:  
  
`webtop/Map/mapname`  
  
 Replace the *mapname* with the name you supplied when creating the map.

**HTML iFrame name**  
Type a name for the iFrame that contains the map on the portlet. Make sure that the name is unique among all other iFrame names for all Web widgets.
  - c. Clear **Show a browser control toolbar** if you do not want the toolbar to appear on the page.
  - d. Set the check boxes for the items that non-administrative users can personalize. Use any combination of the following:
 

**Widget title**

**Home page**

**Help page**

**Browser control toolbar**
  - e. Click **Save**.
8. On the title bar of the page click **Save**.
9. Optional: To add further portlets to the page, proceed as follows:
  - To split the page vertically, click **Vertical Split**.
  - To split the page horizontally, click **Horizontal Split**.

Then, select the portlet that you require for the new section of the page, and customize the portlet as appropriate.
10. Optional: If you have defined Tivoli Integrated Portal views, to add the page to a view:
  - a. Click **View Membership** and click **Add**.
  - b. From the **Available Views** list, select the required views and click **Add**.

## Previewing map resources

Use the Map Resources portlet to preview images and icons for your maps.

To preview resources for a map:

1. Click **Administration > Event Management Tools > Maps > Map Resources**.
2. Select the map and click **Preview**. A list of all image resources available to the selected map is displayed.
3. Click **View** next to the image you want to preview.  
Images are listed by file name and file size, in bytes. The resource is opened in the next page.
4. To return to the list of images, click **Back** on your browser.
5. To return to the Map Resources portlet, click **Cancel**.

### Related concepts

"The Web GUI in a load balancing environment" on page 93

### Related tasks

"Setting map size and background" on page 317

## Deleting map resources

Use the Map Resources portlet to delete images and icons from the list of available resources.

To remove a resource:

1. Click **Administration > Event Management Tools > Maps > Map Resources**.
2. In the Map Resources portlet, select the map that contains the image you want to delete and click **Delete**.
3. From the **Available resources** list, select the image that you want to delete and click **Remove**. The image is removed from the Web GUI server.
4. To return to the Map Resources portlet, click **Cancel**.

### Related concepts

"The Web GUI in a load balancing environment" on page 93

---

## Visualizing high-level event information on charts

To help you visualize and compare large amounts of alert information, create charts based on event data

### Tivoli Netcool/OMNIBus Web GUI charts

Web GUI charts represent information graphically in different forms, and with scales that are added to indicate the values of the displayed data.

## Chart types

The Web GUI supports several different types of chart.

The following types of chart are supported:

### Bar charts

Consist of a set of elongated rectangles, the length of which indicate the number or frequency of a measured variable.

### Stacked-bar charts

The bar is subdivided so that you can see different subcategories within each category of a measured variable.

### Line graphs

Use connected lines to represent data, with each point on the line representing a value in the data range for a measured variable.

### Pie charts

Represent quantities in proportion to other quantities, for example, to show percentages of a whole.

The data displayed by these charts is drawn directly from a single ObjectServer table, and is unfiltered. To refine the appearance of the charts or constrain the breadth of data returned, SQL aggregate functions and comparison operators can be included in the data retrieval criteria.

## How charts are generated

Chart configuration and layout instructions are contained within a properly constructed XML-based definition file, and are processed by the Web GUI when a page containing a chart image tag is accessed.

For performance purposes, the Web GUI reads the chart definition file and stores configuration data in memory. During a page refresh, the server examines the chart configuration file timestamp. If the definition file is newer than the information stored in memory, the data is reloaded and the chart is re-rendered.

After the configuration data has been read, an image of the chart is rendered and placed on the page.

The process by which a chart is generated is as follows:

1. A client connected to the Web GUI requests a page containing one or more chart images.
2. The Web GUI server locates the requested chart page HTML file, and initiates the processing instructions contained within the chart `<img>` element or elements.
3. The `<img>` element contains a query string that provides details of the chart configuration XML file, image output format, image size and so forth. This information is sent to the chart renderer component.
4. The chart renderer component draws the data retrieval and presentation instructions from the specified XML file, and obtains the appropriate field information from the specified ObjectServer.
5. A chart image is rendered.
6. The chart image is returned to the Web GUI server for insertion into the Web page.
7. The chart page is processed.
8. The processed page is returned to the client.



## The chart definition file

The chart definition file contains all the configuration instructions that control how a chart obtains ObjectServer data, and how that data is presented in a graphical format.

A number of generic chart templates are provided with the Web GUI, and are located in *install\_dir/profiles/TIPProfile/etc/webtop/charts/definitions*:

- BAR\_eventsbylocation.xml for bar charts
- LINE\_eventsbylocation.xml for line graphs
- PIE\_eventsbyseverity.xml for pie charts
- SBAR\_eventsbyseverity.xml for stacked bar charts

Each chart schema begins with an element called <chartdata> that holds one child-element called <ncchart>. The <ncchart> element contains all the high-level chart component elements.

High-level component elements are those that govern the display or ObjectServer data-retrieval criteria for the chart. For example, the <header> element determines the content of the header above the chart display area, the <background> element controls the appearance of the chart background, and so forth.

The following syntax shows the structure of the chart definition file.

### Chart definition file

```
<chartdata>
 <ncchart>
 <header>
 </header>
 <footer>
 </footer>
 <chart3Dview>
 </chart3Dview>
 <background>
 </background>
 <antialias text value>
 <antialias value>
 <legend>
 </legend>
 <projector reversed value>
 <chartarea>
 </chartarea>
 <xaxis>
 </xaxis>
 <yaxis>
 </yaxis>
 <xscale>
 </xscale>
 <yscale>
 </yscale>
 <xgrid>
 </xgrid>
 <ygrid>
 </ygrid>
 <chartelement>
 ...
 </chartelement>
 </ncchart>
</chartdata>
```

**Tip:** The order of the high-level elements under <ncchart> is not strictly enforced. If you want, you can reorder the element structure when configuring your chart file.

An element reference is available for each of the elements and their child-elements, and an attribute reference is available for their associated attributes, and the attributes of their child-elements.

#### Related tasks

“Creating a chart definition file” on page 343

#### Related reference

“Element reference” on page 345

“Attribute reference” on page 353

#### <chartelement> syntax:

The <chartelement> element determines what ObjectServer field data is retrieved when a chart is rendered, and the type of chart and legend to be displayed. It is important to configure this element and associated child-elements correctly or your chart may not display properly.

The following example shows a code fragment containing the default contents of the <chartelement> element , with each line number displayed in brackets. The example uses the SBAR\_eventsbylocationseverity.xml as the basis for the example.

#### Chart Schema

```
1 <chartelement>
2 <charttype basetype="Cartesian">
3 <chartrenderer useAlertColors="true" type="BarChart"
4 mode="BarChartStacked">
5 </chartrenderer>
6 </charttype>
7 <dataconfig>
8 <query type="StackedSQL" datatype="ObjectServer">
9 <query_element action="select" fieldName="Location" fieldType="string"
10 tableName="alerts.status" where="" orderBy="Location Asc">
11 <constraint type="dataSetBeginCount" operator="equals" operand="5"
12 />
13 <constraint type="dataSetEndCount" operator="equals" operand="12"
14 />
15 </query_element>
16 <query_element action="select" fieldName="Severity"
17 fieldType="integer" tableName="alerts.status" where=""
18 />
19 <query_element action="count" fieldName="Severity"
20 fieldType="integer" tableName="alerts.status" where=""
21 />
22 </query>
23 <datasetlegend match="5" display="Critical" showValue="true"
24 showStart=" (total: " showEnd=)">
25 />
26 <datasetlegend match="4" display="Major" showValue="true" showStart="
27 (total: " showEnd=)">
28 />
29 <datasetlegend match="3" display="Minor" showValue="true" showStart="
30 (total: " showEnd=)">
31 />
32 <datasetlegend match="2" display="Warning" showValue="true" showStart="
33 (total: " showEnd=")">
34 />
35 <datasetlegend match="1" display="Indeterminate" showValue="true"
```

```

36 showStart=" (total: " showEnd=")"
37 />
38 <datasetlegend match="0" display="Clear" showValue="true" showStart="
39 (total: " showEnd=")"
40 />
41 <datasetlegend match="'__*__'" display="Unspecified" showValue="true"
42 showStart=" (total: " showEnd=")"
43 />
44 <datasource/>
45 </dataconfig>
46 </chartelement>

```

**Note:** The line numbers in the code fragment do not directly correspond to the lines in the SBAR\_eventsbyseverity.xml file.

The following subsections contain a line-by-line description of the element tags in the example XML code fragment.

### Lines 1-2

Line 1 opens the <chartelement> section of the chart configuration file. Line 2 contains the <charttype> element tag. The <charttype> element determines the type of chart created by the chart renderer. It has the attribute basetype which can be either “Pie” or “Cartesian”.

**Note:** Cartesian charts can have different data configuration requirements than pie charts; where applicable these differences will be pointed out.

### Lines 3-5

This line contains the <chartrenderer> element which establishes the image rendering mechanism employed and the type of chart created. If the basetype attribute in line 1 is set to “Cartesian”, then the type attribute for this element must be either “LineChart” or “BarChart.” If the basetype attribute is set to “Pie”, then the type attribute must be “PieChart.”

The <chartrenderer> element has an additional attribute called mode. If you decide to create a cartesian bar chart, you can specify whether you want it to contain stacked data or not. Leave this attribute blank if you want to create an unstacked chart. In this example the bar chart is stacked, and mode is set to “BarChartStacked.”

### Line 6

This line closes the <charttype> element tag.

### Line 7

This line contains the <dataconfig> element tag, which represents the start of the XML statements that control how data is obtained from the ObjectServer.

### Line 8

This line contains the <query> element tag. The type attribute for this element must be either “StackedSQL” or “StackedSQLGrouped” for stacked cartesian charts, or “BasicSQL” or “BasicSQLGrouped” for all other types of chart.

A <query> element containing the “BasicSQLGrouped” or “StackedSQLGrouped” statement uses the ObjectServer SQL GROUP BY clause when selecting data. This selection method enables Web GUI to instruct the ObjectServer to group into a single row all rows that have identical values in a specified column or combination of columns.

Used with aggregate functions, Web GUI can find the aggregate value for each group of column values. Because the data aggregation is taking place within the ObjectServer, fewer queries are sent by Web GUI and performance is enhanced.

The datatype attribute contains the name of the ObjectServer data source, for example NCOMS.

#### Lines 9-20

These lines contain <query\_element> element tags, which are child-elements of query. If your chart is an unstacked cartesian chart or a pie chart, you require two <query\_element> entries under <query>. If the chart is a stacked cartesian chart, you require three (or more) entries.

The <query\_element> tag contains a number of SQL compliant attributes that perform data interrogation functions. The action attribute is used within the <query\_element> element to determine the ObjectServer SQL action exerted on a specified dataset. For example, enter select to select the data located in the field specified by the subsequent fieldName attribute.

In addition to the select attribute, the action attribute can contain ObjectServer SQL aggregate functions such as count, max, min, avg, sum, and dist.

For more information about ObjectServer SQL syntax, see the *IBM Tivoli Netcool/OMNIBus Administration Guide*.

The orderBy attribute determines how selected data is ordered on the chart axis. The options are Asc (ascending) or Desc (descending), prefixed with a field name against which the ordering index is established. The orderBy attribute applies only to values on the x-axis.

The <query\_element> tag positions and their corresponding behaviors as follows:

**First** The first <query\_element> entry selects a field from the ObjectServer. In this example, the selected field is Location. In cartesian charts this entry is used to populate the x-axis. In pie charts, this entry represents the whole dataset (or the pie itself). Lines 11-14 contain range constraint criteria.

The <constraint> child-element is optional, but allows you to control how much data is returned from the selected field. This is useful as very large selection results often cannot be displayed in the available width of the chart. You can create a range of different chart configuration files where the only difference is the data constraint range.

**Last** The last <query\_element> entry measures the incidences of another ObjectServer field for each value returned by the first <query\_element>. In this example, the field name is Severity and the severity value is counted for each entry in the Location field.

In Cartesian charts this entry is used to populate the y-axis. In pie charts, this entry represents subsets of the whole dataset (or slices of the pie).

### Intervening

Intervening <query\_element> entries are only used by stacked cartesian charts. Their purpose is to further break down the data obtained by the first <query\_element> into smaller groups, which are then displayed as stacks within the major group.

In this example, the subset selected is the Severity field. The different levels of alert severity present within the major group—in this case, Location—are displayed as colored stacks within the chart.

### Line 22

This line concludes the <query> element section of the <chartelement> element.

### Lines 23-43

These lines contain the element tag <datasetlegend>. This element and its associated attributes establish the legend label conversion criteria for information received from a data source. For example, the if value 5 is matched in data returned from the ObjectServer, in this example the display attribute specifies a legend label conversion of Critical.

The showValue, showStart, and showEnd attributes provide the information that accompanies the label in the legend. The showStart and showEnd attributes act as a prefix ((total: ) and suffix ( )) to the value returned by showValue. In the previous example, the literal output is:

Critical (total: *number of critical alerts* )

### Lines 44-46

These lines contain the tags that close the <datasource>, <dataconfig>, and <chartelement> elements.

### Related reference

“Attribute reference” on page 353

## Creating a chart definition file

Create an XML chart definition file to specify the layout and display of your chart, and specify the event data on which to base your chart.

To help you create a valid chart, the following templates are located in the *webgui\_home\_dir/etc/charts/definitions* directory.

- BAR\_eventsbylocation.xml: Template for bar charts
- LINE\_eventsbylocation.xml: Template for line graphs
- PIE\_eventsbyseverity.xml: Template for pie charts
- PIE\_eventsbyseverity.xml: Template for stacked bar charts

To create a chart definition file:

1. Create an XML chart schema file to create a chart that displays high-level event data. To ensure that the charts you create are valid, use these template files as the basis for creating new charts.

**Tip:** If you are creating a pie chart, you can omit the following elements from the file: projectorreversed, chartarea, xaxis, yaxis, xscale, yscale, xgrid, and ygrid. Because these elements relate to Cartesian chart layout, they are ignored.

2. Configure your chart schema file, and save the file with a unique name.
3. Upload the file to the following directory on the Web GUI server:  
*webgui\_home\_dir/etc/charts/definitions*

#### Related concepts

"The chart definition file" on page 339

"The Web GUI in a load balancing environment" on page 93

## Publishing charts

After you have created the chart definition file, reference the file in HTML to add your Web GUI charts to Web pages.

To display your charts on Web pages, use one of the following methods:

- After you have created the chart definition file, you create an HTML file to display the chart. The HTML page must contain an `<img>` element that possesses the appropriate chart rendering attributes.  
 For a sample HTML page that contains an `<img>` element, see "Example."
- Create a page in Tivoli Integrated Portal and add the **Chart View** portlet.

### Example

The following example shows a chart `<img>` element within an HTML file.

```
<html>
 <head>
 <title>Demonstration Chart</title>
 <meta http-equiv="Content-Type" content="text/html;
 charset=iso-8859-1"
 >
 </head>
 <body>

 </body>
</html>
```

The attributes of the `<img>` element are as follows.

#### template

This attribute specifies which XML configuration file is used by the chart renderer. In this example, the file is called `barchart`. Do not include the file suffix (if any) when you set this attribute.

#### format

This attribute specifies the image type created by the chart renderer component. In this, example, the image type is a `.png` file. Do not modify this attribute

#### request

System attribute. Do not modify.

#### width

The first width attribute, which is within the double-quotes (" "), determines the width of the image produced by the chart renderer component. The second width attribute determines how the rendered image is resized, if at all, on the page. Set both attributes to the same value

**height**

The first height attribute, which is within the double-quotes (" "), determines the height of the image produced by the chart renderer component. The second height attribute determines how the rendered image is resized (if at all) on the page. Set both attributes to the same value.

**alt**

Use this attribute to enter a text equivalent for a graph image. This text is displayed when the cursor is placed above the image in the client browser.

**Related concepts**

"The Web GUI in a load balancing environment" on page 93

**Related tasks**

"Creating pages" on page 29

**Related information**

Creating pages

## Chart configuration reference

Read about the data contained within the Web GUI chart schema. It provides a comprehensive reference for the elements, attributes, and attribute values defined within the schema.

### Element reference

Read about the elements defined within the chart schema.

Elements often have one or more associated attributes, for which a value may be required.

The following table describes each element defined within the chart schema.

