



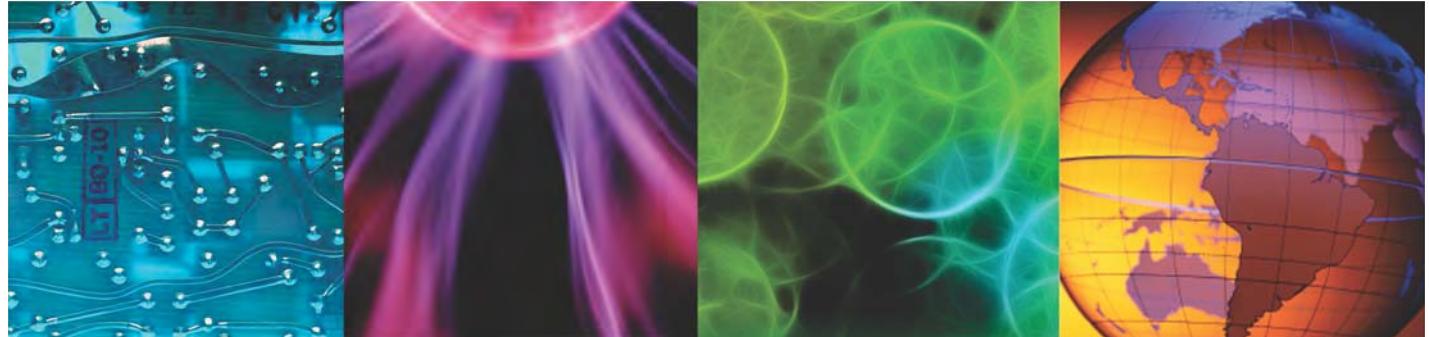
IBM Training

IBM Tivoli Netcool/OMNIbus 8.1 Administration and Maintenance

Student Exercises

Course code TN035 ERC 2.0

May 2016



All files and material for this course are IBM copyright property covered by the following copyright notice.

© Copyright IBM Corp. 2016. All Rights Reserved.

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

IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

References in this publication to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth, savings or other results.



Contents

1 Introduction to Netcool/OMNIbus Administration exercises	1
Exercise 1 Starting the images	1
Starting host1	1
Starting host2	3
Exercise 2 Stopping and starting components	4
Stopping and starting the ObjectServer	4
Implementing controlled ObjectServer shutdown	7
Stopping and starting Dashboard Application Services Hub	10
Exercise 3 Backing up and restoring	13
Backing up and restoring the ObjectServer	13
Backing up and restoring the Web GUI	15
Exercise 4 Applying maintenance	20
Exercise 5 Modifying behavior	23
Modifying ObjectServer behavior	24
2 ObjectServer administration exercises	29
Exercise 1 Modifying the event record	29
Exercise 2 Modifying the gateway	33
Exercise 3 Creating an ObjectServer	36
3 Probes exercises	41
Exercise 1 MIB Manager	41
Generating rules	41
Adding rules to Netcool Knowledge Library	45
Validating the rules	52
Exercise 2 Probe high availability	55
Exercise 3 Probe remote administration	61
Modifying the ProbeWatch configuration	61
Configuring process activity	67
Installing the rules reload feature	69
Using the feature	74
4 Automations exercises	81
Exercise 1 Basic SQL commands	81
Working with the nco_sql utility	81
Working with the SQL workbench	85
Exercise 2 Automations	92
Working with temporal triggers	92
Working with database triggers	97