*Table 69. Chart element definitions*

Element	Description	Attribute(s)	Child Element(s)
annotation	<p>This element controls the text displayed for annotation values on the x-axis or the y-axis. If this element is not defined, the axes annotations are derived from field data.</p> <p>To specify a new annotation you must replace an existing field data annotation. For example, to replace host1 with New York, the annotation element is defined as:</p> <pre>&lt;annotation match="host1" text="New York"&gt;</pre> <p>To select integer indices, use the value attribute instead of the match attribute.</p>	text value match	None

Table 69. Chart element definitions (continued)

Element	Description	Attribute(s)	Child Element(s)
annotations	This element contains the elements that control the text and display characteristics of the labels on the x-axis or the y-axis.	None	labelrenderer annotation
antialias	This element smoothens the line rendering within the chart.	value	None
antialiastext	This element smoothens the text rendering within the chart.	value	None
background	This element determines the background color of the whole chart area.	value	None
border	This element contains the child-elements that control the appearance of the border (if any) around the legend area.	None	title
chartarea	This element controls the color or color gradient (if any), and margin positions of the area behind the chart itself.	None	plotareabackground margin
chartdata	The chartdata element is the root element. The root element is the top-level element in the XML document hierarchy, and contains all other elements in your XML file. Each document can have only one root element, and all other elements must be nested within it.	None	ncchart datasetlegend
chartelement	The chartelement element contains the all elements that govern chart rendering (that is, the type of chart created) and data presentation criteria (using SQL selection syntax).	None	charttype dataconfig



Table 69. Chart element definitions (continued)

Element	Description	Attribute(s)	Child Element(s)
chartrenderer	The chartrenderer element governs the appearance and behavior of each type of chart. It contains the style element which determines how colors are associated with chart data. In addition, this element controls which types of cartesian chart are displayed (bar or line) and whether cartesian charts are stacked.	useAlertColors  type  mode	style
charttype	The charttype element determines the category of chart produced by the chart renderer. It has the attribute basetype that establishes which type of chart is created (Pie or Cartesian).  In addition, this element contains the chartrenderer element that can further refine the appearance of the chart.	basetype	chartrenderer
color	This element provides the values used to create a solid swatch of color (one element entry) or a color gradient (more than one element entry) behind the chart.	value	None
constraint	This element contains attributes that can be used to restrict the range of data displayed on the x-axis or the y-axis. This element is optional.	type  operator  operand	None
dataconfig	This element contains a cluster of child-elements that are used to retrieve ObjectServer field data, and to create and populate the legend area of the chart.	None	query  datasetlegend  datasource

Table 69. Chart element definitions (continued)

Element	Description	Attribute(s)	Child Element(s)
datasetlegend	<p>This element is used to associate an appropriate legend label to matched information returned from the ObjectServer.</p> <p>For example:</p> <pre>&lt;datasetlegend   match="5"   display="Critical"   showValue="true"   showStart="(total: "   showEnd=")" /&gt;</pre> <p>The previous element tag matches returned severity data with the value 5 with the label Critical in the legend area of the chart.</p>	match display showValue showStart showEnd	None
font	<p>This element contains the child-elements that govern font size and appearance.</p> <p><b>Note:</b> Specify a font that supports all characters in the language you want to use for the chart.</p>	fontStyle fontName fontSize	None
footer	<p>This element contains the elements that determine the content of the footer below the chart display area. The footer typically displays additional useful information about the chart.</p>	None	text foreground background font
foreground	<p>This element is used by the header, footer, and legend border chart components to determine the color of text entries.</p>	value	None
gridline	<p>This element determines whether major or minor gridlines on the x-axis or the y-axis of the chart are visible. In addition, gridline also contains the elements that determine line color.</p>	type visible	paint

Table 69. Chart element definitions (continued)

Element	Description	Attribute(s)	Child Element(s)
header	This element contains the elements that determine the content of the header above the chart display area. The header typically displays the title of the chart.	None	text foreground background font
label	This element governs the appearance of the label (if any) displayed on the x-axis or y-axis of a cartesian chart.	color offset rotation visible	font
labelrenderer	This element governs the appearance of the text, background, and orientation of annotations on the x-axis or y-axis of a cartesian chart.	color background rotation	font
legend	<p>This element is used to determine the appearance and position of the legend area of a chart.</p> <p>The legend area contains interpretative information (for example, a color key) and quantitative information (for example, an event total) that further explains the contents of a pie chart or bar chart.</p>	visible position background foreground	border symbol font antialiastext antialias
margin	This element is used to determine the space (in pixels) around the chart data area. This margin area must be wide enough to accommodate axis data labels in cartesian charts.	top bottom left right	None

Table 69. Chart element definitions (continued)

Element	Description	Attribute(s)	Child Element(s)
ncchart	<p>This is the second-level element in the schema, and encapsulates all of the different component areas of the chart.</p> <p>Each ncchart child element is described in more detail elsewhere within this table.</p>	name	<p>header</p> <p>footer</p> <p>background</p> <p>antialiastext</p> <p>antialias</p> <p>legend</p> <p>projectorreversed</p> <p>chartarea</p> <p>xaxis</p> <p>yaxis</p> <p>xscale</p> <p>yscale</p> <p>xgrid</p> <p>ygrid</p> <p>chartelement</p>
paint	This element defines the type of system used to present gridlines.	type	None
plotareabackground	This element is used to create the background color or color gradient behind the chart data area.	orientation	color
projectorreversed	This element determines whether the chart data projection is reversed. When the associated value attribute is set to true, the datasets for the x-axis and y-axis are swapped. Note that the background also rotates.	value	None
query	This element contains the attributes and child-elements that determine what SQL data is obtained from which ObjectServer, and the manner in which it is retrieved.	<p>type</p> <p>datasource</p>	query_element

Table 69. Chart element definitions (continued)

Element	Description	Attribute(s)	Child Element(s)
query_element	The query_element tag contains a number of SQL-compliant attributes that perform data interrogation functions.	action fieldName fieldType tableName where	constraint
style	This element contains the child elements that define the border color and fill color of the pie or bar segments on a chart.	strokepaint fillpaint match	None
symbol	This element specifies the size and position (relative to adjacent text) of the color symbols used in the legend area of the chart.	height width textSpacing	None
text	This element is used by many other elements to determine the textual content of headers, footers, and so forth.	value	None
title	This element is used to control the position of the title in the legend area of the chart. It also contains the elements that control the text content and color.	position	text foreground font
xaxis	See xaxiselement.	None	xaxiselement
xaxiselement	This element controls how data is displayed on the x-axis, and covers presentation aspects such as axis orientation and data range parameters.	reversed autoDataMin autoDataMax autoDataRange dataMin dataMax visibleMin visibleMax	None
xgrid	This element contains the xgridelement element that controls how gridlines appear on the x-axis of the chart.	None	xgridelement

Table 69. Chart element definitions (continued)

Element	Description	Attribute(s)	Child Element(s)
xgridelement	This element determines the color of the gridlines on the x-axis, and contains the xgridelement element that controls what type of gridlines appear on the x-axis of the chart.	None	gridline
xscale	See xscaleelement.	None	xscaleelement
xscaleelement	This element allows you to control how chart components are arranged. This covers such features as the color of the foreground, the point at which the x-axis intersects with the y-axis, whether the data is logarithmically transformed, the axis title position, and so forth.	axisVisible crossingValue foreground logarithmic majorTickSize minorTickSize title titleRotation titlePlacement visible	annotations
yaxis	See yaxiselement.	None	yaxiselement
yaxiselement	This element controls how data is displayed on the y-axis, and covers presentation aspects such as axis orientation and axis data range parameters.	reversed autoDataMin autoDataMax autoDataRange dataMin dataMax visibleMin visibleMax	None
ygrid	This element contains the ygridelement element that controls how gridlines appear on the y-axis of the chart.	None	ygridelement
ygridelement	This element determines the color of the gridlines on the y-axis, and contains the ygridelement element that controls what type of gridlines appear on the y-axis of the chart.	None	gridline

Table 69. Chart element definitions (continued)

Element	Description	Attribute(s)	Child Element(s)
yscale	See yscaleelement.	None	yscaleelement
yscaleelement	This element allows you to control how chart components are arranged. This covers such features as the color of the foreground, the point at which the y-axis intersects with the x-axis, whether the data is logarithmically transformed, the axis title position, and so forth.	axisVisible crossingValue foreground logarithmic majorTickSize minorTickSize title titleRotation titlePlacement visible	annotations label

### Related concepts

“The chart definition file” on page 339

### Attribute reference

Read about the attribute types and values used within the chart schema.

Some attributes are enumerated and the values of these attributes are constrained to a list of predefined text strings. When enumerated attributes are used within the XML command file, they must be set to one of the values shown in the list. Default values (if any) are provided in the description.

The following table describes each attribute defined within the chart schema.

Table 70. Chart attribute definitions

Attribute	Constrained Values	Description
action	select   count   min   max	The attribute is used within the query_element element to specify the ObjectServer SQL action performed on a dataset. For example, select to select the data located in the field specified by the subsequent fieldName attribute.  A value must be provided for this attribute.
autoDataMax	true   false	This attribute is used by the xaxiselement and yaxiselement elements to specify whether you want the highest value in a dataset displayed as the highest value on the x-axis or y-axis of a graph.  If you do not provide a value for this attribute, the default is false.

Table 70. Chart attribute definitions (continued)

Attribute	Constrained Values	Description
autoDataMin	true   false	<p>This attribute is used by the <code>xaxis</code> element and <code>yaxis</code> element elements to specify whether you want to the lowest value in a dataset displayed as the lowest value on the x-axis or y-axis of a graph.</p> <p>If you do not provide a value for this attribute, the default is false.</p>
autoDataRange	true   false	<p>This attribute is used by the <code>xaxis</code> and <code>yaxis</code> elements, and specifies whether or not the data ranges on the axes are created automatically.</p> <p>If you set <code>autoDataRange</code> to false you can specify your own data ranges by populating the <code>dataMin</code> and <code>dataMax</code> attributes.</p> <p>If you do not provide a value for this attribute, the default is false.</p>
axisVisible	true   false	<p>This attribute is used by the <code>xscale</code> element and <code>yscale</code> element elements to specify if the x-axis or y-axis lines and labels are displayed.</p> <p>If you do not provide a value for this attribute, the default is false.</p>
background	None	<p>This attribute is used to specify the color for the background of a chart component. The attribute value is a base 16 hexadecimal color code and must be prefixed with a # symbol.</p> <p>If you do not provide a value for this attribute, a system default is used.</p>
basetype	Cartesian   Pie	<p>This attribute specifies the basic type of chart style. Note that bar charts, stacked-bar charts, and line graphs are all Cartesian.</p> <p>A value must be provided for this attribute.</p>
bottom	None	<p>This attribute is used within the <code>margin</code> element to specify the margin space at the bottom of the chart. The unit of measurement is in pixels.</p> <p>If you do not provide a value for this attribute, a system default is used.</p>
color	None	<p>This attribute is used within the <code>label</code> and <code>labelrenderer</code> elements to specify the color of label or annotative text on a chart. The attribute value is a base 16 hexadecimal color code and must be prefixed with a # symbol.</p> <p>A value must be provided for this attribute.</p>
crossingValue	-9218868437227405311 +9218868437227405311  IEEE754 Standard for Double Precision for Floating-Point Numbers. Values beyond these ranges incur rounding.	<p>This attribute is used by the <code>xaxis</code> element and <code>yaxis</code> element elements to determine where one chart axis crosses the other.</p> <p>If you do not provide a value for this attribute, the x-axis and y-axis intersect at 0.</p>



Table 70. Chart attribute definitions (continued)

Attribute	Constrained Values	Description
dataMax	-9218868437227405311 +9218868437227405311  IEEE754 Standard for Double Precision for Floating-Point Numbers. Values beyond these ranges incur rounding.	This attribute is used by the <code>xaxiselement</code> and <code>yaxiselement</code> elements to specify the highest value on the x-axis or y-axis of a graph. This attribute is ignored when <code>autoDataMax</code> is set to true.
dataMin	-9218868437227405311 +9218868437227405311  IEEE754 Standard for Double Precision for Floating-Point Numbers. Values beyond these ranges incur rounding.	This attribute is used by the <code>xaxiselement</code> and <code>yaxiselement</code> elements to specify the lowest value on the x-axis or y-axis of a graph. This attribute is ignored when <code>autoDataMin</code> is set to true.
datasource	None	This attribute is used by the <code>query</code> element and specifies the ObjectServer data source used to create the chart.
defaultGridColor	None	This attribute is used by the <code>xgridelement</code> and <code>ygridelement</code> elements to specify the color of the chart grid (if any). The attribute value is a base 16 hexadecimal color code and must be prefixed with a <code>#</code> symbol.
display	None	This attribute is used by the <code>datasetlegend</code> element to specify the text in the legend panel that accompanies the values returned by the <code>match</code> attribute. The attribute entry is a string.
fieldName	None	This attribute is used by the <code>datasetlegend</code> element to specify an ObjectServer field. For example, <code>Severity</code> .
fieldType	string   integer	This attribute is used by the <code>datasetlegend</code> element to describe the ObjectServer field-data characteristics.
fillpaint	#000000#FFFFFF	This attribute defines the fill color of the pie or bar segments on a chart. The attribute value is a base 16 hexadecimal color code and must be prefixed with a <code>#</code> symbol.  A value must be provided for this attribute.
fontName	None	This attribute is used by the <code>font</code> element and specifies the name of the font used in a text entry. The attribute entry is a string, and must correspond to a font located on the Web GUI server.  A value must be provided for this attribute.
fontSize	None	This attribute is used by the <code>font</code> element and specifies the size of the font used in a text entry. The attribute entry is in points, rounded to the integer.  A value must be provided for this attribute.
fontStyle	plain   bold	This attribute is used by the <code>font</code> element and specifies the style of the font used in a text entry. The attribute entry is a string, and must be supported by the font specified in the corresponding <code>fontName</code> attribute.  A value must be provided for this attribute.

Table 70. Chart attribute definitions (continued)

Attribute	Constrained Values	Description
foreground	#000000 → #FFFFFF	<p>This attribute is used to specify the color for the foreground of a chart component and often refers to the text entry. The attribute value is a base 16 hexadecimal color code and must be prefixed with a # symbol.</p> <p>A value must be provided for this attribute.</p>
height	$0 \rightarrow 2^{31}-1$	<p>This attribute is used by the symbol component to specify the height of the symbol or symbols used in the legend panel. The attribute entry is in points, rounded to the integer.</p> <p>The default value corresponds to a system-calculated best fit.</p>
left	$0 \rightarrow 2^{31}-1$	<p>This attribute is used by the margin element to specify the margin space at the left of the chart. The unit of measurement is in pixels and is an integer.</p> <p>The default value corresponds to a system-calculated best fit.</p>
logarithmic	true   false	<p>This attribute is used by the xscaleelement and yscaleelement elements and specifies whether the scale data is logarithmically transformed.</p> <p>If you do not provide a value for this attribute, the default is false.</p>
majorTickSize	None	<p>This attribute is used by the xscaleelement and yscaleelement elements and controls the size of the large ticks on the x-axis or the y-axis. The unit of measurement is in pixels and is an integer.</p>
match (annotation)	None	<p>This attribute is used by the annotation element to establish the string election criteria for annotations displayed on the x-axis or the y-axis. To select an integer, use the value attribute instead.</p>
match (datasetlegend)	None	<p>This attribute is used by the datasetlegend element to establish the selection criteria for the information displayed in the legend panel. This attribute works in combination with the display attribute.</p>
minorTickSize	$0 \rightarrow 2^{31}-1$	<p>This attribute is used by the xscaleelement and yscaleelement and controls the size of the small ticks on the x-axis or the y-axis. The unit of measurement is in pixels and is an integer.</p> <p>The default value corresponds to a system-calculated best fit.</p>
mode	BarChartStacked	<p>This attribute is used by the chartrenderer element to specify if a barchart is stacked. The corresponding type attribute must be set to BarChart for this attribute to be enabled.</p> <p>If you do not provide a value for this attribute, no value is used and the renderer uses its default mode.</p>

Table 70. Chart attribute definitions (continued)

Attribute	Constrained Values	Description
name	Must exactly correspond to the chart file name.  Do not including the .xml file extension.	This attribute provides a name for the chart.  For example, if the schema is called mychart.xml, the name attribute must be set to mychart.
operand		This attribute is used by the constraint element to provide an upper or lower limit (as determined by the accompanying type attribute) on the x-axis data returned by a query_element selection.
operator		System attribute—do not modify.
orderBy	<i>fieldname</i> Asc   <i>fieldname</i> Desc	This attribute determines how selected data is ordered on the chart axis. The options are Asc (ascending) or Desc (descending), prefixed with a fieldname against which the ordering index is established. For example:  <pre>&lt;query_element action="select"   fieldName="Location"   fieldType="string"   tableName="alerts.status"   where=""   orderBy="Location Asc"&gt;</pre> In this example, the Location field data is arranged in ascending order.
orientation	xaxis   yaxis	This attribute is used by the plotareabackground element to specify the color gradient direction for the chart background.  If you do not provide a value for this attribute, the default is xaxis.
position (legend)	NorthBottom   NorthEast   SouthTop	This attribute is used by the legend element to specify where the chart legend area is displayed on the chart. The anchor positions are defined relative to the chart.  If you do not provide a value for this attribute, the default is NorthBottom.
position (title)	AboveBottom   AboveTop   BelowBottom   BelowTop   Bottom   Center   Leading   Left   Right   Trailing   Top	This attribute is used by the title element to specify where the title is displayed in the legend. The anchor positions are defined relative to the chart. If you do not provide a value for this attribute, the default is Top.
reversed	true   false	This attribute is used by the xaxiselement and yaxiselement elements to specify the orientation order of the axis. For example, when set to true for the xaxiselement element, the x-axis appears to the left-hand side of the origin of the y-axis.  If you do not provide a value for this attribute, the default is false.
right	0->2 <sup>31</sup> -1	This attribute is used by the margin element to specify the margin space at the right of the chart. The unit of measurement is in pixels and is an integer.  The default value corresponds to a system-calculated best fit.

Table 70. Chart attribute definitions (continued)

Attribute	Constrained Values	Description
rotation	0->360	<p>This attribute is used by the <code>label</code> and <code>labelrenderer</code> elements to specify the clockwise rotation angle for labels and annotations. The angle value is expressed in degrees and is an integer.</p> <p>If you do not provide a value for this attribute, the default is 0.</p>
showEnd	None	This attribute is used by the <code>datasetlegend</code> element to specify a suffix to any count data displayed in the legend.
showStart	None	This attribute is used by the <code>datasetlegend</code> element to specify a prefix to any count data displayed in the legend.
showValue	true   false	<p>This attribute is used by the <code>datasetlegend</code> element to specify whether count data is displayed in the legend.</p> <p>If you do not provide a value for this attribute, the default is false.</p>
strokepaint	#000000#FFFFFF	<p>This attribute defines the border color of the pie or bar segments on a chart. The attribute value is a base 16 hexadecimal color code and must be prefixed with a # symbol.</p> <p>A value must be provided for this attribute.</p>
tableName	None	This attribute is used by <code>query_element</code> to specify which ObjectServer data table is used. For example, <code>alerts.status</code> .
text	None	This attribute is typically used to provide textual information within the chart configuration file.
textSpacing	0->2 <sup>31</sup> -1	<p>This attribute is used by the <code>symbol</code> element to specify the distance between symbols used in the legend area and their associated text. The unit of measurement is in pixels and is an integer.</p> <p>The default value corresponds to a system-calculated best fit.</p>
title	None	This attribute is used by the <code>xcscaleelement</code> and <code>yscaleelement</code> elements to specify the title (if any) associated with the x-axis or y-axis.
titlePlacement	0->100	<p>This attribute is used by the <code>xcscaleelement</code> and <code>yscaleelement</code> elements to determine the placement of an axis title. The measurement is a percentage of the entire width of the chart. For example, a value of 50 centers the title under the x-axis.</p> <p>If you do not provide a value for this attribute, the default is 50.</p>
titleRotation	0->360	<p>This attribute is used to specify the clockwise rotation angle for the title (if any) displayed on the x-axis or the y-axis. The angle value is expressed in degrees and is an integer.</p> <p>If you do not provide a value for this attribute, the default is 0.</p>

Table 70. Chart attribute definitions (continued)

Attribute	Constrained Values	Description
top	0->2 <sup>31</sup> -1	<p>This attribute is used by the margin element to specify the margin space at the top of the chart. The unit of measurement is in pixels and is an integer.</p> <p>The default value corresponds to a system-calculated best fit.</p>
type (chartrenderer)	LineChart   PieChart   BarChart	<p>This attribute is used by the chartrenderer element to specify the type of chart to graphically render. Note that if you enter LineChart or BarChart, the basetype attribute of the charttype element must be set to Cartesian. If you enter PieChart, the basetype attribute of the charttype element must be set to Pie.</p>
type (constraint)	dataSetBeginCount   dataSetEndCount	<p>This attribute is used by the constraint element to restrict the range of data returned from a selected field. These attributes work in combination with the operator and operand attributes to specify a data startpoint and endpoint.</p> <p>For the dataSetBeginCount the operand indicates the start point. For the dataSetEndCount the operand indicates the end point. Consider the following example query_element:</p> <pre>&lt;query_element action="select"   fieldName="Location"   fieldType="string"   tableName="alerts.status"   where=""   orderBy="Location Asc"</pre> <p>To return a range of locations starting at the 5th returned and ending at the 12th, set constraint as follows:</p> <pre>&lt;constraint type="dataSetBeginCount"   operator="equals" operand="5" /&gt;&lt;constraint type="dataSetEndCount"   operator="equals" operand="12" /&gt;</pre> <p>By default, returned data has no default constraints. To return the first 8 locations, set constraint as follows:</p> <pre>&lt;constraint type="dataSetEndCount"   operator="equalss" operand="8" /&gt;</pre>
type (gridline)	major   minor	<p>This attribute is used by the gridline element to specify the type of chart gridline to configure. All subsequent gridline attributes then perform actions on this selection.</p>
type (paint)	color	<p>System attribute—do not modify.</p>
type (query)	BasicSQL   BasicSQLGrouped   StackedSQL   StackedSQLGrouped	<p>This attribute is used by the query element to specify the type of SQL query to use. If you create a stacked chart you must use StackedSQL or StackedSQLGrouped, otherwise use BasicSQL or BasicSQLGrouped.</p>

Table 70. Chart attribute definitions (continued)

Attribute	Constrained Values	Description
useAlertColors	true   false	This attribute allows you to use the alert status colors provided by the alerts.colors table in the ObjectServer.  If you do not provide a value for this attribute, the default is false.
value	Various	This attribute is typically used to populate or define text, color, or boolean information within the chart configuration file.
visible	true   false	This attribute is used to specify whether an associated chart feature is displayed.  For example, the legend element uses visible to specify whether a legend displaying information about the drawn data is associated with a chart. When enabled, all the graphical representations displayed within the chart have a corresponding legend item.
visibleMax	-9218868437227405311+ 9218868437227405311  IEEE754 Standard for Double Precision for Floating-Point Numbers. Values beyond these ranges incur rounding.	This attribute is used by the xaxis element and the yaxis element to impose a visible constraint on the data range presented in a chart. The visibleMax attribute specifies the highest number displayed.  If you do not provide a value for this attribute, the default is the highest data entry on the chart.
visibleMin	-9218868437227405311+ 9218868437227405311  IEEE754 Standard for Double Precision for Floating-Point Numbers. Values beyond these ranges incur rounding.	This attribute is used by the xaxis element and the yaxis element to impose a visible constraint on the data range presented in a chart. The visibleMin attribute specifies the lowest figure displayed.  If you do not provide a value for this attribute, the default is the lowest data entry on the chart.
where	None	This attribute is used by the query_element query to specify any SQL search constraints you want applied to the selected data source.
width	0->2 <sup>31</sup> -1	This attribute is used by the symbol component to specify the width of the symbol or symbols used in the legend panel. The attribute entry is in points, rounded to the integer.

#### Related concepts

"The chart definition file" on page 339

#### Related reference

"<chartelement> syntax" on page 340

# Visualizing event information on gauges

You can graphically represent the values of various metrics on gauges.

To open the supplied pages of gauges click **Availability > Events > Performance** or **Availability > Events > Details and Journals**.

## Gauges and metrics

The Gauges page displays the values of selected metrics as a set of gauges. Each gauge has a number of properties.

- The metric that the gauge displays
- The appearance of the gauge
- Text labels
- Thresholds
- The actions taken when you click a gauge

In addition, the Web GUI is supplied with two pages of gauges.

### Metric

A metric is a type of measurement that is used to determine a quantifiable value from tables or properties in the ObjectServer. Examples of metrics are:

- The current number of client connections to a server
- The number of unresolved network events
- The number of escalated network events

The Web GUI has a number of pre-defined metrics that cover many of the commonly-used scenarios in network monitoring. However a site can define its own collection of metrics in addition to the supplied ones.

### Appearance of a gauge

Each gauge can have any of five appearances:

Table 71. Gauge icons






Name	Examples
Dial	
Thermometer	

Table 71. Gauge icons (continued)

Name	Examples
Traffic lights	
Weather symbol	
Emoticon	

The dial and thermometer indicate the value of the metric through the pointer and the height of the indicator. For example, on a dial, the needle moves up and down the scale as the value of the metric changes. The remaining icons show whether the value of the metric is low, medium, or high, as determined by the thresholds of the gauge.

Each gauge can use any of these appearances. This enables a site to choose the type of gauge most suitable for each metric.

## Text labels

Each gauge has three text labels that identify it:

- **Label:** The name of the gauge that appears on the Gauges page.
- **Unit label:** Which indicates the units of the gauge's value. For example, the number of connections, or the number of faults.
- **Description:** A more detailed description of the gauge and what it shows. The Web GUI displays this text when the user hovers the mouse pointer over the gauge.

## Thresholds

The set of values for each gauge is divided into three ranges: low, medium, and high. There are two thresholds that determine the transition from one range to another. Each threshold is expressed as a percentage of the gauges complete value range. For example, the threshold between low and medium may be 50% and the one between medium and high may be 90%.

The administrator can set the values of the thresholds for any gauge. This enables the displays to be tailored to the importance of the metric.

## Click actions

When the user clicks a gauge, the system can:

- Display a page associated with a fully-qualified URL, such as a map or a Lightweight Event List (LEL).
- Run JavaScript code



A gauge can also have no click action.

The URL and JavaScript capabilities mean that a wide range of actions can occur when a user clicks the gauge. For example, the system could display list of event associated with the gauge's metric, or launch another Web GUI application, or even applications from other parts of Tivoli that are based on Tivoli Integrated Portal.

## Working with gauges

A user accesses the Gauges to monitor conditions on the network. They can use the gauge values and the low, medium, and high ranges to identify conditions that need immediate attention. If a gauge has an associated click action, the user can get more detail of the gauge's metric.

A user with suitable user roles, such as the Web GUI administrator, can customize the page and its gauges. The user can:

- Change the title of the page.
- Set the refresh rate for the gauges.
- Select the data sources and ObjectServers that supply the page with data.
- Modify the properties of any gauge.
- Add and remove gauges.
- Change the order of the gauges on a page.
- Set the thresholds for any gauge's low, medium, and high ranges.
- Create new metrics.
- Change the internationalization settings for the page, such as text direction.

Administrators can also set the default appearance and content of the page for all users. So, the administrator can define the initial page for the site. Then an individual can tailor their copy of the page to suit their needs, if they have the necessary user roles.

## Supplied gauges

As supplied the Web GUI includes two pages of gauges:

- Performance: which shows key performance metrics for the gateway and the Web GUI such as the number of connections to the Web GUI.
- Details and Journals: which shows key event metrics such as the number of unresolved events and the number of acknowledged events.

To open the supplied pages of gauges click **Availability > Events > Performance** or **Availability > Events > Details and Journals**.

### Related tasks

"Setting the thresholds for a gauge" on page 368

"Setting gauge portlet preferences" on page 231

### Related reference

"Using the gauge page" on page 364

## Using the gauge page

A Gauges page provides a high-level overview of Tivoli Netcool/OMNIBus data.

The page displays data retrieved from the ObjectServer using *metrics*. Examples of metrics are the number of clients connected to a server, the total number of escalated events, or the time taken to process a trigger.

The Web GUI presents the data on the page as a number of gauges, one for each metric. The number of gauges on the page and the metrics they display are site-specific. Your Web GUI administrator can customize the page to show only the metrics that are relevant to you, and display each metric in the most appropriate way. The administrator can also create additional Gauges pages for specific types of user and/or specific functional purposes. If you have suitable user roles, you can:

- Customize your copy of a page still further.
  - Create an HTML representation of a page that you can send to mobile devices.
- This allows people to monitor the data from almost anywhere.

## Using the Gauges page

When you open the Gauges page it displays the gauges and sets each to the current value of its associated metric using data from the ObjectServer. The gauges periodically update to show the latest information. The frequency of these updates is 10 seconds, as supplied, but your administrator can specify any value between 10 and 99000 seconds.

Clicking a gauge may provide further information about the metric. For example, the Web GUI may display a Lightweight Event List (LEL) for the events that the metric measures.

What happens when you click a gauge is site-specific. Your Web GUI administrator can associate any URL or JavaScript program with a gauge. Typically, however, actions launch other Web GUI or Tivoli applications.

## User Roles

To use a Gauges page you need the following user roles:

- To use a page to monitor metrics you need to have either the `ncw_gauges_viewer` or `ncw_user` roles.
- To customize gauges you need to have the `ncw_gauges_editor` role and either of the `ncw_gauges_viewer` or `ncw_user` roles.

You must have the `ncw_gauges_viewer` role.

A user without administration rights can customize their own copy of the page only. Administrators can customize their own copy of a page and customize the default page, and its contents, for all users.

### Related concepts

“Gauges and metrics” on page 361

### Related tasks




“Setting gauge portlet preferences” on page 231

“Publishing Gauges pages to mobile devices” on page 365

## Publishing Gauges pages to mobile devices

You can make the gauges displayed on a Gauges page available to Web GUI users on mobile devices, or in a Web browser.

To publish a Gauges page to a mobile device, generate an HTML file of the page and send its URL to a mobile device:

1. Open the gauge page that you want to publish.
2. Edit your portlet preferences, or, as an administrator, edit the portlet defaults:
  - Click **Personalize**  .
  - To edit the portlet defaults of all users, click **Edit Options**  > **Edit shared settings**, or to edit your own preferences, click **Edit Options**  > **Personalize**.
3. Select the **HTML for mobile devices** check box if it is cleared.
4. Optional: Change any other properties of the page or the gauges as required.
5. Click **OK**.

The Web GUI generates the HTML page and displays its URL in **URL for mobile devices**.

6. Open the portlet preferences or portlet defaults again.
7. Optional: To preview the page, click the URL.
8. Copy the URL and send it to users of mobile devices in an e-mail or SMS message. you can also send the URL to any other user of a browser who needs access to the page.
  - Before they can view the page, each recipient needs to log in to the Web GUI as a user with the `ncw_gauges_viewer` role.
  - Ensure that each recipient bookmarks the URL so that they can return to the page at any time.
  - Repeat this step each time you change any properties of the Gauges page or its gauges.

### Related tasks

“Setting gauge portlet preferences” on page 231

## Creating and managing metrics for gauges

Use the WAAPI capabilities to create and manage metrics for gauges.

WAAPI contains methods, elements and attributes that you use to manage metrics:

- Create a metric
- Replace a metric, or create it if it does not exist
- Modify a metric
- Delete a metric
- Obtain a list of the existing metrics

### WAAPI methods, elements, and attributes

Use the following WAAPI methods to manage metrics for gauges:

Table 72. WAAPI elements for gauge metrics

Task	WAAPI method name
Create a metric	<code>metric.createMetric</code>

Table 72. WAAPI elements for gauge metrics (continued)

Task	WAAPI method name
Replace or create a metric	metric.createOrReplaceMetric
Modify a metric	metric.modifyMetric
Delete a metric	metric.deleteMetric
List the existing metrics	metric.getList

For operations on metrics, the <method> element contains the <metric:metric> element. This element contains attributes that identify the metric and its characteristics. It also contains the <metric:command> element that defines the query to obtain the required data from the ObjectServer. In addition, there are two modes for creating, replacing, or modifying a metric: basic and advanced. Basic mode ensures that any restriction filters placed on the referenced ObjectServer tables are applied and so the user receives only the data that they are entitled to see. Advanced mode does not apply any restriction filters and the user sees all the data returned by the query. The examples show the use of these modes, attributes and the child element.

## Examples

The following examples show how to use WAAPI to manage metrics for gauges. The `samlrequest_metric.xml` and `samlrequest_metric_basic.xml` files in `webgui_home_dir/waapi/etc/samples` contains these examples.

### Create a metric

The following WAAPI command creates a metric named `metricsample1` that displays the count of all critical events in the ObjectServer. It uses advanced mode, the default mode, and so does not apply any restriction filters that may be defined.

```
<methodCall xmlns:metric="http://www.ibm.com/tivoli/netcool/webtop/metrics/7.3.1">
 <method methodName="metric.createMetric">
 <metric:metric name="metricsample1"
 displayName="MetricSample1"
 description="Shows count of all Critical events. Created using WAAPI."
 units="events"
 maxValue="10000"
 minValue="0"
 threshold1="30"
 threshold2="70">
 <metric:command type="sql">
 <metric:text data="select sum(Tally) from alerts.status where Severity=5;"/>
 </metric:command>
 </metric:metric>
 </method>
</methodCall>
```

The following example creates the same metric but uses basic mode that applies any restriction filters defined for the user.

```
<methodCall xmlns:metric="http://www.ibm.com/tivoli/netcool/webtop/metrics/7.3.1">
 <method methodName="metric.createMetric">
 <metric:metric name="metricsample1"
 displayName="MetricSample1"
 description="Shows count of all Critical events. Created using WAAPI."
 units="events"
 maxValue="10000"
 minValue="0"
 threshold1="30">
 </metric:metric>
 </method>
</methodCall>
```

```

 threshold2="70">
<metric:command type="sql" mode="basic">
 <metric:text selectField="sum(Tally)" whereClause="Severity=5"
 databaseName="alerts" tableName="status"/>
</metric:command>
</metric:metric>
</method>
</methodCall>

```

## Replace or create a metric

The following WAAPI command creates or replaces a metric named `metricsample2` that shows the count of all major events in the `ObjectServer`.

```

<methodCall xmlns:metric="http://www.ibm.com/tivoli/netcool/webtop/metrics/7.3.1">
<method methodName="metric.createOrReplaceMetric">
<metric:metric name="metricsample2"
 displayName="MetricSample2"
 description="Shows count of all Major events. Created using WAAPI."
 units="events"
 maxValue="100"
 minValue="0"
 threshold1="40"
 threshold2="80">
<metric:command type="sql">
<metric:text data="select sum(Tally) from alerts.status where Severity=4;"/>
</metric:command>
</metric:metric>
</method>
</methodCall>

```

To ensure any restriction filters are applied, use the same form of the `<metric:text>` element as shown in the basic mode example of creating a metric. The sample file `samplerrequest_metric_basic.xml` has an example of creating and replacing a metric in basic mode.

## Modify a metric

The following WAAPI command makes three modifications to the metric named `metricsample1`:

- A new maximum value
- New thresholds for the transitions between the low, medium, and high ranges of the metric
- A modified description

```

<methodCall xmlns:metric="http://www.ibm.com/tivoli/netcool/webtop/metrics/7.3.1">
<method methodName="metric.modifyMetric">
<metric:metric name="metricsample1"
 displayName="MetricSample1"
 description="Shows count of all Critical events. Modified using WAAPI."
 units="events"
 maxValue="250"
 minValue="0"
 threshold1="40"
 threshold2="90">
 <metric:command type="sql">
 <metric:text data="select sum(Tally) from alerts.status where Severity=5;"/>
 </metric:command>
</metric:metric>
</method>
</methodCall>

```

To ensure any restriction filters are applied, use the same form of the <metric:text> element as shown in the basic mode example of creating a metric. The sample file `samplerquest_metric_basic.xml` has an example of modifying a metric in basic mode.

### Delete a metric

The following WAAPI command deletes the metric named `metricsample2`.

```
<methodCall xmlns:metric="http://www.ibm.com/tivoli/netcool/webtop/metrics/7.3.1">
 <method methodName="metric.deleteMetric">
 <metric:metric name="metricsample2"/>
 </method>
</methodCall>
```

### List the existing metrics

The following WAAPI command returns a list of the available metrics.

```
<methodCall xmlns:metric="http://www.ibm.com/tivoli/netcool/webtop/metrics/7.3.1">
 <method methodName="metric.getList" />
</methodCall>
```

## Setting the thresholds for a gauge

A gauge has three display levels: low, medium, and high. This task shows how to set the thresholds for each display level.

### Before you begin

First note the following for each gauge you want to modify:

- The name of the gauge  
The name appears beneath each gauge on the Gauges page.
- The value of the thresholds for transitions from low to medium and from medium to high.

You express each threshold as a percentage of the gauge's range (maximum value minus minimum value). Use the following method to determine the percentage value of each threshold:

1. Decide on the actual values on the gauge that are to be the threshold points.
2. Calculate the percentage equivalent of each point as follows:

$$\text{threshold(\%)} = (\text{threshold} - Gmin) / (Gmax - Gmin) * 100$$

where:

*threshold(%)*

is the threshold expressed as a percentage.

*threshold*

is the threshold point on the gauge.

*Gmin* is the minimum value of the gauge.

*Gmax* is the maximum value of the gauge.

To change the thresholds for one or more gauges:

1. Log in to the Web GUI server using a command-line interface.
2. Go to the directory: `webgui_home_dir/etc/configstore/ncwMetrics`.
3. Open the XML file for a gauge using a text editor. The name of the file is based on the name of the gauge on the Web GUI.

4. Locate the **<metric>** element and:
  - a. Set the value of the threshold1 attribute to the percentage value for the low to medium transition.
  - b. Set the value of the threshold2 attribute to the percentage value for the medium to high transition.
5. Save the file.
6. Repeat steps 3 on page 368 to 5 for the other gauges you want to change.
7. Exit from the text editor.
8. Depending on the configuration of your site, you might need to restart the server.

## Example

The characteristics and requirements for a gauge are:

- The gauge has a minimum value of 5 and a maximum of 55.
- The threshold points are 10 and 50.

The thresholds, expressed as percentages, are:

- Lower threshold:  

$$(10 - 5) / (55 - 5) * 100 = 10\%$$
- Upper threshold:  

$$(50 - 5) / (55 - 5) * 100 = 90\%$$

The **<metric>** element for this gauge is:

```
<metric name="adminconnections"
 displayName="Administrator"
 displayNameKey="ncw.metric.adminconnections.displayName"
 description="Number of Administration client connections"
 descriptionKey="ncw.metric.adminconnections.description"
 units="clients"
 unitsKey="ncw.metric.adminconnections.units"
 maxValue="55"
 minValue="5"
 threshold1="10"
 threshold2="90">
```

## Related concepts

“The Web GUI in a load balancing environment” on page 93

## Related tasks

“Restarting the server” on page 1





---

## Chapter 7. Monitoring events in the Web GUI

You can monitor and manage Tivoli Netcool/OMNIBus by using the Web GUI.