5 Web GUI administration exercises	99
Exercise 1 Creating filters, and views	99
Updating grouping columns	99
Creating a view	100
Creating filters	105
Exercise 2 Creating prompts, and tools	110
Configuring command tool support for the Event Viewer	110
Enabling command tools for Linux	116
Creating a command tool	117
Adding a tool to a menu	118
Creating an SQL tool, and prompt	120
Testing the tool	123
Exercise 3 Creating maps	125
Exercise 4 Working with gauges	132
Examining an existing gauge	132
Creating metrics	138
Creating gauges	140
Exercise 5 Creating a dashboard	145
Exercise 6 Working with Web GUI administration API	148
Configuring the client	148
Working with the client	149
6 User administration exercises.....	152
Exercise 1 Configuring users for event access	152
Creating a group	152
Adding roles to groups	155
Limiting access to event records	157
Exercise 2 Configuring default startup pages	160
Creating a role	160
Assigning the role to a page	161
Creating a view	163
Creating a console preference profile	165
Modifying group roles	168
Validating user access	170
Exercise 3 Configuring users for native event list access	174
Configuring the ObjectServer for LDAP password authentication	174
Configuring the ObjectServer user environment	177
7 Customizing Tivoli Common Reporting reports exercises	182
Exercise 1 Modifying the event archive database	182
Exercise 2 Modifying the archive gateway	184
Exercise 3 Installing DB2	187
Exercise 4 Installing Framework Manager	190
Exercise 5. Modifying and publishing the Cognos data model	193
Database view modifications	196
Consolidation view modifications	199
Presentation view modifications	204
Publishing the updated model	204
Exercise 6. Accessing and testing the new model	208

Accessing the new package in Tivoli Common Reporting208
8 Web GUI high availability exercises	216
Exercise 1 Configuring Dashboard Application Services Hub on host1	216
Configuring Dashboard Application Services Hub to allow logins if LDAP is down	228
Exercise 2 Creating the configuration database on host2	233
Exercise 3 Creating the cluster with host2	234
Exercise 4 Adding host1 to the cluster	239
Exercise 5 Configuring server to server trust	243
Configuring trust on host1	244
Configuring trust on host2	246
Exercise 6 Verifying cluster configuration	248
Exercise 7 Configuring Web GUI load balancing	250
Configuring host2	250
Configuring host1	251
Adding a data source	253
Validating Web GUI load balancing	255
Troubleshooting Web GUI load balancing	256
9 Security exercises	259
Overview	259
Exercise 1 Configuring FIPS 140-2 encryption	260
Configuring the host1 image	260
Configuring the host2 image	264
Exercise 2. Configuring the ObjectServers for SSL access	268
Configuring the host1 image	268
Configuring the host2 image	276
Distributing ObjectServer keys	280
Exercise 3 SSL property value encryption	282
Configuring the host1 image	283
Configuring the host2 image	287
Exercise 4 Validating SSL with FIPS compliance	291
Verifying the encryption configuration	291
Verifying the SSL configuration	293
Modifying the process activity startup script	298
10 Multitiered architecture exercises	301
Exercise 1 Preparing for the deployment	301
Stopping components on host1	301
Stopping components on host2	302
Exercise 2 Creating the configuration	302
Creating the configuration on host1	303
Creating the configuration on host2	310



1 Introduction to Netcool/OMNIbus Administration exercises

In this unit, you learn how to perform some of the basic administrative functions.

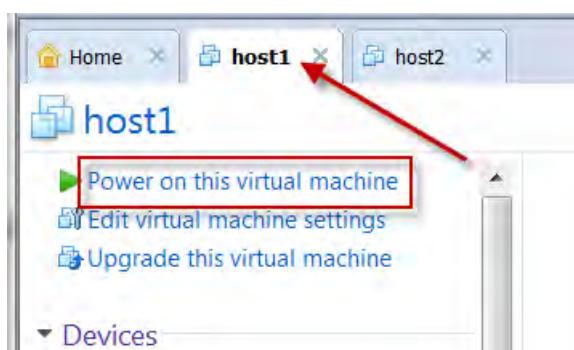
Exercise 1 Starting the images

The classroom environment consists of two VMware images. Depending on the environment, the images might or might not be started. If the images are started, you can skip to the next exercise.

Starting host1

If the host1 image is started, skip to step 2.

1. Locate the **host1** tab in the VMware console.
 - a. Click the **host1** tab to select it.
 - b. Click the line that is labeled **Power on this virtual machine**.

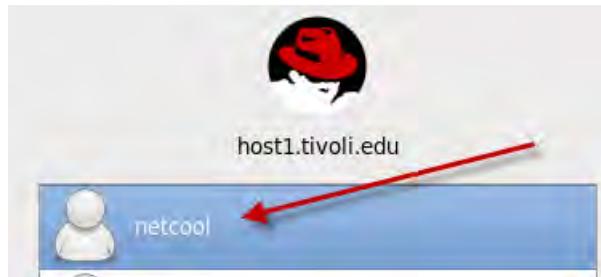


The image takes several minutes to initiate. The login screen opens when the image is available.