---

### Monitoring events in the AEL

Use the Active Event List (AEL) to interactively monitor and manage events.

#### **Related concepts**

“Event Dashboard overview” on page 305

#### **Related tasks**

“Administering users, roles, and groups” on page 65

### Event management in Active Event Lists

The Active Event List (AEL) is an interactive table that displays network alert data from an ObjectServer. Use the AEL to monitor alert data and manage information relating to faults in your network.

Interactive alert management is characterized by two-way communication with the ObjectServer. Alerts are viewed by operators, and are then addressed. When connected to the Web GUI, all interactive alert management activities take place from within the AEL window. From here you can monitor, prioritize, and address network alerts.

To use SQL tools in an AEL, you must be ObjectServer-validated with appropriate privileges on the corresponding ObjectServer.

An alert is created when the ObjectServer receives an event, alarm, message, or data item. Each alert is made up of several fields from a particular row in the ObjectServer alerts.status table.

The manner in which alert data is displayed in the AEL is controlled by predefined *filters* and *views*. You can use filters to choose which rows from the ObjectServer alerts.status table to display, and you can use views to choose which fields within a row to display.

Read-write users have access to Filter Builder and View Builder components. These tools can be used to apply defined filters and views to the AEL, either other predefined filters and views, or user-defined filters and views. Administrators can provide all read-write clients with publicly-accessible filters and views for them to use if they require.

After alert information is presented within the AEL display area, you can conduct various administrative actions on the table entries. Network event information can be examined, and alerts acknowledged and assigned to the appropriate network management personnel. In addition, journals attached to alerts can be modified to record these actions, and administrative tools within the AEL can be run based on alert data.

## Accessing Active Event Lists

You can access AELs in several ways to monitor and manage event data.

To access an AEL:

- Open the AEL in the navigation. To open the default AEL portlet, click **Availability > Events > Active Event List**.
- Open the AEL from an Event Dashboard portlet by clicking the distribution indicator of a monitor box. To open the default Event Dashboard portlet, click **Availability > Events > Event Dashboard**. AEL opening must be specified as a single-click action in the Event Dashboard portlet preferences.
- Open the AEL from a linked active object in a map.
- Click a hyperlink containing a correctly-constructed AEL query string.

### Related tasks

"Visualizing event information on maps" on page 313

"Setting Event Dashboard portlet preferences and defaults" on page 228

### Related reference

Appendix F, "URLs for opening Web GUI pages," on page 415

"insert:AEL command" on page 402

## Acknowledging and deacknowledging events

You can acknowledge and deacknowledge events in the event list.

**Note:** You can acknowledge and deacknowledge only the events that are assigned to you, your group, or the nobody user.

To acknowledge an alert:

1. Open an AEL and select the events you want to acknowledge.
2. Select **Alerts > Acknowledge** or use Ctrl+A.

**Note:** If configured by your administrator, you are required to enter a journal entry whenever you acknowledge an event.

### Results

You can also deacknowledge a previously acknowledged event:

1. Open an AEL and select an event that has already been acknowledged.
2. Select **Alerts > Deacknowledge** or use Ctrl+D.

### Related concepts

"Prompt types" on page 253

## Assigning ownership to events

When an event arrives from a probe, the event is owned by the nobody user unless ownership of the event has been assigned to a specific user. Web GUI users with appropriate privileges can take ownership of events throughout the problem lifecycle, or assign events to a specific user or group.

The OwnerUID and OwnerGID columns provide this capability to work as an individual or as a member of a team. The OwnerUID column holds the user ID of the owner of the event, and the OwnerGID column holds the group ID.

As soon as an event is owned by a user, only that user or a user with higher privileges can modify the event. If an event is assigned to a group, the OwnerUID is reassigned to the nobody user. After an event is assigned to a group, only a member of that group or a user with higher privileges can modify the event.

**Note:** The users or groups to which you can assign events depends on your security permissions. If you are a normal user, you can assign only the events assigned to you, your group, or the nobody user.

To assign ownership of events:

1. From the event list, select one or more events.
2. Assign the events to another user, another group, or yourself, as follows:
  - To assign selected events to a particular user, click **Alerts > User Assign**, and then select a user from the submenu.

**Note:** If there are a large number of names to choose from, the menu structure may include submenus that contain names within alphabetical ranges (for example, A to G, H to M, and so on). If the name you want is grouped in a submenu (for example, A to G), open the submenu, then select the user's name.

- To assign selected events to another group, click **Alerts > Group Assign**, and then select a group from the submenu (for example, **Network Support**).

**Note:** When an event is assigned to a group, the OwnerUID does not change, meaning that it is not reassigned to the nobody user.

- To assign selected events to yourself, click **Alerts > Take ownership**. You take ownership of an event when it has been assigned to you for action, or if you are in a position to resolve it.

## Changing the event information displayed

You can set what event information is displayed from the available event data by editing the event list view, or by selecting and applying a different view. You can also edit the filter criteria used by the current event list, or select a different filter to apply to the event list.

From the event list, perform any of the following actions:

- To edit the current view within the current event list and change the columns displayed from the available data, click **Edit > Views**. The View Builder opens, which you can use to edit the view.
- To select a different view to apply to the event list, click the drop-down list of views on the toolbar and select an item from the list. After being selected, the event list columns update according to the view settings.

- To edit the current filter, click **Edit > Filters**. The Filter Builder opens, which you can use to edit the filter.
- To select a different filter to apply to the event list, select a filter from the Filter list. After being selected, the event list rows update with the filter settings.

#### Related tasks

“Setting up filters for event data” on page 286

“Setting up views for event lists” on page 298

## Copying data from the event list

You can copy event data from the event list to the clipboard for use in other applications.

To copy event data:

1. From the event list, select an event field.
2. Click **Edit > Copy** or press Ctrl+C.
3. Paste this information into another application as required.

#### Results

**Tip:** You can also copy alert information from the Event Information window.

#### Related tasks

“Displaying event information in full”

## Deleting events

You can remove events from the event list.

To delete one or more events in the event list, select the events, and click **Alerts > Delete**.

**Note:** Any user with access to SQL tools can access the Delete tool. However, your administrator can restrict the use of the **Delete** function by assigning you to a group that has no access.

#### Related concepts

“Access criteria for tools” on page 253

## Displaying event information in full

From the event list, you can view full details of any selected events. Event information that is stored in the alerts.status, alerts.details, and alerts.journal database tables is shown.

Access to the information in the tabs is determined on an individual user basis. For a non-administrative user to be able to access the information in the **Fields** tab, the permissions for **Show basic event information** must be selected in their user profile.

For a non-administrative user to be able to view the **Detail** tab, the permissions for both the **Show basic event information** and **Show event details** must be selected in their user profile.

For a non-administrative user to be able to view the **Journal** tab, the permissions for both the **Show basic event information** and **Show journals** must be selected in their user profile.

1. From the event list, perform one of the following actions:
  - To view information for a single event, double-click the event, or select the event and then click **Alerts > Information**.
  - To view information for multiple events, select the events and then click **Alerts > Information**.

The Event Information window opens.

2. Use this window as follows:

**Fields** Click this tab to view a list of all the columns and their corresponding values for a selected event. This information is stored in the ObjectServer alerts.status table. You can click any column in the list to see the complete text for its value in the field below the list of columns.

**Details** Click this tab to view alert details that are stored in the ObjectServer alerts.details table.

**Journal** Click this tab to view the journal entries for the event, as stored in the ObjectServer alerts.journal table.

You can enter additional journal entries by clicking **Add To Journal** to open the full Journal window for the event. On completion, close this window to return to the Event Information window.

**Previous** If you selected multiple events from the event list, click this button to display detailed information for the previous event in your selection. This action can fail if events have been deleted elsewhere in the system.

**Next** If you selected multiple events from the event list, click this button to display detailed information for the next event in your selection. This action can fail if events have been deleted elsewhere in the system.

**Close** Click this button to close this window.

## Results

### Related tasks

“Modifying the preferences of a Web GUI user” on page 70

“Maintaining a journal for an event” on page 376

## Freezing the event data

You can freeze any open event list so that updates from the ObjectServer are not displayed.

Freezing the alerts enables you to take a snapshot of the alert status before it is changed by further updates from the ObjectServer. Although alert freezing does not prevent you from making changes using the tools in the **Alerts** menu, the updates conferred are not displayed until the display is unfrozen.

To freeze the contents of an AEL, click **View > Freeze** or press Ctrl-Z. A check is displayed against the **Freeze** menu item to indicate that the option has been selected.

To restart automatic updates from the ObjectServer, click **View > Freeze** or press Ctrl-Z.

You can also freeze all open event lists and take snapshots of any AELs you have open on your browser.

To freeze the contents of all AELs open in your browser, click **View > Freeze All** or press Ctrl+Shift+Z. A check is displayed against the **Freeze All** menu item to indicate that the option has been selected.

To restart automatic updates from the ObjectServer, click **View > Freeze All** or press Ctrl+Shift+Z.

## Maintaining a journal for an event

You can add and save your own event history information. You can maintain a journal for any event in the event list.

1. From the event list, select the event, and then click **Alerts > Journal**. The Journal window opens.

**Tip:** You can also access the Journal window while within the Event Information window for a selected event. From the **Journal** tab, click the **Add to Journal** button.

2. Complete this window as follows:

### *Journal information list boxes*

The upper list box is read-only and displays the existing journal history text. For each entry, the name of the user who entered the information, and the date and time when they entered this information, are shown. You can use the **Alerts** menu while within this window by right-clicking within this list box.

Use the lower list box to add a text entry to the journal. There is a limit of 4096 characters for each journal entry. When you have completed the text entry, click **Apply** to save the text within the upper list box. The new text is saved as the last entry, and your user name, and the date and time, are automatically added.

**Note:** To be able to add text entries to the journal, you must be an ObjectServer user, you must have the Web GUI read-write user role (netcool\_rw) assigned, and you must have the **Edit journals (read write role)** check box selected on the **User Preferences for Tivoli Netcool/OMNIBus Web GUI** page.

### **Apply to All Selected**

Select this check box if you want to add the newly-entered text to all events that are selected in the event list, and not just to the event whose serial number is displayed at the top of the window. To save the text entry to the journal for each selected event, click **Apply**.

**OK** Click this button to save the newly-entered text and close the window.

### **Previous**

If you selected multiple events in the event list, click this button to move to the journal entry for the previous event in your selection.

**Next** If you selected multiple events in the event list, click this button to move to the journal entry for the next event in your selection.

**Apply** Click this button to save newly-entered text to the journal. The Journal

window remains open for further entries. This button is useful if you have selected multiple events and want to add different journal entries for them.

**Close** Click this button to close this window. You are prompted to save any unsaved changes before the window closes.

## Prioritizing events

You can prioritize events in the AEL by changing the event severity. Each event in the event list has an associated severity, which is indicated by the color of the severity in the display.

The following severity states are used to categorize events:

- Critical
- Major
- Minor
- Warning
- Intermediate
- Clear

You can change the severity of events only if you have permission to do so, and you can change only the severity of events assigned to you, your group, or the nobody user.

To change the priority of an event:

1. Open an AEL.
2. Select the events for which you want to change priority.
3. Click **Alerts > Prioritize** and select a priority setting from the submenu.

The status line displays the number of events that have had their severity set, and the number of events whose severity could not be set because they are not assigned to you.

## Refreshing the event data

The event list refreshes automatically at regular intervals to show all incoming alerts from the ObjectServer. You can choose to refresh the event list manually between the configured intervals to view all the latest alerts at the current point in time.

To refresh the AEL manually between automatic refresh updates, click **View > Refresh**, or use Ctrl-R.

**Note:** In order to enable manual refreshing of the AEL, you must turn off row data caching.

### Related tasks

“Changing the AEL refresh rate” on page 241

“Turning data row caching on or off” on page 242

## Running predictive eventing tools in the Web GUI

You can use the Active Event List (AEL) to monitor predictive events generated within IBM Tivoli Monitoring if your installation has been set up to support predictive eventing.

### Before you begin

Before you can run predictive eventing tools, the following prerequisites must be met:

- IBM Tivoli Monitoring V6.2.2 must be installed and configured for predictive eventing
- Tivoli Netcool/OMNIBus must be configured for predictive eventing.
- The Web GUI setup for predictive eventing must be configured.
- If you want to launch from the AEL directly into Tivoli Enterprise Portal without having to log in, single sign-on must be configured.

For more information about configuring single sign-on and configuring the setup for predictive eventing, see the *IBM Tivoli Netcool/OMNIBus Installation and Deployment Guide*.

A default filter and view are provided for use with predictive events:

- The default filter contains the following SQL WHERE clause: where Class = 89300.
- The default view contains the following fields, which are displayed in the following order, from left to right: Node, TrendDirection, Summary, FirstOccurrence, LastOccurrence, Count, PredictionTime. The sorting order of the view is as follows:
  - The Severity column is sorted in descending order
  - The LastOccurrence column is sorted in ascending order
  - The PredictionTime column is sorted in ascending order

In an AEL, you can restrict the events displayed to predictive events by selecting **PredictiveEvents** from the **Filters** list. To display the predictive eventing fields, select **PredictiveEventsView** from the **Views** list. Alternatively, you can sort the AEL by using either the **Summary** column or the **Class** column. The summary text for predictive events is always prefixed with Prediction from. The default entry in the **Class** column for a predictive event is always Predictive Event, although this value can be changed by an administrator.

**Restriction:** You can run predictive eventing tools against only one event at a time.

To run the predictive eventing tools in the AEL:

- For all events: To display predictive events that meet a certain level of confidence:
  1. Right click any event and click **Alerts > Show Prediction Confidence**.
  2. In the Internal Command Parameters window, select an operator and type a value between 0 and 100. The confidence level is expressed as a percentage. The default operator and confidence level are >80.

A new AEL window opens displaying predictive events that match the specified confidence threshold.

- For predictive events only: To display critical threshold events, right-click a predictive event and click **Alerts > Show Predictive Events Threshold**.



A new AEL window opens displaying all critical threshold events.

- For predictive events only: To display extended predictive attributes, right-click a predictive event and click **Alerts > Show Extended Attributes**.

A new window opens displaying the following extended attributes of the predictive event in table form, for example:

- Confidence
- Number of samples
- Strength

- For predictive events only: To show the details of a predictive event in the default workspace of Tivoli Enterprise Portal:

1. Right click a predictive event and click **Alerts > Show Details in TEP**
2. Optional: If single sign-on has not be configured, log into Tivoli Enterprise Portal

## Monitoring TADDM events in the Web GUI

TADDM events are generated within IBM Tivoli Application Dependency Discovery Manager when a configuration change is detected in your IT environment. You can monitor TADDM events from the Web GUI if your system is configured to support this feature.

A **TADDM** filter is available in your Active Event List for filtering TADDM events; to use this filter, select **TADDM** from the **Filters** list. You can also identify TADDM events in the Active Event List by sorting on the Class column, if this column is available in your current view. By default, the entry is Tivoli Application Dependency Discovery Manager, although this can be changed by your system administrator.

TADDM events have an Indeterminate severity level by default.

From the Active Event List, you can launch across to the TADDM Java console or Web client to view further details about the configuration items for which alerts have been raised:

- To view all the attribute details for a configuration item from within the TADDM Java console, right-click a TADDM event in the Active Event List, and then click **TADDM > Config Item Details (console)** from the pop-up menu.
- To view all the attribute details for a configuration item from within the TADDM Web client, right-click a TADDM event in the Active Event List, and then click **TADDM > Config Item Details (web)** from the pop-up menu.
- To view a change history report for a configuration item from within the TADDM Java console, right-click a TADDM event in the Active Event List, and then click **TADDM > Change History (console)** from the pop-up menu.
- To view a change history report for a configuration item from within the TADDM Web client, right-click a TADDM event in the Active Event List, and then click **TADDM > Change History (web)** from the pop-up menu.

You can double-click an event to view its complete set of details in the Event Information window.

## Searching for event list data

You can search for event list data in a number of ways. You can enter specific text to search for, use a filtering facility to quickly find matching occurrences of data, and filter event list data by severity.

### Filtering events by severity

You can filter the event list data to display only those events that match a particular severity.

To filter by a particular severity, click the required severity color on the event list distribution status bar.

For example, to see only those events with a severity level of minor, click the yellow button on the distribution status bar. The event list refreshes to display only alerts with a severity level of minor.

To remove severity filtering and restore the event list to its original view of all events, click the **All Events** button on the distribution status bar.

### Quickly filtering events

You can use the quick filtering facility as a fast way of displaying events in the event list that match a selected criteria. You can filter for event data and display events that correspond to the value of a specific cell. For example, you can quickly display only those events that occurred at the same time as the selected event, or before the selected event.

To use the quick filter:

1. From the event list, select a cell that contains a value on which to base the quick filter.
2. Click **Alerts > Quick Filter**, and then select one of the following options from the submenu:
  - **Equals** shows all rows with the same field value as the selected cell.
  - **Not Equals** shows all rows with a different field value from the selected cell.
  - **Greater Than** shows all rows with a greater field value than selected cell.
  - **Greater Than or Equals** shows all rows with a field value greater than or equal to the selected cell.
  - **Less Than** shows all rows with a lesser field value than selected cell.
  - **Less Than or Equals** shows all rows with a field value less than or equal to the selected cell.
  - **Like** shows all rows that contain the same string as the selected cell.
  - **Not Like** shows all rows that do not contain the same string as the selected cell.

The event list refreshes to display only those alerts that match the specified filter criteria.

3. To remove quick filtering and restore the event list to its original view of all events, click **Alerts > Quick Filter > Off**. Alternatively, click the **All Events** button on the distribution status bar.

## Searching for text in the event list

You can search the event list for event data that matches a specific value that you enter. You can search for data within a specific column and specify options to narrow down the search.

1. From the event list, click **View > Find**. The Find window opens.
2. Complete this window as follows:

### Column

Select the column to search.

**Value** Type the search value that you want to match. You can enter an exact value to search for or a regular expression.

### Options

Specify the type of match required by selecting one of the following:

- **Exact Match** to find rows where the data in the selected column exactly matches the specified search value.
- **Regular Expression** to find rows where the data in the selected column matches the specified regular expression.
- **Sub String** to find rows where the data in the selected column contains the specified value somewhere within it.

**Find** Click this button to find the first matching occurrence. If a matching row is found in the event list, any currently-selected rows are deselected, and the matching row is selected. The Find window remains open so that you can view any additional matching occurrences.