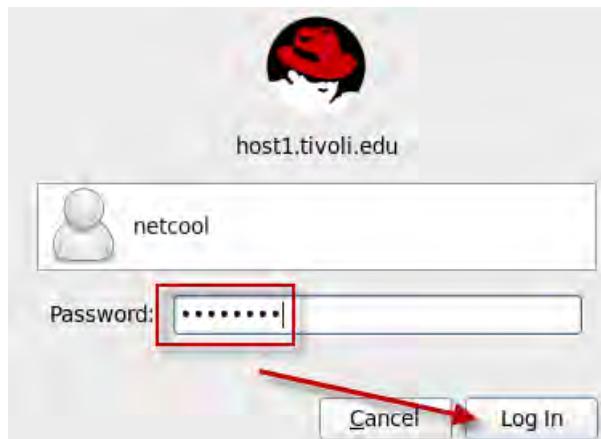
Exercise 1 Starting the images

2. Log in as the netcool user.

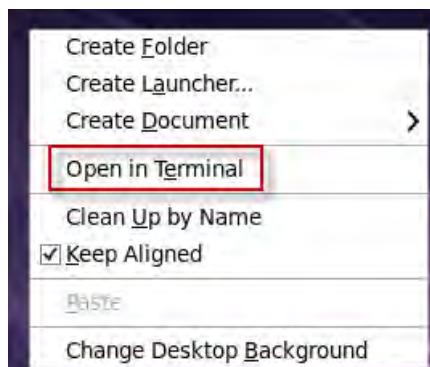
- a. Click **netcool**.



- b. Enter **object00** for the password and click **Log In**.



3. Right-click the desktop and select **Open in Terminal**.



4. Verify Netcool/OMNIbus core component status.

```
nco_pa_status -server HOST1_PA -user netcool -password object00
```

Service Name	Process Name	Hostname	User	Status	PID
Core	MasterObjectServer	host1.tivoli.edunetcool	netcool	RUNNING	2078
	SyslogProbe	host1.tivoli.edunetcool	netcool	RUNNING	2079
	SmpProbe	host1.tivoli.edunetcool	netcool	RUNNING	2080

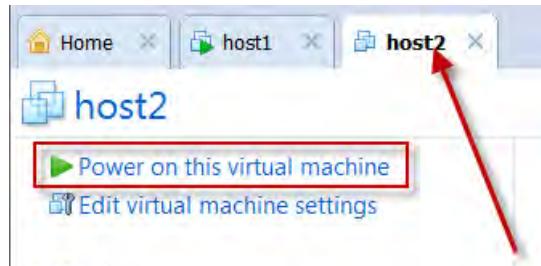
All components must be running before you proceed. If any component is not running, notify the instructor.

Leave the image as is.

Starting host2

If the host2 image is started, skip to step 2.

1. Locate the **host2** tab in the VMware console.
 - a. Click the **host2** tab to select it.
 - b. Click the line that is labeled **Power on this virtual machine**.



The image takes several minutes to initiate. The login screen opens when the image is available.

2. Log in as the **netcool** user with password **object00**.
3. Right-click the desktop and select **Open in Terminal**.
4. Verify Netcool/OMNIbus core component status.

```
nco_pa_status -server HOST2_PA -user netcool -password object00
```

Service Name	Process Name	Hostname	User	Status	PID
Core	BackupObjectServer	host2.tivoli.edunetcool		RUNNING	2170
	BackupGateway	host2.tivoli.edunetcool		RUNNING	2171
	ArchiveGateway	host2.tivoli.edunetcool		RUNNING	2172

All components must be running before you proceed. If any component is not running, notify the instructor.

5. Verify Netcool/OMNIbus Web GUI component status.

```
cd /opt/IBM/JazzSM/profile/bin
./serverStatus.sh server1 -user smadmin -password object00
ADMU0116I: Tool information is being logged in file
        /opt/IBM/JazzSM/profile/logs/server1/serverStatus.log
ADMU0128I: Starting tool with the JazzSMProfile profile
ADMU0500I: Retrieving server status for server1
ADMU0508I: The Application Server "server1" is STARTED
```

The component must be running before you proceed. If it is not running, notify the instructor.