**Next** Click this button to show the next match, and subsequent matches, in the event list.

**Close** Click this button to close this window.

## Selecting and deselecting events

To work with one or more event in the AEL, you must first select the events. You can then use the options in the **Alerts** menu to manipulate the events. You can also deselect one or more events from a list of selected events.

**Note:** You can right-click an event to access an **Alerts** pop-up menu. The **Alerts** menu is configured by your system administrator.

To select one or more events:

1. Open an AEL.
2. Select single or multiple events in the following ways:
  - To select one event and deselect all other events, click the event row.
  - To select multiple, nonsequential events, hold down the Ctrl key and click each event.
  - To select multiple, sequential events, click an event row to be the first selected. Then hold down the Shift key and then click the last event row to select all in between on the list.
  - To select all events, click **Edit > Select all** or use Ctrl+S.
  - To select all events that match the value in a particular field (smart matching) hold down the Shift key (Ctrl+Alt on Windows) and click the right mouse button over the field you want to match.

## Results

You can also deselect one or more events:

1. Open an AEL.
2. Deselect single or multiple events in the following ways:
  - To deselect all events in the event list, click **Edit > Deselect all** or use Ctrl+E.
  - To deselect an event in a selection of multiple rows, hold down the Ctrl key and click the event.

## Sorting columns

To organize the data displayed in the Active Event List (AEL), use the sorting functions.

Sorted columns are denoted by an upwards-pointing arrow or downwards-pointing arrow in the column header, depending on whether the column is sorted in ascending or descending order.

The data is sorted only by the fields contained in the alerts.status table; conversions are not sorted.

To sort columns:

1. Open an AEL. If you open the AEL with a view in which a sorting order is specified, the sorting order is applied, but not indicated on the column headers.
2. Sort single columns as follows:
  - a. To sort a column, click the column header once. The rows are sorted in ascending order.
  - b. To sort in descending order, click the column header again.
  - c. To unsort the column, click the column header a third time.
3. Sort multiple columns as follows:
  - a. To sort multiple columns, press Ctrl and click the required column headers. The sorting importance of the columns is indicated in square brackets ([ ]) in the column header.
  - b. To alternate the sorting of individual columns within the selection between ascending and descending order, keep Ctrl pressed and click the column headers.
  - c. To unsort the columns, release Ctrl and click any header from among the sorted columns. The previously-sorted columns are unsorted; the column that you clicked is sorted in ascending order.
  - d. To unsort the column that you clicked in step 3c, click the column another two times.

## What to do next

If you opened the AEL with a view, after all columns are unsorted, the sorting order specified in the view is reapplied.

### Related tasks

“Creating views” on page 298

“Changing the sort order in a column” on page 303

## Using tools to manage events

Use the tools in the event list to run SQL commands against events from one or more data sources, or to run external commands, for example, a local application batch file or script. Default tools include the Ping tool, Telnet tool, and Tracepath tool.

### Before you begin

If you want to run a tool against events from more than one data source, note the following criteria:

- The tool must be valid against the ObjectServer from which the events originate. If you select events from multiple ObjectServers, the tool must be valid against all the ObjectServers. For example, if the tool is configured to run against fields that are not contained in one ObjectServer, the tool cannot be run against the entire selection of events.
- You must have write permission against all the ObjectServers from which the selected events originate. If you select events from multiple ObjectServers, and you do not have write permission in all the ObjectServers, the tool runs against only the ObjectServers that you are permitted to modify.
- If you run a tool containing a prompt against events from multiple ObjectServers, the prompt permits you to select only fields or field values that are common to all ObjectServers.

To run a tool:

1. Select one or more events in the AEL and right-click the selection.
2. Run a tool against the selected event or events by selecting the appropriate tool in the **Tools** menu.

---

## Monitoring events in the LEL

You can use the Lightweight Event List (LEL) to monitor and manage alerts.

The JavaScript LEL provides users with a similar list of events as the AEL. However, the LEL does not possess a SmartPage tag and therefore cannot be embedded into a Web page. While you can access all the event information stored in the alerts.status, alerts.details, and alerts.journal database tables, the LEL does not provide you with the additional AEL functionality accessed through the AEL menus. In addition, the LEL displays alerts from the default data source only.

You can use any of the following methods to access an LEL:

- Open the LEL from the navigation. To open the default LEL, click **Availability > Events > Lightweight Event List (LEL)**
- Launch an LEL from a linked active object in a map.
- Click a hyperlink containing a correctly-constructed LEL query-string.

## Viewing events in the Lightweight Event List

Use the functions of the Lightweight Event List (LEL) to control what events are displayed and how the event information is displayed.

The number of alert rows in each page is determined by the `lel.pagesize.default` property in the `server.init` file. The default is 500.

To view event data in the LEL:

1. Open an LEL.
2. Use the Lightweight Event List (LEL) window as follows:

### **Refresh event data**

Manually refreshes the LEL display area.

The LEL is refreshed automatically at predefined intervals to show all incoming alerts from the ObjectServer.

### **Freeze updates to this window**

Freezes the contents of the LEL display area so no further updates take place.

### **Select/deselect all rows**

Selects or deselects all rows in the LEL display area.

### **Page Slider Buttons**

Use the page slider to step backwards or forwards through the alert table pages, skip to the first or last alerts, and move to a specific page displaying a block of alerts (the screen page range is constrained to blocks of five).

## Displaying full event information in the Lightweight Event List

From the Lightweight Event List (LEL) list, you can view full details of any selected event. All information stored in the following database tables is shown: `alerts.status`, `alerts.details`, and `alerts.journal`.

To display full information about one or more events:

1. From the LEL, double-click an event row. The Event Information window opens and the **Fields** tab is displayed by default.
2. Use this window as follows:

**Fields** Click this tab to view a list of all the columns and their corresponding values for a selected event. This information is stored in the ObjectServer `alerts.status` table.

### **Details**

Click this tab to view alert details that are stored in the ObjectServer `alerts.details` table.

### **Journal**

Click this tab to view the journal entries for the event, as stored in the ObjectServer `alerts.journal` table.

### **Previous**

If you selected multiple events from the event list, click this button to display detailed information for the previous event in your selection. This action can fail if events have been deleted elsewhere in the system.

**Next** If you selected multiple events from the event list, click this button to

display detailed information for the next event in your selection. This action can fail if events have been deleted elsewhere in the system.

**Close** Click this button to close this window.

---

## Monitoring events in the Table View

You can view alert data in a tabular format in the Table View. The tables are static, and they present all available events from the default data source. The Table View can be embedded in a Web page using a SmartPage tag.

You can use any of the following methods to view events in a static table format:

- Open a page containing a Table View. To open the default Table View, click **Availability > Events > Table View**.
- Open the Table View from a linked active object in a map.
- Click a hyperlink that contains a Table View query string

### Related reference

Appendix D, “SmartPage commands and templates,” on page 401

“insert:TABLEVIEW command” on page 408

Appendix F, “URLs for opening Web GUI pages,” on page 415





---

## Appendix A. Accessibility features for the Web GUI

Accessibility features help users who have a disability, such as restricted mobility or limited vision, to use information technology products successfully.

The following list includes the major accessibility features in the Tivoli Netcool/OMNIBus Web GUI:

- The Active Event List (AEL) supports keyboard-only operation.
- The Web GUI administrative screens are readable by screen readers.

### Keyboard navigation

The navigation functions of the Web GUI can be accessed using the keyboard.

This product uses standard Microsoft Windows navigation keys.

The AEL and the Java Map Editor menu functions can also be accessed using accelerator keys. Accelerator keys or mnemonics refer to the underlined characters in menus and dialogs such as F for File.

### Alternative text

All non-text content used in the Web GUI has associated alternative text.



---

## Appendix B. Data source configuration file data reference

Web GUI communication configuration files must conform in structure to the content described by a Document Type Definition (DTD).

---

### XML overview

The Extensible Markup Language (XML) is a standard, self-describing set of rules for structuring data so that it can be processed and exchanged across a variety of hardware types, operating systems, and applications.

A reasonable degree of XML knowledge on the part of the Web GUI administrator is assumed.

To write your own data source configuration files you must understand:

- The rules, logic, and components used by XML
- The concepts of *elements*, *attributes*, and *markup*
- How to create documents that are well-formed, and valid against an XML Document Type Definition (DTD)

---

### DTD reference

XML is hierarchical in structure and the DTD specifies whether each element permits *child elements*, that is, whether other elements can be used under an element within the hierarchy.

**Attention:** XML is case-sensitive. Elements, attributes, and attribute values used within an XML command file must be used exactly as they are shown in the DTD.

#### Data types and legends

The data types and legends that accompany the Web GUI DTD elements and attributes are as follows:

**NM** Indicates that the attribute types are names consisting of XML NMTOKEN character (letters, periods, numbers, underscores, dashes, and colons). NM often also indicates that the attribute contains a list of predefined choices.

**CDATA**

Indicates that the attribute contains unparsed character data.

**IMP** Indicates that the presence of the attribute is implied (optional).

**REQ** Indicates that the presence of the attribute is required.

#### Attributes and elements of the DTD

The XML elements and attributes defined in the Web GUI configuration DTD as follows:

**<chart>**

This element has the following attributes:

- maxAge (type: CDATA, presence: IMP)
- enabled (type: NM, presence: IMP)

- cleantime (type: CDATA, presence: IMP)

This element has no child elements.

#### **<config>**

This element has the following attributes:

- maxAge (type: CDATA, presence: IMP)

This element has no child elements.

#### **<eventList>**

This element has the following attributes:

- maxAge (type: CDATA, presence: IMP)
- enabled (type: NM, presence: IMP)
- cleantime (type: CDATA, presence: IMP)

This element has no child elements.

#### **<eventSummary>**

This element has the following attributes:

- maxAge (type: CDATA, presence: IMP)
- enabled (type: NM, presence: IMP)
- cleantime (type: CDATA, presence: IMP)

This element has no child elements.

#### **<metric>**

This element has the following attributes:

- maxAge (type: CDATA, presence: IMP)
- enabled (type: NM, presence: IMP)
- cleantime (type: CDATA, presence: IMP)

This element has no child elements.

#### **<ncwBackUpServer>**

This element has no attributes. This element has the following child elements:

- <ncwOSConnection>

#### **ncwConnectionParameters**

This element has no attributes. This element has the following child elements:

- <ncwstatementParameters> There are zero or one occurrences of this element.

#### **<ncwDataSourceCredentials>**

This element has the following attributes:

- userName (type: CCDATA, presence: IMP)
- password (type: CCDATA, presence: IMP)
- encrypted (type: NM, presence: IMP)
- algorithm (type: NM, presence: IMP)

This element has no child elements.

#### **<ncwDataSourceDefinition>**

This element has the following attributes:

- type (type: NM, presence: IMP)
- name (type: CDATA, presence: REQ)

This element has the following child elements:

- <results-cache>
- <ncwDataSourcePollingParameters>
- <ncwConnectionParameters> There are zero or one occurrences of this element.
- <ncwDataSourceCredentials>
- <ncwFailOverPairDefinition>
- <ncwReadCloudDefinition> There are zero or one occurrences of this element.

#### **<ncwDataSourceDefinitions>**

This is the root element and has no attributes. This element has the following child elements:

- <ncwDefaultDataSourceList>
- <ncwDataSourceDefinition> There is at least one occurrence of this element.

#### **<ncwDataSourceEntry>**

This element has the following attributes:

- name (type: CDATA, presence: REQ)

This element has no child elements.

#### **<ncwDataSourcePollingParameters>**

This element has no attributes. This element has the following child elements:

- <ncwFailOverPollingParameters>
- <ncwHeartBeatParameters>

#### **<ncwDefaultDataSourceList>**

This element has no attributes. This element has the following child elements:

- <ncwDataSourceEntry> There is at least one occurrence of this element.

#### **<ncwFailOverPairDefinition>**

This element has no attributes. This element has the following child elements:

- <ncwPrimaryServer>
- <ncwBackUpServer> There are zero or one occurrences of this element.

#### **<ncwFailOverPollingParameters>**

This element has the following attributes:

- backOffMultiplier (type: CDATA, presence: IMP)
- basePollingTime (type: CDATA, presence: IMP)

This element has no child elements.

#### **<ncwHeartBeatParameters>**

This element has the following attributes:

- basePollingTime (type: CDATA, presence: IMP)

This element has no child elements.

#### **<ncwOSConnection>**

This element has the following attributes:

- host (type: CDATA, presence: REQ)
- port (type: CDATA, presence: IMP)

- ssl (type: NM, presence: IMP)
- minPoolSize (type: CDATA, presence: IMP)
- maxPoolSize (type: CDATA, presence: IMP)

#### **<ncwPrimaryServer>**

This element has no attributes. This element has the following child elements:

- <ncwOSConnection>

#### **<ncwQueryTimeout>**

This element has the following attributes:

- baseTime (type: CDATA, presence: IMP)

This element has no child elements.

#### **<ncwReadCloudDefinition>**

This element has no attributes. This element has the following child elements:

- <ncwOSConnection> There is at least one occurrence of this element.

#### **<ncwStatementParameters>**

This element has no attributes. This element has the following child elements:

- <ncwQueryTimeout> There are zero or one occurrences of this element.

#### **<results-cache>**

This element has no attributes. This element has the following child elements:

- <chart>
- <config>
- <eventList>
- <eventSummary>
- <metric>

---

## **Element reference**

The elements used in the Web GUI configuration DTD often have one or more associated attributes, for which a value can be required.

The elements defined within the configuration DTD are as follows.

#### **<chart>**

This element is a child element of the <results-cache> element. This element specifies caching options for chart results. If caching is enabled, the maxAge attribute specifies the expiry time, in seconds, for the cache. The cleantime attribute specifies the time interval, in seconds, at which cache entries are checked and removed. Cache data that exceeds the time imposed by the maxAge attribute is removed.

#### **<config>**

This element is a child element of the <results-cache> element. This element specifies whether data caching is enabled. If caching is enabled, the maxAge attribute specifies the expiry time, in seconds, for the cache. For example:

```
<config maxAge="60" enabled="true">
```

#### **<eventList>**

This element is a child element of the <results-cache> element. This element

specifies caching for results in the event lists. If caching is enabled, the `maxAge` attribute specifies the expiry time, in seconds, for the cache. The `cleantime` attribute specifies the time interval, in seconds, at which cache entries are checked and removed. Cache data that exceeds the time imposed by the `maxAge` attribute is removed.

#### **<eventSummary>**

This element is a child element of the `<results-cache>` element. This element specifies caching for event summary results, such as maps and Event Dashboards. If caching is enabled, the `maxAge` attribute specifies the expiry time, in seconds, for the cache. The `cleantime` attribute specifies the time interval, in seconds, at which cache entries are checked and removed. Cache data that exceeds the time imposed by the `maxAge` attribute is removed.

#### **<metric>**

This element is a child element of the `<results-cache>` element. This element specifies caching for results in Gauges pages. If caching is enabled, the `maxAge` attribute specifies the expiry time, in seconds, for the cache. The `cleantime` attribute specifies the time interval, in seconds, at which cache entries are checked and removed. Cache data that exceeds the time imposed by the `maxAge` attribute is removed.

#### **<ncwBackUpServer>**

This element is a child element of `<ncwDefaultDataSourceList>` and contains the `ncwOSConnection` element specifying host and port of the failover ObjectServer. For example:

```
<ncwBackUpServer>
 <ncwOSConnection
 host="192.168.0.3"
 port="4141"
 />
</ncwBackUpServer>
```

#### **<ncwConnectionParameters>**

This element is a child element of `<ncwDataSourceDefinition>` and contains elements that control the connection to a data source.

#### **<ncwDataSourceCredentials>**

This element is a child element of `<ncwDataSourceDefinition>` and holds the login information required by the Web GUI to access the data source. If the `encrypted` attribute is set to `true`, a password encrypted using the Tivoli Netcool/OMNIBus **nco\_g\_crypt** encryption utility can be used. For example:

```
<ncwDataSourceCredentials
 password=""
 userName="root"
 encrypted="false"
/>
```

#### **<ncwDataSourceDefinition>**

This element is a child element of the `<ncwDataSourceDefinitions>` element and contains the tags that define configuration and communication parameters for an individual data source.

#### **<ncwDataSourceDefinitions>**

This is the root element of the DTD.

#### **<ncwDataSourceEntry>**

This element is a child element of `<ncwDefaultDataSourceList>` and contains the names of the default data sources that communicate with the Web GUI. These entries are subsequently defined in the configuration file by corresponding `<ncwDataSourceDefinition>` tags. The first entry in the list is the

default data source used by the Web GUI for client authentication. If this data source is not present, the next entry in the list is used as a default. For example:

```
<ncwDefaultDataSourceList>
 <ncwDataSourceEntry name="NCOMS"/>
 <ncwDataSourceEntry name="NILKA"/>
</ncwDefaultDataSourceList>
```

#### **<ncwDataSourcePollingParameters>**

This element is a child element of <ncwDataSourceDefinition> and contains the elements that control failover and data source heartbeat polling.

#### **<ncwDefaultDataSourceList>**

See <ncwDataSourceEntry>.

#### **<ncwFailOverPairDefinition>**

This element is a child element of <ncwDataSourceDefinition> and contains the tags that specify the primary and backup ObjectServers. The inclusion of a backup ObjectServer is optional, but only one is permitted per data source. For example:

```
<ncwFailOverPairDefinition>
 <ncwPrimaryServer>
 <ncwOSConnection
 host="192.168.0.7"
 port="4545"
 />
 </ncwPrimaryServer>
 <ncwBackUpServer>
 <ncwOSConnection
 host="192.168.0.8"
 port="4646"
 />
 </ncwBackUpServer>
</ncwFailOverPairDefinition>
```

#### **<ncwFailOverPollingParameters>**

This element specifies the time interval at which the data source is polled in the event of a failover. This element is used only when there is a failover server available, as defined by the <ncwBackUpServer> element. For example:

```
<ncwFailOverPollingParameters backOffMultiplier="2" basePollingTime="10"/>
```

#### **<ncwHeartBeatParameters>**

This element is a child element of <ncwDataSourcePollingParameters> and specifies the time interval, in seconds, for the Web GUI to poll an active data source. For example:

```
<ncwHeartBeatParameters basePollingTime="15"/>
```

#### **<ncwOSConnection>**

This element is a child element of both <ncwPrimaryServer> and <ncwBackUpServer> and specifies the communication criteria for a primary or failover data source. For example:

```
<ncwOSConnection host="192.168.0.3" port="4141"/>
```

#### **<ncwPrimaryServer>**

This element is a child element of <ncwDefaultDataSourceList> and contains the ncwOSConnection element specifying host and port of the primary ObjectServer. For example:



```

<ncwPrimaryServer>
 <ncwOSConnection
 host="192.168.0.3"
 port="4141"
 />
</ncwPrimaryServer>

```

#### <ncwQueryTimeout>

This element is a child element of <ncwStatementParameters> and defines the time out period, in seconds, for SQL statements sent to a data source. For example:

```
<ncwQueryTimeout baseTime="60" />
```

#### <ncwReadCloudDefinition>

This element is a child element of <ncwDataSourceDefinition> and holds the addresses of all the display servers you want to use with this master ObjectServer. One <ncwReadCloudDefinition> element permitted per data source. You cannot have multiple display server clouds communicating with a single master ObjectServer. For example:

```

<ncwReadCloudDefinition>
 <ncwOSConnection
 host="192.168.0.9"
 port="4747"
 />
 <ncwOSConnection
 host="192.168.0.10"
 port="4848"
 />
 <ncwOSConnection
 host="192.168.0.11"
 port="4949"
 />
</ncwReadCloudDefinition>

```

#### <ncwStatementParameters>

This element is a child element of <ncwConnectionParameters> and contains elements that control the exchange of SQL statements with a data source.

#### <results-cache>

The <results-cache> element is a child element of the <ncwDataSourceDefinition> element. It contains the child elements <chart>, <config>, <eventList>, <eventSummary>, and <metric>.

## Attribute reference

Use this information to understand the attributes used in the Web GUI configuration DTD. Some attributes are enumerated and the values of these attributes are constrained to a list of predefined text strings. When enumerated attributes are used within the XML command file, they must use one of the values shown in the list.

The following table describes each attribute defined within the configuration DTD. Default values (if any) are provided in the description.

Table 73. Configuration DTD attribute definitions

Attribute	Constrained values	Description
algorithm	DES   AES	Specifies whether a DES or an AES algorithm is used.

Table 73. Configuration DTD attribute definitions (continued)

Attribute	Constrained values	Description
backOffMultiplier	None	<p>The multiplier for the backoff algorithm used to calculate the polling backoff time during a failover.</p> <p>The default value is 1.</p>
basePollingTime	None	<p>The seed time, in seconds, for the algorithm used to calculate the polling backoff time during a failover.</p> <p>The default value is 20 seconds for the &lt;ncwFailoverPollingParameters&gt; element or 15 seconds for the &lt;ncwHeartbeatParameters&gt; element.</p>
baseTime	None	<p>The timeout period, in seconds, for a query statement sent to the data source. If the Web GUI receives no response within this time, it attempts to reconnect to the data source.</p> <p>The default value is 30 seconds.</p>
cleantime	None	<p>The time interval, in seconds, the Web GUI server waits before checking for how long each user session has been inactive.</p> <p>When this check takes place, cache data that exceeds the time imposed by the maxAge attribute is removed.</p> <p>The default value is 120 seconds for the &lt;chart&gt; and &lt;eventList&gt; elements or 20 seconds for the &lt;eventSummary&gt; and &lt;metric&gt; elements.</p>
enabled	true   false	<p>Specifies if page caching is turned on or off.</p> <p>The default value is true for the &lt;ncwDataSourceDefinition&gt;, &lt;eventSummary&gt;, and &lt;metric&gt; elements or false for the &lt;chart&gt; and &lt;eventList&gt; elements.</p>
encrypted	true   false	<p>Specifies whether the user password is encrypted.</p> <p>The default value is false.</p>
host	None	<p>The host name or IP address of a specified data source.</p>
maxAge	None	<p>The cache expiry time limit in seconds.</p> <p>The default value is 10 seconds for the &lt;eventSummary&gt; and &lt;metric&gt; elements, 60 seconds for the &lt;chart&gt; and &lt;eventList&gt; elements, or 3600 seconds for the &lt;config&gt; element.</p>

Table 73. Configuration DTD attribute definitions (continued)

Attribute	Constrained values	Description
maxPoolSize	Maximum value: 1024	The maximum number of pooled connections to an ObjectServer data source that can exist at any one time.  The default value is 10.
minPoolSize	None	The minimum number of pooled connections to an ObjectServer data source to maintain.  The default value is 5.
name	None	The name given to an ObjectServer data source displayed within the Web GUI during administrative activities.  This value also links each data source definition that is listed at the start of the configuration file to its subsequent definition.
password	None	The password used to log in to the ObjectServer.  The default is a blank password.
port	None	The port number of a specified data source.  The default value is 8080.
ssl	true   false	Specifies whether to use a SSL connection to an ObjectServer.  The default value is false.
type	singleServerOS DataSource   multipleServerOS DataSource	The type of data source configuration required for the specified data source. The required types are as follows:  <b>singleServerOSDataSource</b> Use this type for a single primary data source configuration, or for a backup data source configuration.  <b>multipleServerOSDataSource</b> Use this type for a dual-server desktop configuration.  The default value is singleServerOSDataSource.
userName	None	The user name of the user connecting to the ObjectServer. The user must have root privileges on the ObjectServer.  The default value is root.



---

## Appendix C. Invalid characters in filters, views, and tools

You cannot use certain characters in the names of any Web GUI objects, such as filters, views, or tools. You also cannot create filters, views, or tools with spaces in the name.

The invalid characters are defined in the following file.

`webgui_home_dir/etc/illegalChar.prop`

**Attention:** Only change the invalid characters as directed in the documentation or by IBM Support.

This file contains the following properties:

### **INVALID\_NAME\_CHARS**

This property defines characters that are not permitted in any user-defined names. The default characters listed in this property are as follows:

`$ ! £ % ^ & * ( ) + = - ` ~ # @ ' : ; < > { } [ ] ? / \ \ | , "`

### **INVALID\_NAME\_START\_CHARS**

This property defines characters that are not permitted as the initial character of any user-defined names. The default characters listed in this property are as follows:

`/ \ \ * ? " < > | & .`



---

## Appendix D. SmartPage commands and templates

You can use SmartPage commands to populate Web pages that are served by the Web GUI. You can also use variables in SmartPage commands to dynamically populate predefined template files.

### Related reference

“Map SmartPage administration” on page 406

---

## SmartPage commands overview

SmartPage commands are single-line HTML instructions that can be used to validate user access, and to allow Web GUI components such as applets to be added to a Web page.

You can use SmartPage commands in your Web pages to perform the following tasks:

- Enable a Web page for SmartPage command processing.
- Validate user access to a Web page against a group.
- Insert an AEL applet.
- Insert an AEL applet for each filter in a map.
- Insert a map applet.
- Insert a map applet containing a grid of filters that are associated with a user.
- Insert a hyperlinked list of accessible maps that are associated with a user.
- Insert a Table View.
- Insert the name of the user who is currently logged in.
- Insert the home URL that is associated with a user.

## SmartPage code example

This example shows the source code for a Web page that contains SmartPage commands.

```
<!-- enable:SMARTPAGE -->
<!-- Validate: [*,redirect.html] -->
<html>
 <head>
 <title>SmartPage FilterPage Command</title>
 </head>
 <body>
 <p>
 <center>
 <table border =1>
 <tr><td>
 <!-- insert:FILTERPAGE[bgcolor="gray"] -->
 </td></tr>
 </table>
 </center>
 </p>
 </body>
</html>
```

---

## SmartPage command reference

When you create a Web page, you can include SmartPage commands to dynamically provide a page with the data display components that are generated by the Web GUI.

Take note of the following conventions when using SmartPage commands:

- Do not add any spaces within SmartPage commands.
- To insert an object into a page, you must place the insert command on its own in a line with no other commands or characters on that line.

### enable:Smartpage command

The **enable:Smartpage** command instructs the Web GUI server to use SmartPage commands.

You must enter this command as the first line in the HTML source file of any page that uses other SmartPage commands.

The following example shows the **enable:Smartpage** command.

```
<!-- enable:Smartpage -->
```

### Validate command

The **Validate** command opens a login window before displaying the requested page. The user name is checked against groups associated with the page.

You must enter this command as the second line in the HTML source file.

The following example shows the **Validate** command.

```
<!-- Validate: [group,where.html] -->
```

In this line, *group* is the name of the group to check against, and *where.html* is the alternative page to display if the user does not have permission to open the current page.

### insert:AEL command

The **insert:AEL** command inserts an AEL applet into a Web page.

- “Examples”
- “Parameters” on page 403
- “Compatibility of parameters” on page 404

### Examples

The following example shows a sample **insert:AEL** command in which the AEL is launched with a predefined filter.

```
<!--insert:AEL[filtertype=filtertype,filtername=filtername,metric=metric,
metricof=metricof,showinframe=true|false,width=width,height=height,
transientname=name,debuglevel] -->
```

The following example shows a sample **insert:AEL** command in which the AEL is launched with a transient filter.

```
<!--insert:AEL[sql=sql,transientname=name,showinframe=true|false,cols=numcols,
width=width,height=height,debuglevel] -->
```



## Parameters

The parameters are as follows. If you use a deprecated parameter, an entry is recorded in the following location: *tip\_home\_dir/profiles/TIPProfile/logs/*.

### **cols**

The number of columns. The default value is 4.

### **datasource**

A data source defined in the *ncwDataSourceDefinitions.xml* data source configuration file. If you do not specify this parameter, the default data source is used. For more information about the default data source, see the *IBM Tivoli Netcool/OMNIbus Installation and Deployment Guide*.

**Tip:** If you have defined multiple data sources, you can specify them in the string by using a comma-separated list, as shown in the following example:

```
datasource=datasource1,datasource2
```

### **debuglevel**

Shows debug information in the Java plug-in console. The level can be set from 0 to 9.

### **entity | entities**

Deprecated: Specifies the name of the entity that is associated with the AEL. A single AEL applet is created for this entity. No special layout is provided. The **entities** property specifies an entity list. An applet is created for each entity and presented in a table. Omit either property if you are using the **filter** and **view** properties.

### **entityviewname**

Deprecated: The name of the view that you want to apply to the AEL. This view is defined within the View Builder.

### **filter**

Deprecated: The SQL syntax that is used to create a filter. When the filter string is applied to an AEL, only the rows that meet the criteria set by the filter are displayed. Omit this property if you are using the **entity** property.

### **filtername**

The name of the filter that you want to apply to the AEL. This filter is defined in the Filter Builder. If you do not specify the **filtertype** parameter, the command searches the filter types to find a filter with a matching name.

### **filtertype**

The type the filter that you want to apply to the AEL. Possible values are as follows:

- user
- global
- system
- user\_transient

### **height**

The height of the applets. The default value is 165.

### **metric**

The metric value for a transient filter. This can be any of Average, Count, Sum, Minimum or Maximum.

### **metricof**

The metric value for a transient filter. This is a field type; for example Tally.

**showinframe**

If true, a monitor box is displayed and the AEL window opens. If false, the AEL applet is embedded in a Web page. The default is true.

**sql**

The SQL syntax that is used to create a transient filter. Do not use this parameter in combination with the deprecated **entity** or **filter** parameters. If you do not specify the **transientname** parameter, the text entered in the **sql** is also used as the filter name in the AEL.

**transientname**

The transient filter name. Requires the **filter** parameter, which is deprecated, or the **sql** parameter. This is the name of the transient filter and is used as the title for the associated AEL monitor box.

**view**

The view that is applied to the AEL, as defined in the View Builder. When a view is applied to an AEL, only the columns that are contained within the view are displayed. Omit this property if you are using the deprecated **entity** or **entities** properties.

**width**

The width of the applets. The default value is 130.

## Compatibility of parameters

The following list shows the compatibility of the parameters, indicating whether a parameter can be used in combination with another parameter. If a parameter is listed for a parameter, then they can be used in combination.

**cols** data source; entities; height; monitortitle; showinframe; width

**datasource**

cols; entities; entity; filter; filename; height; metric; metricof; monitortitle; showinframe; view; viewname; width

**entities**

Deprecated: cols; data; source; height; monitortitle; showinframe; width

**entity** Deprecated: data source; height; monitortitle; showinframe; width

**entityviewname**

data source; filter; filename; height; monitortitle; showinframe; width

**filter** data source; height; metric; metricof; monitortitle; showinframe; view; viewname; width

**filename**

data source; filtertype height; monitortitle; showinframe; view; viewname; width

**filtertype**

data source; filename height; monitortitle; showinframe; view; viewname; width

**height** cols; data source; entities; entity; filter; filename; metric; metricof; monitortitle; showinframe; view; viewname; width

**metric** data source; filter; height; metricof; monitortitle; showinframe; view; viewname; width

**metricof**

data source; filter; height; metric; monitortitle; showinframe; view;  
viewname; width

**monitortitle**

cols; data source; entities; entity; filter; filtername; height; metric; metricof;  
showinframe; view; viewname; width

**showinframe**

cols; data source; entities; entity; filter; filtername; height; metric; metricof;  
monitortitle; view; viewname; width

**sql**

data source; transientname; height; monitortitle; showinframe; view;  
viewname; width

**transientname**

data source; sql; height; monitortitle; showinframe; view; viewname; width

**view**

data source; filter; filtername; height; monitortitle; showinframe; width

**width**

cols; data source; entities; entity; filter; filtername; height; metric; metricof;  
monitortitle; showinframe; view; viewname; width

## **insert:AELMAPVIEW command**

The **insert:AELMAPVIEW** command inserts an AEL applet for each filter in a map, into a Web page.

```
<!-- insert:AELMAPVIEW[map=map,showinframe=showinframe,cols=cols,width=width,
height=height,debuglevel=debuglevel] -->
```

The properties used in this command are as follows:

**cols**

The number of columns to use in the table when displaying multiple applets.  
The default value is 4.

**debuglevel**

Shows debug information in the plug-in console. The level can be set from 0 - 9.

**height**

The height of the applets. The default value is 165.

**map**

The map name.

**showinframe**

If true, a monitor box is displayed and the Active Event List (AEL) window opens. If false, the AEL is embedded in a Web page. The default is true.

**width**

The width of the applets. The default value is 130.

## insert:MAPLET command

The **insert:MAPLET** command inserts a map into a Web page.

The following command can be entered in the HTML source file:

```
<!-- insert:MAPLET[map=map,soundurl=soundurl,refresh=n, enabletooltips=true] -->
```

The parameters of this command are as follows:

### **map**

Required: Specifies the name of the map to be displayed.

### **soundurl**

Optional: Specifies the URL of an audio .au file. This file is played when the maximum severity of any of the filters on a map increases. The URL must be specified in the following format:

*protocol://host:port/path/filename.fileextension*

### **refresh**

Optional: Specifies the refresh rate, in seconds, for the map.

### **enabletooltips**

Optional: Enables hover help for active map objects that are associated with a filter. The hover help displays information from the filter.

### **Related reference**

"Map SmartPage administration"

## Map SmartPage administration

SmartPage commands are special extensions to the HTML language that control user validation, and allow the placement of generated components in Tivoli Integrated Portal pages. SmartPage commands are expressed in HTML as comment tags.

To display a map you must create or edit an existing HTML file, add an **insert:MAPLET** SmartPage tag to the content, and upload the HTML file to the Web GUI server.

## Map SmartPage

The following example shows the contents of a simple HTML file.

```
[1] <!-- enable:smartpage -->
[2] <html>
[3] <head>
[4] <title>Demonstration Map</title>
[5] <meta http-equiv="Content-Type" content="text/html;
[6] charset=iso-8859-1">
[7] </head>
[8] <body>
[9] <!-- insert:MAPLET[mapname] -->
[10] </body>
[11] </html>
```

An explanation of the lines in this example is as follows:

**Line 1** This must be the first line of text in any HTML page that contains a SmartPage tag. This line instructs the Web GUI server that this page is enabled for SmartPage commands.

**Lines 2 to 8**

Contain generic HTML heading information and the body tag that indicates the start of the body of the Web page.

**Line 9** Contains the **insert:MAPLET** SmartPage tag that instructs the Web GUI to place a map applet on the Web page. The name of the map you want to display—*mapname* in this example—is enclosed within the square brackets in the tag. The **insert:MAPLET** tag is valid anywhere within the body of an HTML document.

**Lines 10 to 11**

Contain the tags which close the <body> and <html> elements and conclude the file.

**Related reference**

Appendix D, “SmartPage commands and templates,” on page 401

“insert:MAPLET command” on page 406

**insert:FILTERPAGE command**

The **insert:FILTERPAGE** command inserts a map applet into a Web page that displays either all filters that are associated with a read-write user, or specific filters for a read-write user.

Parameters are optional. All user filters are displayed by default. The map arranges rows of monitor boxes, using user preferences where appropriate. The following command can be entered in the HTML source file:

```
<!-- insert:FILTERPAGE[filters="filter1,filter2...",bgcolor="color",bgimage="image",
monitorwidth="width",monitorheight="height",action="action",url="url",
target="target",soundurl="soundurl",datasource="name"] -->
```

The properties used in this command are as follows.

**action**

Specifies the hyperlink action that takes place when a monitor box is clicked. Use any of the following statements to define the action:

- go to link to a URL.
- ael to link to an embedded AEL in a Web page.
- lel to link to an embedded LEL in a Web page.
- table to link to a Table View.

**bgcolor**

Sets the background color of the map.

**bgimage**

Sets the background image of the map. The background image is uploaded using the Resource Manager and is stored as a resource in:

*install\_dir/profiles/TIPProfile/etc/webtop/resources/\_\_common*

**datasource**

Highlights the ObjectServer used.

**filters**

Constrains the map to a set of predefined filters that are specified using a comma-separated list. If you omit this property, all filters that are associated with the read-write user are displayed.

**monitorheight**

Sets the height of the monitor box applets on the map.

**monitorwidth**

Sets the width of the monitor box applets on the map.

**soundurl**

Specifies the URL of an audio .au file that is played when the maximum severity of the filters in a filter page increases. The URL must be specified in the following format:

*protocol://host:port/path/filename.fileextension*

**target**

Specifies where in the browser window a linked target page is displayed when a monitor box is clicked. For example:

- `_self` indicates the same frame as the map, so that the target page replaces the map.
- `_blank` indicates a new Web browser window.
- `_parent` indicates the parent frame set containing the source link.
- `_top` indicates the frame containing the source link.

**url**

The destination URL if you specified `go` as the value for the **action** property.

## insert:USERMAPLIST command

The **insert:USERMAPLIST** command inserts a list of hyperlinked maps, for a specific user, into a Web page. The list displays all the maps on the server that the presently logged-in user can access.