Leave the image as is.

Exercise 2 Stopping and starting components

All Netcool/OMNIbus core components are configured as managed by Netcool Process Activity. In the following exercises, you use process activity and command-line actions to stop and start various core components.

Stopping and starting the ObjectServer



Important: You perform the following exercises on **host1**.

1. Locate the ObjectServer process name.

```
nco_pa_status -server HOST1_PA -user netcool -password object00
```

[netcool@host1 var]\$ nco_pa_status -server HOST1_PA -user netcool -password object00					
Service Name	Process Name	Hostname	User	Status	PID
Core	MasterObjectServer	host1.tivoli.edunetcool	running	12091	
	SyslogProbe	host1.tivoli.edunetcool	running	12203	
	SnmpProbe	host1.tivoli.edunetcool	running	12291	



Important: All process activity commands require a UNIX/Linux user id, and password. The user must belong to *ncoadmin* group.

2. Stop the ObjectServer.

```
nco_pa_stop -server HOST1_PA -user netcool -password object00 -process  
MasterObjectServer
```

3. Verify the process status.

```
nco_pa_status -server HOST1_PA -user netcool -password object00
```

[netcool@host1 var]\$ nco_pa_status -server HOST1_PA -user netcool -password object00					
Service Name	Process Name	Hostname	User	Status	PID
Core	MasterObjectServer	host1.tivoli.edunetcool	DEAD	0	
	SyslogProbe	host1.tivoli.edunetcool	running	12203	
	SnmpProbe	host1.tivoli.edunetcool	running	12291	

4. Start the ObjectServer from the command line.

```
nco_objserv -name NYC_AGG_P &  
Netcool/OMNibus Object Server - Version 8.1.0 64-bit
```

(C) Copyright IBM Corp. 1994, 2012

.

.

.

Server 'NYC_AGG_P' initialised - entering RUN state.



Hint: Press **Enter** to get the command-line prompt.

5. Stop the ObjectServer from the command line.

a. Connect to the ObjectServer with the SQL utility.

```
nco_sql -server NYC_AGG_P -user root -password object00  
1>
```



Important: The nco_sql command requires an ObjectServer user id, and password.

The prompt characters indicate successful connection to the ObjectServer.

b. Enter the SQL commands to shut down the ObjectServer;

```
1> ALTER SYSTEM SHUTDOWN;  
2> go  
(0 rows affected)  
1>
```

The message **0 rows affected** is normal.

c. Exit the SQL utility.

```
1> quit
```

6. Test the access to the ObjectServer.

```
nco_ping NYC_AGG_P  
NCO_PING: Server unavailable.
```

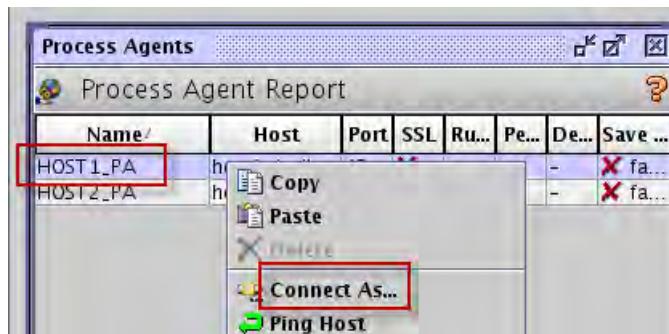
The ObjectServer is not running.

7. Start the ObjectServer with Visual Process Activity.

a. Start the Netcool/OMNibus Administrator utility.

```
nco_config &
```

- b. Right-click **HOST1_PA** and select **Connect As...**.



- c. Log in as **netcool** with password **object00**.

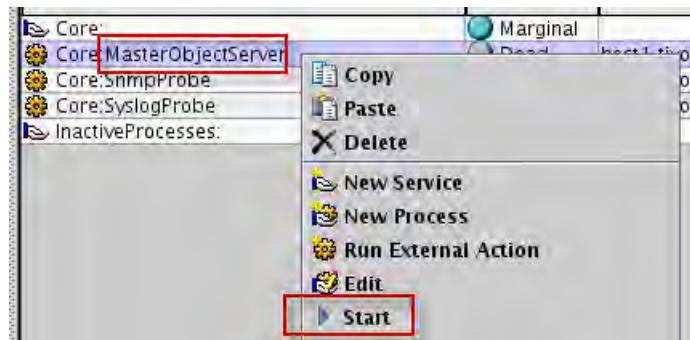


The process status opens. The MasterObjectServer process is DEAD.

Name	Status	Host
Core	Marginal	
Core:MasterObjectServer	Dead	host1.tivoli.edu
Core:SnmpProbe	Running	host1.tivoli.edu
Core:SyslogProbe	Running	host1.tivoli.edu
InactiveProcesses:	Stopped	

8. Start the ObjectServer.

- a. Right-click **MasterObjectServer** and select **Start**.



- b. Verify that the process is running.

Name	Status	Host
Core	Running	
Core:MasterObjectServer	Running	host1
Core:SnmpProbe	Running	host1
Core:SyslogProbe	Running	host1
InactiveProcesses:	Stopped	

9. Close the Netcool/OMNibus Administrator utility.

Implementing controlled ObjectServer shutdown



Important: You perform the following exercises on **host2**.

1. Modify the SQL file as follows:

- a. Change to the target directory.

```
cd /opt/IBM/tivoli/netcool/omnibus/extensions/control_shutdown
```

- b. Create a copy of the SQL file.

```
cp control_shutdown.sql control_shutdown_host2.sql
```

- c. Change file permissions to allow updates.

```
chmod +w control_shutdown_host2.sql
```

- d. Modify the copy of the SQL file.

```
gedit control_shutdown_host2.sql
```

- e. Scroll down in the file and locate the following section:

```
-- External procedure to shutdown OS using nco_pa_stop
-- 'omnihost' - need to be replaced with hostname where external procedure
--               will be executed.
-- 'user' - user name to execute external procedure
-- 'group' - group name to execute external procedure.

create or replace procedure ext_shutdown (in process_name Char(255), in
username Char(255), in pass Char(255), in paserver Char(255))
executeable 'OMNIHOME/bin/nco_pa_stop'
host 'omnihost'
user 0 group 0
arguments '-process ' + process_name + ' -user ' + username + ' -
password ' + pass + ' -server ' + paserver
go
```

- f. Change the value for *host* to **host2.tivoli.edu**.

- g. Change the value for *user* to **500**.

The netcool user is UID 500.

- h. Change the value for **group** to **501**.

The ncoadmin group is GID 501.

```
create or replace procedure ext_shutdown (in proce
username Char(255), in pass Char(255), in paserver
executable 'OMNIHOME/bin/nco_pa_stop'
host 'host2.tivoli.edu'
user 500 group 501
arguments ' -process ' + process_name + ' -user '
```

- i. Scroll down in the file and locate the following section:

```
-- call external procedure to shutdown OS using nco_pa_stop
-- Replace 'MasterObjectServer' with ObjectServer process
name in PA conf file
-- Replace 'USERNAME' with username to execute nco_pa_stop.
-- Replace 'PASSWORD' with password to execute nco_pa_stop.
-- Replace 'PASERVER' with PA server name to connect.
execute procedure ext_shutdown ( 'MasterObjectServer',
'USERNAME', 'PASSWORD', 'PASERVER' );
end if;
end;
```

- j. Change MasterObjectServer to **BackupObjectServer**.

BackupObjectServer is the process activity process name for the NYC_AGG_B ObjectServer.

- k. Change USERNAME to **netcool**.

The netcool user is allowed to control process activity.

- l. Change PASSWORD to **EDEAAPAIANFMCHCB**.

The password is object00. The value is the encrypted password. The password is encrypted with the nco_pa_crypt utility.

- m. Change PASERVER to **HOST2_PA**.