You can insert the name of each map as a hypertext link, which can be clicked to show the map. The list of maps is validated against the ACLs to which the user belongs. The following command can be entered in the HTML source file:

```
<!-- insert:USERMAPLIST[type=type,target=target] -->
```

Where the parameters are as follows:

**type**

Specifies the type of list, which can be unordered or displayed in a table.

**target**

Specifies the target frame.

## insert:TABLEVIEW command

The **insert:TABLEVIEW** command inserts a Table View into a Web page.

```
<!-- insert:TABLEVIEW[filtername/filtertype/datasource/maxrows=n] -->
```

The parameters are as follows. If you use a deprecated parameter, an entry is recorded in the following location: *tip\_home\_dir/profiles/TIPProfile/logs/*.

**datasource**

A data source defined in the *ncwDataSourceDefinitions.xml* data source configuration file. If you do not specify this parameter, the default data source is used. For more information about the default data source, see the *IBM Tivoli Netcool/OMNIBus Installation and Deployment Guide*.

**Tip:** If you have defined multiple data sources, you can specify them in the string by using a comma-separated list, as shown in the following example:

```
datasource=datasource1,datasource2
```

If you specify multiple data sources, make sure that the filter, either a predefined filter or a transient filter, queries fields that are contained in all data sources. If you select a data source that is not defined in the filter, an error message is displayed in the AEL instead of event data.

**entity**

Deprecated: The name of an entity. Use the **filtername** and **filtertype** parameters instead.

**filtername**

The name of a filter. The default view that is associated with the filter is applied to the Table View.

If you do not specify a filter type, the system searches the filter types for a filter with a name that matches the **filtername** parameter.

**filtertype**

The type of filter. Possible values are as follows:

- user
- global
- system
- user\_transient

**maxrows**

Optional: The number of rows returned in the view.

## insert:USERNAME command

The **insert:USERNAME** command inserts the name of the user who is logged in to the Web GUI, into a Web page:

```
<!-- insert:USERNAME[example] -->
```

In this line, *example* is the name of the user who is logged in.

## insert:USERSHOMEURL command

The **insert:USERSHOMEURL** command inserts the Web GUI home page URL for the user who is currently logged in, into a Web page.

You can use this command to create a link to the home page of the user.

The following example shows the **insert:USERSHOMEURL** command.

```
<!-- insert:USERSHOMEURL -->
```

---

## Template overview

The Web GUI supports dynamically-processed generic Web pages called *templates*. Templates contain SmartPage tag instances that are composed of attribute variables rather than hard-coded attribute data entries.

When a template page is displayed, information in the query string of the URL is passed to a corresponding variable placeholder that dynamically populates the SmartPage attribute.

By using templates, you can avoid creating new Web pages or source code entries to accommodate functionally similar page components. Instead, you can create generic pages where the SmartPage component attributes are governed by information that is stored in a hyperlink to the page URL.

Basic templates are supplied with the Web GUI , which you can customize for your own use. The default templates are located in the following directory:

`webgui_home_dir/etc/templates`

## Template example

This example shows one of the default templates, `table.html`, that is supplied with Tivoli Netcool/OMNIBus.

Each line number is displayed in square brackets:

```
<!-- ENABLE:SMARTPAGE -->
<!-- Validate: [all,redirect.html] -->
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>
 <head>
 </head>
 <body>
 <!-- INSERT:TABLEVIEW[<NCO_V type="str" default="">Map_Name</NCO_V>
 /<NCO_V type="str" default="">
System_Filter</NCO_V>] -->
 </body>
</html>
```

The **insert:TABLEVIEW** command cannot include carriage returns.

In this example, the template file contains a SmartPage command for displaying a Table View. The **insert:TABLEVIEW** command has two attributes: **Map\_Name** and **System\_Filter** . Instead of a static data entry for each attribute value, the source contains an `<NCO_V>` element, the attributes of which are as follows.

### **type="type"**

This entry specifies the type of variable data that the `<NCO_V>` tag can receive from the URL. The data types are as follows:

**str** A text string that can include any text except quotation marks.

### **BOOLEAN**

Used for attribute entries such as `showinframe` in the **insert:AEL** SmartPage tag. Permitted values are "true" or "false".

### **number**

Any whole integer. Used for attribute entries such as `monitorwidth` in the **insert:FILTERPAGE** tag.

### **default="value"**

This entry specifies the default variable value that is used if one is not provided by the URL.

In the case of the **insert:TABLEVIEW** command, the attributes are names, and therefore strings. The expected variable data for both `<NCO_V>` instances is set to "str". No default attributes are provided, so both default entries are empty.

The text between the opening and closing `<NCO_V>` tags establishes the variable name that the URL uses to establish a name-value pair. When you specify a variable name, the text must be alphanumeric and multiple words must be separated by an underscore. For example, `Map_Name`. Variable entries must be separated by a comma.

The URL of template pages containing variable data must be in the following format:



`$(SERVER)/path/filename?variable1=variabledata1&variable2=variabledata2`

Where *path* is the path to the template location, *filename* is the name of the template Web page, *variable* is the variable enclosed within the `<NCO_V>` element, and *variabledata* is the string, Boolean, or numeric entry that you want to populate the variable. For example:

`$(SERVER)/Template/table.html?Map_Name=Example_Europe&System_Filter=Example_All`

## Guidelines for using SmartPage templates

When you create a new Web page that is to be used as a template, you must consider a number of guidelines.

These guidelines are as follows:

- You must include a `<!-- ENABLE:SMARTPAGE -->` tag at the top of the template page.
- You must ensure that the template page is valid and opens correctly in a Web browser.
- Choose sections of the page that you want to make variable and use `<NCO_V>` and `</NCO_V>` container elements to specify the required variable.

---

## Creating SmartPage templates

You can create your own template that is either based on a template in the default directory, or that is based on a Web page.

To create a template that is based on a default template:

1. Navigate to the directory that holds the default templates:  
`webgui_home_dir/etc/templates`
2. Copy and paste one of the templates in this directory. For example, `table.html`.
3. Rename the copy, ensuring that you retain the `.html` file extension.
4. Customize the HTML template heading as required.
5. Define the variables that you require by inserting them between the `<NCO_V>` and `</NCO_V>` tag containers.
6. Save the changes.

To create your own template that is based on a Web page:

1. Create a Web page as normal, by using smart tags.
2. Check that the Web page displays correctly without errors.
3. Replace the smart tags with `<NCO_V>` and `</NCO_V>` tags as appropriate.



---

## Appendix E. Web GUI database tables

When operating in a load balancing cluster, a database holds the configuration information for propagating to all nodes in the cluster.

**Attention:** Do not modify the contents of these tables manually as this can adversely affect the operation of the cluster.

---

### OMNIBUS\_WEB\_GUI.CONFIG\_ITEMS table

The OMNIBUS\_WEB\_GUI.CONFIG\_ITEMS table contains the items of Web GUI configuration data to be replicated across a load balancing cluster.

*Table 74. Columns in the OMNIBUS\_WEB\_GUI.CONFIG\_ITEMS table*

Column name	Data type	Description
CONFIG_ITEM_ID	bigint	A unique identifier for this entry in the table. This column is the primary key for the table.
CHECKSUM	char(20)	A checksum for the entry.
KEY	varchar(128)	A key for the entry.
LAST_UPDATED	timestamp	The date and time when this item of data was last updated.
NAMESPACE	varchar(64)	The namespace for this item of data.
PATH	varchar(256)	The path name for this item of data in the Web GUI directory tree.
TYPE	varchar(12)	The type of this data item.
VALUE	blob(10485760)	The content of the data item.

---

### OMNIBUS\_WEB\_GUI.NODES

The OMNIBUS\_WEB\_GUI.NODES table contains details of the nodes that make up the load balancing cluster. There is one entry for each of the nodes in the cluster.

*Table 75. Columns in the OMNIBUS\_WEB\_GUI.NODES table*

Column name	Data type	Description
NODE_ID	bigint	A unique identifier for this entry in the table.
URI	varchar(512)	The URI of a node in the cluster.

---

## OMNIBUS\_WEB\_GUI.NODES\_CONFIG\_ITEMS

The OMNIBUS\_WEB\_GUI.NODES\_CONFIG\_ITEMS table contains information on when each node in the cluster last updated each item of configuration data defined in the OMNIBUS\_WEB\_GUI.CONFIG\_ITEMS table. There is one entry in the table for each combination of an item of configuration data and cluster node.

*Table 76. Columns in the OMNIBUS\_WEB\_GUI.NODE\_CONFIG\_ITEMS table*

Column name	Data type	Description
CONFIG_ITEM_ID	bigint	The identifier for an item of configuration data held in the OMNIBUS_WEB_GUI.CONFIG_ITEMS table.
NODE_ID	bigint	The identifier of a node in the OMNIBUS_WEB_GUI.NODES table.
LAST_UPDATED	timestamp	The date and time when the data item was last updated on the node.

---

## Appendix F. URLs for opening Web GUI pages

Use the URL to open Web GUI portlets and applets from a map, from a link on an HTML page, or through a link generated by script tool or a CGI tool.

You can use URLs to open the following Web GUI pages:

- “Active Event List”
- “Filter Builder” on page 417
- “Lightweight Event List” on page 417
- “Map pages” on page 418
- “Table View” on page 418

The parameters are as follows. In each URL, *context-root* refers to the context root of the Web GUI. This can be configured during installation of the product and its default value is *ibm/console*.

If you use a deprecated parameter, an entry is recorded in the following location:  
*tip\_home\_dir/profiles/TIPProfile/logs/*.

### Active Event List

To open the Active Event List (AEL), use a URL of the following format:

```
protocol://server.domain:portcontext-root/webtop/AELView?filtertype=typeoffilter
&filtername=filtername&viewname=viewname&viewtype=viewtype&
datasource=datasourcename
```

To use a transient filter, use a URL of the following format:

```
protocol://server.domain:portcontext-root/webtop/AELView?sql=string&
transientname=filtername&viewname=viewname&viewtype=viewtype&
datasource=datasourcename
```

The possible parameters are as follows:

#### **datasource**

A data source defined in the *ncwDataSourceDefinitions.xml* data source configuration file. If you do not specify this parameter, the default data source is used. For more information about the default data source, see the *IBM Tivoli Netcool/OMNIbus Installation and Deployment Guide*.

**Tip:** If you have defined multiple data sources, you can specify them in the string by using a comma-separated list, as shown in the following example:

```
datasource=datasource1,datasource2
```

If you specify multiple data sources, make sure that the filter, either a predefined filter or a transient filter, queries fields that are contained in all data sources. If you select a data source that is not defined in the filter, an error message is displayed in the AEL instead of event data.

#### **entity|entities**

Deprecated: Use the **filtertype** and **filtername** parameters instead.

**Tip:** If your installation of the Web GUI was migrated or upgraded from IBM Tivoli Netcool/Webtop, and you have custom pages that open with the **entity** parameter, the Web GUI interprets the parameter as follows:

`filtertype=system&filtername=filtername`

Where *filtername* is the entity; entities are migrated to system filters.

**filtername**

The name of the filter that you want to apply. If you do not specify this parameter, the default filter is used.

If you do not specify a **filtertype** parameter, all filter types are searched to identify a matching filter.

**filtertype**

The type of filter that you want to apply. Use this parameter together with the **filtername** parameter. Possible values are as follows:

- global
- system
- user
- user\_transient

**showmenubar**

Optional: Use this parameter to control whether the displayed AEL contains a menu bar. Possible values are:

- true
- false

**showtitlebar**

Optional: Use this parameter to control whether the displayed AEL contains a title bar. Possible values are:

- true
- false

**sql**

Optional: An SQL filter string. This filter is transient, and does not persist beyond your current session.

**transientname**

Optional: Use this parameter to specify a filter name if you specify a filter string by using the **sql** parameter. The value of this parameter is used to populate the **Filters** list of the AEL.

**viewname**

A view that overrides the default view associated with the filter. If you do not specify this parameter, the default view associated with the filter is used.

**viewtype**

The type of view that you want to apply. Use this parameter together with the **viewname** parameter. Possible values are as follows:

- global
- system
- user

## Filter Builder

To open the Filter Builder, use a URL of the following format:

`protocol://server.domain:portcontext-root/webtop/startFB.do`

## Lightweight Event List

To open the Lightweight Event List (LEL), use a URL of the following format:

`protocol://server.domain:portcontext-root/webtop/lwsel/lwsel.jsp?`

`filtertype=typeoffilter`

`&filtername=filtername&viewname=viewname&viewtype=viewtype&datasource=datasourcename`

The parameters are as follows:

### **datasource**

A data source defined in the `ncwDataSourceDefinitions.xml` data source configuration file. If you do not specify this parameter, the default data source is used. For more information about the default data source, see the *IBM Tivoli Netcool/OMNIBus Installation and Deployment Guide*.

### **entity|entities**

Deprecated: Use the **filtertype** and **filtername** parameters instead.

**Tip:** If your installation of the Web GUI was migrated or upgraded from IBM Tivoli Netcool/Webtop, and you have custom pages that open with the **entity** parameter, the Web GUI interprets the parameter as follows:

`filtertype=system&filtername=filtername`

Where *filtername* is the entity; entities are migrated to system filters.

### **filtername**

The name of the filter that you want to apply. If you do not specify this parameter, the default filter is used.

If you do not specify a **filtertype** parameter, all filter types are searched to identify a matching filter.

### **filtertype**

The type of filter that you want to apply. Use this parameter together with the **filtername** parameter. Possible values are as follows:

- global
- system
- user
- user\_transient

### **viewname**

A view that overrides the default view associated with the filter. If you do not specify this parameter, the default view associated with the filter is used.

### **viewtype**

The type of view that you want to apply. Use this parameter together with the **viewname** parameter. Possible values are as follows:

- global
- system
- user

## Map pages

To open a map page, use a URL of the following format:

*protocol://server.domain:portcontext-root/webtop/Map/map*

The parameters are as follows:

### **map**

Required: The name of a map.

## Table View

To open the Table View, use a URL of the following format:

*protocol://server.domain:portcontext-root/webtop/TableView/?  
filtertype=typeoffilter&filtername=filtername&viewname=viewname&viewtype=viewtype&  
datasource=datasource&maxrows=n*

The parameters are as follows:

### **datasource**

A data source defined in the *ncwDataSourceDefinitions.xml* data source configuration file. If you do not specify this parameter, the default data source is used. For more information about the default data source, see the *IBM Tivoli Netcool/OMNIBus Installation and Deployment Guide*.

### **entity|entities**

Deprecated: Use the **filtertype** and **filtername** parameters instead.

**Tip:** If your installation of the Web GUI was migrated or upgraded from IBM Tivoli Netcool/Webtop, and you have custom pages that open with the **entity** parameter, the Web GUI interprets the parameter as follows:

*filtertype=system&filtername=filtername*

Where *filtername* is the entity; entities are migrated to system filters.

### **filtername**

The name of the filter that you want to apply. If you do not specify this parameter, the default filter is used.

If you do not specify a **filtertype** parameter, all filter types are searched to identify a matching filter.

### **filtertype**

The type of filter that you want to apply. Use this parameter together with the **filtername** parameter. Possible values are as follows:

- global
- system
- user
- user\_transient

### **maxrows**

Optional: Specifies the number of rows returned in the Table View.

### **viewname**

A view that overrides the default view associated with the filter. If you do not specify this parameter, the default view associated with the filter is used.



**viewtype**

The type of view that you want to apply. Use this parameter together with the **viewname** parameter. Possible values are as follows:

- global
- system
- user



---

## Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
958/NH04  
IBM Centre, St Leonards  
601 Pacific Hwy  
St Leonards, NSW, 2069  
Australia

IBM Corporation  
896471/H128B  
76 Upper Ground  
London SE1 9PZ  
United Kingdom

IBM Corporation  
JBFA/SOM1  
294 Route 100  
Somers, NY, 10589-0100  
United States of America

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

---

## Trademarks

IBM, the IBM logo, `ibm.com`<sup>®</sup>, Netcool, Netcool/OMNIbus, Tivoli, and WebSphere are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both

Adobe, Acrobat, Portable Document Format (PDF), PostScript, and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.



Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.



---

# Index

## Special characters

<button> 128  
<cgi> 172  
<columns> 119  
<file> 146  
<filter> 175  
<filterCollection> 181  
<icon> 133  
<img> tag attributes  
    charts 344  
<line> 136  
<map> 125  
<menu> 150  
<metric:command> 187  
<metric:metric> 185  
<metric:text> 188  
<monitor> 130  
<prompt:additionalParams> 165  
<prompt:choice> 166  
<prompt:item> 166  
<prompt:param> 165  
<prompt:parameters> 164  
<prompt:prompt> 164  
<resource> 144  
<resources> 144  
<separator> 150  
<sortColumn> 120  
<sorting> 120  
<supermenu> 150  
<text> 125  
<tool:access> 154  
<tool:cgiurl> 157  
<tool:cmdline> 158  
<tool:command> 158  
<tool:criterion> 155  
<tool:equals> 155  
<tool:field> 157  
<tool:fieldlist> 157  
<tool:journal> 156  
<tool:osfield> 155  
<tool:script> 158  
<tool:security> 156  
<tool:sql> 156  
<tool:tool> 154  
<tool> 151  
<user> 110  
<view> 119  
<visualEntry> 119

## A

about this profile 2  
accessibility x, 387  
Active Event List  
    accessing 372  
    acknowledging events 372  
    adding sound files 250  
    appearance and behavior 241  
    assigning ownership to events 373  
    brightness 245

Active Event List (*continued*)  
    changing the displayed event  
        data 373  
    colors 245, 250  
    copying data to the clipboard 374  
    data row caching 242  
    date and time formats 246, 247  
    deacknowledging events 372  
    defaults 225  
    deleting events 374  
    deselecting events 381  
    displaying full event information 374  
    event severity  
        icons 248  
    event severity icons 242  
    filtering 380  
    filtering by severity 380  
    flashing 245  
    fonts 245, 250  
    freezing event data 375  
    linking with map object 324  
    maintaining an event journal 376  
    monitoring TADDM events 379  
    notifications 244  
    prioritizing events 377  
    refresh rate 241  
    refreshing event data 377  
    restricting rows 243  
    running predictive eventing  
        tools 378  
    searching for text 381  
    selecting events 381  
    sorting columns 382  
    toolbars 245  
    tools 383  
adding a user to groups 76  
adding more groups as members of a  
    group 90  
adding more users as members of a  
    group 79  
adding tools to menu 277  
adding users to a group 78  
administration  
    Tivoli Integrated Portal 68  
    user tasks 68  
    Web GUI 68  
advanced commands 51  
AEL  
    event management overview 371  
Alert menu 276  
application server  
    ports 3  
    profile 2  
attributes  
    charts 353  
    configuration files 395  
audience vii  
authentication  
    client messages 208  
    client-server 196  
        FIPS 140-2 198

authentication (*continued*)  
    server-only 195  
        FIPS 140-2 197

## B

back up 21  
    server settings 21, 211  
basic commands 49

## C

CGI request 172  
    create or replace 173  
    modify 173  
    register 172  
    unregister 174  
CGI scripts  
    environment variables 255  
CGI support 214  
changing display options for a group 86  
changing display options for a user 75  
changing user information 77, 88  
Chart portlet  
    defaults 227  
charts  
    <chartelement> syntax 340  
    <img> tag attributes 344  
    attribute reference 353  
    definition file 339  
    definition file, creating 343  
    element reference 345  
    generation 338  
    overview 337  
    presenting 344  
    schema 340  
    types 338  
client-server  
    security 196  
        FIPS 140-2 198  
cloning  
    server settings 21, 22, 211  
cloning data between servers 16  
    exporting data 16  
    importing data 18  
    overview 10  
command-line 252  
command-line tools  
    creating 263  
commands 402  
    confvmm4ncos 5, 6  
    nco\_p\_simnet 315  
    SmartPage  
        conventions 402  
        enable:Smartpage 402  
        insert:AELMAPVIEW 405  
        insert:FILTERPAGE 407  
        insert:MAPLET 406  
        insert:TABLEVIEW 408  
        insert:USERMAPLIST 408

- commands (*continued*)
  - SmartPage (*continued*)
    - insert:USERNAME 409
    - insert:USERSHOMEURL 409
    - templates 409, 410
    - Validate 402
- communications 103
- comparison
  - WAAPI and Web GUI procedure 102
- configuration files
  - attributes 395
  - DTD 389
  - elements 392
  - structure 389
  - XML 389
- conventions, typeface x
- copying data between servers 10
  - cloning all data 16
  - defining the data to export 11
  - defining the data to import 14
  - exporting data from the source server 11
  - exporting the data 13
  - importing data to the target server 14
  - importing the data 15
  - overview 10
  - setting the log level 20
  - verifying the exported data 14
  - verifying the imported data 16
- creating filters 74, 86
- creating users 75
- CTGES1504E and CTGES1505E 208
- customizing
  - adding tools to menu 277

## D

- data row caching 242
- database schema 413, 414
- database tables 413, 414
- date and time formats 246, 247
- defaults
  - setting
    - Active Event List 225
    - Chart portlet 227
    - Event Dashboard 228
    - Inline Frame portlet 235
    - Lightweight Event List 236
    - Map portlet 237
    - Table View portlet 238
    - Web widget portlet 239
- definition file
  - chart, creating 343
  - charts 339
  - syntax 339
- deleting groups 91
- deleting users 80
- dependent filters 283
  - creating 292
- dependentlist 176
- Deployment Engine
  - managing 23
- Document Type Definition 108
- DTD
  - See also* Document Type Definition for configuration files 389

- duplicating group assignments for groups 91
- duplicating group assignments for users 79

## E

- editing
  - properties files 203
- education
  - see* Tivoli technical training x
- element
  - <button> 128
  - <cgi> 172
  - <columns> 119
  - <file> 146
  - <filter> 175
  - <filterCollection> 181
  - <icon> 133
  - <line> 136
  - <map> 125
  - <menu> 150
  - <metric:command> 187
  - <metric:metric> 185
  - <metric:text> 188
  - <monitor> 130
  - <prompt:additionalParams> 165
  - <prompt:choice> 166
  - <prompt:item> 166
  - <prompt:param> 165
  - <prompt:parameters> 164
  - <prompt:prompt> 164
  - <resource> 144
  - <resources> 144
  - <separator> 150
  - <sortColumn> 120
  - <sorting> 120
  - <supermenu> 150
  - <text> 125
  - <tool:access> 154
  - <tool:cgiurl> 157
  - <tool:cmdline> 158
  - <tool:command> 158
  - <tool:criterion> 155
  - <tool:equals> 155
  - <tool:field> 157
  - <tool:fieldlist> 157
  - <tool:journal> 156
  - <tool:osfield> 155
  - <tool:script> 158
  - <tool:security> 156
  - <tool:sql> 156
  - <tool:tool> 154
  - <tool> 151
  - <user> 110
  - <view> 119
  - <visualEntry> 119
- element<dependentlist 176
- elements
  - charts 345
  - configuration files 392
- environment variables
  - CGI scripts 255
- environment variables, notation x
- Event Dashboard 305
- Event Dashboarddefaults 228

- Event Dashboards
  - freezing 310
  - interaction with Active Event List 311, 312
  - monitor boxes
    - adding 306
    - arranging 309
    - customizing 308
    - removing 307
  - preferences 308
  - troubleshooting 209
- event severity icons 242
- events
  - filtering 380
  - managing
    - tools 383
  - managing in the AEL 371
  - maps
    - changing color 334
  - monitoring
    - Active Event List 371
    - Lightweight Event List 383
  - querying 8
  - searching for text 381
  - severity
    - icons 248
  - severity icons 242
  - top n 243
  - visualizing
    - charts 337
    - Event Dashboards 305
    - gauges 361
    - maps 313
- export 10
  - defining the data to export 11
  - exporting data from a server 11
  - exporting the data 13
  - overview 10
  - server settings 21
  - setting the log level 20
  - verifying the exported data 14
- exporting 47, 49, 51
  - basic export console preference profiles 50
  - basic export pages 49
  - basic export views 50
  - export all 51
  - export pages 53
  - export views 54
  - rules 55, 57
  - settings file 52

## F

- file request 146
  - add a directory 146
  - add a file 147
  - create or replace a file 148
  - delete a file 148
  - recursively remove a directory 149
  - remove a directory 148
- files
  - deployment.xml 3
  - ncwDataSourceDefinitions.xml 6
  - server.init 243, 334
  - waapi.init 195, 196, 197, 198, 201, 202



- Filter Builder
  - accessing 286
  - comparators 294
  - overview 286
- filter collection request 180
  - add filter 183
  - create 180
  - create or replace 181
  - delete 183
  - delete filter 184
  - list 184
  - modify 182
  - set view 184
- filter collections 283
  - editing 293
- filter request 174
  - add 174
  - create or replace 177
  - delete 179
  - list 179
  - modify 178
  - set default view 180
- filtering a list of groups 86
- filtering a list of users 74
- filters
  - categories 283
  - collections 283
    - editing 293
  - copying 296
  - creating
    - advanced mode 290
    - basic mode 287
    - dependent 292
  - deleting 297
  - dependent 283
  - Event Dashboards 308
  - invalid characters 399
  - regular expressions 295
  - transient 283
- FIPS 140-2
  - client-server 198
  - server-only 197
- flashing
  - AEL 245
- fonts
  - Active Event List 245, 250

## G

- gauges
  - changing 231
  - metrics 365
  - publishing 365
  - setting thresholds 368
- global filters 283
- global views 285
- groups 67
  - adding more as members of a group 90
  - changing information about 88
  - changing viewing options 86
  - creating filters 86
  - deleting 91
  - duplicating group assignments 91
  - filtering a list of 86
  - managing 84
  - removing members from 90

- groups *(continued)*
  - searching for 85, 88, 91
  - supplied 67
  - viewing group information 88
  - viewing membership of 89

## H

- HTML Map Editor 328
  - buttons 328
  - creating maps 316
  - icons 330
  - lines 331
  - Table View 330
- HTML Map Editortext 333

## I

- icons
  - Active Event List 248
- import 10
  - defining the data to import 14
  - importing data to a server 14
  - importing the data 15
  - overview 10
  - setting the log level 20
  - verifying the imported data 16
- importing 47, 56
  - import data 56
  - rollback 57
  - server settings 22
- Inline Frame portlet
  - defaults 235
- invalid characters
  - when defining filters, views, and tools 399
- isc\_admin role 35
- isc\_admins role 69

## J

- Java Map Editor 317, 327
  - access control to maps 317
  - adding objects 319
  - colors 326
  - creating maps 316
  - event lists 324
  - fonts 326
  - setting background 317
  - setting grid 318
  - setting object position 325
  - setting object size 325
  - setting object URLs 322
  - setting size 317
- Java Map Editorsetting object properties 320
- Java Runtime Environment 202
- JavaScript
  - tools 253

## L

- Lightweight Event List
  - accessing 383
  - defaults 236

- Lightweight Event List *(continued)*
  - displaying full information 384
  - linking with map object 324
  - viewing events 384
- load balancing 413, 414
  - best practices 98, 221
  - maintain list of files to replicate 97, 219
  - recovering from database corruption 100, 222
  - remove node 192
  - resynchronize a node with the cluster 100, 222
  - troubleshooting 100, 222
- load balancing cluster
  - administering 92, 215
- log files
  - creating a performance log 205
  - location 207
  - setting the log level 203
- login
  - must have roles 208
  - slow response 208
  - troubleshooting 203
  - users 208

## M

- managing groups 84
- managing users 73
- manuals viii
- Map portlet
  - defaults 237
  - preferences 237
  - publishing maps 335
- map request 124
  - create 125
  - create or replace 139
  - delete 140
  - list 140
  - modify 139
- map visual request
  - add 141
  - add or replace 141
  - delete 143
  - modify 142
- maps
  - colors picker 327
  - configuring in HTML 328
  - configuring with the Java Map Editor 317
  - creating
    - HTML Map Editor 316
    - Java Map Editor 316
  - customizing
    - access control 317
    - adding objects 319
    - background 317
    - buttons 328
    - event lists 324
    - grid 318
    - icons 330
    - lines 331
    - map object URLs 322
    - object position 325
    - object properties 320
    - object size 325

- maps (*continued*)
  - customizing (*continued*)
    - size 317
    - text 333
  - customizingcolors 326
  - customizingfonts 326
  - customizingTable View 330
  - deleting resource 337
  - linking 333
  - overview 313
  - previewing 315
  - previewing resources 337
  - publishing Web widget portlet 335
  - publishingMap portlet 335
  - samples 315
  - specifying a URL target 333
  - Table View 332
  - troubleshooting 210
- memory usage 202
- menu request 149
  - create 149
  - create or replace 151
  - delete 153
  - list 153
  - modify 152
- menus
  - Alert 276
  - changing the order of items 280
  - removing items 280
  - renaming items 280
  - submenus 279
    - creating 278
  - Tools 276
- metric request 185
  - create 185
  - create or replace 189
  - delete 191
  - list 191
  - modify 190

## N

- navigation pane
  - changing access 35
- ncw\_admin role 66
  - assigning to user 69
- ncw\_dashboard\_editor role 66
- ncw\_gauges\_editor 66
- ncw\_gauges\_viewer role 66
- ncw\_user role 66
- netcool\_ro role 66
- netcool\_rw role 66
- notifications 244
  - adding sound files 250

## O

- ObjectServer
  - adding as user registry 5
  - password
    - changing 6
    - querying 8
- ObjectServer properties 202
- online publications viii
- operating system information 202
- ordering publications viii

- other request 192
  - reload filters and views 193
  - remove a node from a cluster 192
  - resynchronize the cache 192
  - system status report 192

## P

- password
  - change 69
  - encryption
    - AES 201
    - FIPS 140-2 201
  - SSL 69
- passwords
  - encryption 104, 200
- performance tuning 212
  - restricting rows in the Active Event List 243
  - Web GUI 211
- port
  - numbers 2
- port assignments 3
- predictive events 378
- preferences
  - Active Event List 225
  - Chart portlet 227
  - Event Dashboard 228
  - Inline Frame portlet 235
  - Lightweight Event List 236
  - modifying 70
  - setting
    - Active Event List 225
    - Chart portlet 227
    - Event Dashboard 228
    - Inline Frame portlet 235
    - Lightweight Event List 236
    - Map portlet 237
    - Table View portlet 238
    - Web widget portlet 239
  - Web widget portlet 239
- prompt request 164
  - create 164
  - create or replace 170
  - delete 171
  - list 172
  - modify 171
  - usage notes 166
- prompts 253
- properties
  - editing files 203
- publications viii

## R

- refresh rate 241
- regular expressions
  - filters 295
- remote host
  - installation
    - WAAPI client 193
- removing members of a group 90
- removing users from a group 78, 89
- request 106
  - CGI 172
  - file 146

- request (*continued*)
  - filter 174
  - filter collection 180
  - map 124
  - menu 149
  - metric 185
  - other 192
  - prompt 164
  - resource 143
  - structure 107
  - tool 153
  - user 110
  - view 118
- resource request 143
  - add 144
  - create or replace 145
  - list 146
  - remove 145
- roles 65, 66
  - isc\_admins 35, 69
  - ncw\_admin 69
  - of supplied groups 67
- runtime parameters
  - script tools 274

## S

- samples
  - maps 315
  - ObjectServer query 8
  - script tools 266
- script tools
  - creating 265
- scripts
  - CGI
    - creating 255
    - registering 257
  - searching for groups 85, 88, 91
  - searching for users 73, 76, 77, 78, 79, 80, 87, 89
- security 194
  - FIPS 140-2
    - client-server 198
    - server-side 197
  - password encryption 104, 200
    - AES 201
    - FIPS 140-2 201
  - WAAPI
    - client-server 196
    - overview 104, 194
    - server-side 195
- server 21
  - back up 21
  - ObjectServer connection 6
  - security 195
    - FIPS 140-2 197
  - set a trace 205
  - stopping or starting 1
- server settings
  - back up 21
  - clone 21
  - cloning 22, 211
  - export 21
  - importing 22
- server version and properties 202
- SmartPage
  - code sample 401

- SmartPage *(continued)*
  - commands
    - enable:Smartpage 402
    - insert:AEL 402
    - insert:AELMAPVIEW 405
    - insert:FILTERPAGE 407
    - insert:MAPLET 406
    - insert:TABLEVIEW 408
    - insert:USERMAPLIST 408
    - insert:USERNAME 409
    - insert:USERSHOMEURL 409
    - Validate 402
  - commands overview 401
  - conventions 402
  - creating templates 411
  - insert:AEL 402
  - templates
    - example 410
    - guidelines 409, 411
    - location on server 409
- SmartPage commands
  - insert:MAPLET 406
- SQL
  - filters 287, 290
  - query
    - sample 8
- SSL
  - password encryption
    - AES 201
- startup pages 35
- stopping the application server 1
- submenus
  - adding to menu 279
  - creating 278
  - deleting 279
- support information x
- system filters 283
- system information
  - checking 202
- system views 285

**T**

- Table View
  - linking with map object 324
  - maps 330, 332
  - viewing events 385
- Table View portlet
  - defaults 238
  - preferences 238
- TADDM events
  - monitoring 379
- tipcli command
  - export 58, 61
- Tivoli software information center viii
- Tivoli technical training x
- tool request 153
  - create 154
  - create or replace 161
  - delete 163
  - list 163
  - modify 162
  - usage notes 159
- tools 252
  - access criteria 253
  - adding to menu 277
  - CGI 251, 255

- tools *(continued)*
  - creating 258
  - examples 258
  - registering 257
  - scripting 255
  - copying 271
  - deleting 274
  - environment variables 255
  - invalid characters 399
  - managing events 383
  - modifying 267, 277
  - multiple data sources 250
  - overview 251
  - prerequisites 250
  - prompts 253
  - renaming 270
  - script 253
    - runtime parameters 274
    - samples 266
  - SQL 252
    - creating 261
- Tools menu 276
- trace 205
- training, Tivoli technical x
- transient filters 283
- trouble-shooting
  - log files 207
- troubleshooting 202
  - auditing the usage of objects 206
  - creating a performance log 205
  - Event Dashboard 209
  - load balancing cluster 100, 222
  - login errors 203
  - maps 210
  - query timeout 211
  - recovering from database
    - corruption 100, 222
  - resynchronize a node with the
    - cluster 100, 222
  - setting the log level 203
- truststore
  - WAAPI
    - signer certificate 195, 196, 197, 198
- typeface conventions x

**U**

- usage notes
  - prompt request 166
  - tool request 159
- user filters 283
- user registry
  - ObjectServer
    - adding 5
- user request 110
  - list 118
  - maintain 118
  - modify 110
- user timeout
  - changing 3
- user views 285
- users
  - adding a user to groups 78
  - adding more as members of a
    - group 79
  - adding to groups 76

- users *(continued)*
  - administration 68
  - changing display options 75
  - changing information about 77
  - creating 75
  - creating filters 74
  - creating Web GUI administrator
    - user 69
  - deleting 80
  - duplicating group assignments 79
  - filtering a list of 74
  - groups 67
    - removing from groups 76
  - managing 73
  - modifying preferences 70
  - removing a group from groups 76
  - removing from a group 78, 89
  - roles 65, 66
  - searching for 73, 76, 77, 78, 79, 80, 87, 89
  - supplied 67
  - viewing group information 77
  - viewing information about 76, 87
- utilities
  - acknowledge 252
  - nco\_ping 251
  - ncw\_aes\_crypt 201
  - ncw\_fips\_crypt 201
  - ping 252

**V**

- variables, notation for x
- View Builder
  - accessing 298
- view request 118
  - create 118
  - create or replace 122
  - delete 124
  - list 124
  - modify 123
- viewing group information for a
  - group 88
- viewing group information for a user 77
- viewing the members of a group 89
- viewing user information 76, 87
- views
  - adding columns 299
  - categories 285
  - column order 300
  - column sort order 303
  - column titles 300
  - column width 300
  - copying 303
  - creating 298
  - invalid characters 399
  - locking columns 301
  - removing columns 299
  - sorting precedence 302
  - View Builder 298

**W**

- WAAPI
  - communication with the server 103
  - comparison with Web GUI 102

- WAAPI (*continued*)
  - configuration overview 100
  - installation
    - remote host 193
  - password encryption 104, 200
    - AES 201
    - FIPS 140-2 201
  - properties 202
  - security 194
    - overview 104, 194
  - truststore
    - signer certificate 195, 196, 197, 198
  - using 104
- WAAPI request 106
  - case sensitivity 109
  - CGI 172
  - characteristics 108
  - comments 109
  - content and value restrictions 109
  - Document Type Definition 108
  - file 146
  - filter 174
  - filter collection 180
  - map 124
  - menu 149
  - metric 185
  - order of elements 108
  - other 192
  - prompt 164
  - resource 143
  - root element 107
  - structure 107
  - tool 153
  - user 110
  - view 118
  - XML declaration 107
- Web widget portlet
  - defaults 239
  - publishing maps 335

## X

- XML
  - for configuration files 389





Printed in the Republic of Ireland

SC14-7606-00