The name of the host2 process activity daemon is HOST2_PA.

```
name in PA conf file
-- Replace 'USERNAME' with username to execute nco_pa_stop.
-- Replace 'PASSWORD' with password to execute nco_pa_stop.
-- Replace 'PASERVER' with PA server name to connect.
execute procedure ext_shutdown ( 'BackupObjectServer',
'netcool', 'EDEAAPAIANFMCHCB', 'HOST2_PA' );
end if;
```

- n. Scroll down in the file and locate the following section.

```
-- disable this trigger group and shutdown ObjectServer.
execute procedure disable_control_shutdown;
-- call external procedure to shutdown OS using nco_pa_stop
-- Replace 'MasterObjectServer' with ObjectServer process
name in PA conf file
-- Replace 'USERNAME' with username to execute nco_pa_stop.
-- Replace 'PASSWORD' with password to execute nco_pa_stop.
-- Replace 'PASERVER' with PA server name to connect.
execute procedure ext_shutdown ( 'MasterObjectServer',
'USERNAME', 'PASSWORD', 'PASERVER' );
end if;
end;
end;
```

- o. Make the same changes as above.

```
name in PA conf file
    -- Replace 'USERNAME' with username to execute nco_pa_stop.
    -- Replace 'PASSWORD' with password to execute nco_pa_stop.
    -- Replace 'PASERVER' with PA server name to connect.
    execute procedure ext_shutdown ( 'BackupObjectServer',
'netcool', 'EDEAAPATANFMCHCB', 'HOST2_PA' );
end if;
```

- o. Save the modified file and exit the gedit utility.

2. Import the file into NYC_AGG_B.

```
nco_sql -server NYC_AGG_B -user root -password object00 <
control_shutdown_host2.sql
(0 rows affected)
```

 **Note:** You can safely ignore the messages.

3. Shut down the ObjectServer.

```
nco_sql -server NYC_AGG_B -user root -password object00
1> execute procedure control_shutdown;
2> go
(1 row affected)
1> quit
```

4. Verify whether the ObjectServer is running.

```
nco_ping NYC_AGG_B
NCO_PING: Server unavailable.
```

The ObjectServer is shut down. Process activity is used by the control_shutdown procedure to stop the ObjectServer. Therefore, you restart the ObjectServer with process activity.

5. Restart the ObjectServer.

a. Determine the process name.

```
nco_pa_status -server HOST2_PA -user netcool -password object00
```

Service Name	Process Name	Hostname	User	Status	PID
Core	BackupObjectServer	host2.tivoli.edunetcool		DEAD	0
	BackupGateway	host2.tivoli.edunetcool		RUNNING	13119
	ArchiveGateway	host2.tivoli.edunetcool		RUNNING	13247

b. Start the process.

```
nco_pa_start -server HOST2_PA -user netcool -password object00 -process
BackupObjectServer
```

Service Name	Process Name	Hostname	User	Status	PID
Core	BackupObjectServer	host2.tivoli.edunetcool		DEAD	0
	BackupGateway	host2.tivoli.edunetcool		RUNNING	13119
	ArchiveGateway	host2.tivoli.edunetcool		RUNNING	13247

c. Verify the status.

```
nco_pa_status -server HOST2_PA -user netcool -password object00
```

Service Name	Process Name	Hostname	User	Status	PID
Core	BackupObjectServer	host2.tivoli.edunetcool		RUNNING	15469
	BackupGateway	host2.tivoli.edunetcool		RUNNING	13119
	ArchiveGateway	host2.tivoli.edunetcool		RUNNING	13247

Stopping and starting Dashboard Application Services Hub

Dashboard Application Services Hub is not managed with Netcool Process Activity.



Important: You perform the following exercises on **host2**.

1. Stop Dashboard Application Services Hub as follows:

a. Change to the target directory.

```
cd /opt/IBM/JazzSM/profile/bin
```

b. Stop the server.

```
./stopServer.sh server1 -user smadmin -password object00
```

```
ADMU0116I: Tool information is being logged in file
/opt/IBM/JazzSM/profile/logs/server1/stopServer.log
```

```
ADMU0128I: Starting tool with the JazzSMPProfile profile
```

```
ADMU3100I: Reading configuration for server: server1
```

```
ADMU3201I: Server stop request issued. Waiting for stop status.
```

```
ADMU4000I: Server server1 stop completed.
```



Important: The administrator user name and password are required to stop the server.

2. Verify the status of the server.

```
./serverStatus.sh server1 -user smadmin -password object00
```

```
ADMU0116I: Tool information is being logged in file  
/opt/IBM/JazzSM/profile/logs/server1/serverStatus.log
```

```
ADMU0128I: Starting tool with the JazzSMPProfile profile
```

```
ADMU0500I: Retrieving server status for server1
```

```
ADMU0509I: The Application Server "server1" cannot be reached. It appears to be  
stopped.
```



Important: The administrator user name and password are required to verify the server status.

3. Start the server.

```
./startServer.sh server1
```

```
ADMU0116I: Tool information is being logged in file  
/opt/IBM/JazzSM/profile/logs/server1/startServer.log
```

```
ADMU0128I: Starting tool with the JazzSMPProfile profile
```

```
ADMU3100I: Reading configuration for server: server1
```

```
ADMU3200I: Server launched. Waiting for initialization status.
```

```
ADMU3000I: Server server1 open for e-business; process id is 10437
```

The application might take several minutes to start.



Important: The administrator user name and password are not required to start the server.

4. Verify the status of the server.

```
./serverStatus.sh server1 -user smadmin -password object00
```

```
ADMU0116I: Tool information is being logged in file  
/opt/IBM/JazzSM/profile/logs/server1/serverStatus.log
```

```
ADMU0128I: Starting tool with the JazzSMPProfile profile
```

```
ADMU0500I: Retrieving server status for server1
```

```
ADMU0508I: The Application Server "server1" is STARTED
```

The classroom image is configured with a custom script to start Dashboard Application Services Hub when the server starts. The script is not included with Netcool/OMNibus. The script was created manually.

5. Change to the **root** user.

```
su -
Password: object00
```

6. Examine the custom script.

```
more /etc/init.d/jazz
```

```
case "$1" in
  'start')
    # start Jazz SM
    su - netcool -c "$JAZZ_HOME/bin/startServer.sh server1"
    ;;
  'stop')
    # stop Jazz SM
    su - netcool -c "$JAZZ_HOME/bin/stopServer.sh server1 -username smadmin -password object00"
    ;;
  'status')
    # Status Jazz SM
    su - netcool -c "$JAZZ_HOME/bin/serverStatus.sh server1 -username smadmin -password object00"
    ;;
esac
```

The script can be used to start, stop, or query the status of Dashboard Application Services Hub.

The script is run as the **root** user. However, the commands to start, stop, and status the server are configured to run as the **netcool** user.

The administrator user name and password are hardcoded in the script.

7. Use the script to check the server status as follows:

```
/etc/init.d/jazz status
```

```
ADMU0116I: Tool information is being logged in file
          /opt/IBM/JazzSM/profile/logs/server1/serverStatus.log
ADMU0128I: Starting tool with the JazzSMProfile profile
ADMU0500I: Retrieving server status for server1
ADMU0508I: The Application Server "server1" is STARTED
```

8. Exit the **root** user back to the **netcool** user.

```
exit
```

Exercise 3 Backing up and restoring

Backing up and restoring the ObjectServer

In the following exercise, you simulate a disk or file system failure. The failure affects the backup ObjectServer on host2.



Important: You perform the following exercises on **host2**.

The first step is to create a copy of the backup ObjectServer. You use the ObjectServer Report Generator utility to accomplish the backup.

1. Create a directory to hold the exported SQL files.

```
cd /home/netcool  
mkdir NYC_AGG_B
```

2. Export the ObjectServer configuration.

```
nco_osreport -dbinit -server NYC_AGG_B -user root -password object00 -directory  
/home/netcool/NYC_AGG_B
```

3. Verify the output files.

```
cd /home/netcool/NYC_AGG_B  
ls -l
```

```
alertsdata.sql  
application.sql  
automation.sql  
desktop.sql  
security.sql  
system.sql
```

The SQL files contain the commands necessary to rebuild the NYC_AGG_B ObjectServer.



Hint: In a production environment, create an archive of the NYC_AGG_B directory, and save a copy of the archive file to some external backup.

The next step is to simulate the file system failure. You manually remove the files that are required to start the NYC_AGG_B ObjectServer to simulate the failure.

4. Stop the ObjectServer synchronizer gateway.

```
nco_pa_stop -server HOST2_PA -user netcool -password object00 -process  
BackupGateway
```

5. Stop the NYC_AGG_B ObjectServer.

```
nco_pa_stop -server HOST2_PA -user netcool -password object00 -process  
BackupObjectServer
```

6. Remove the ObjectServer checkpoint files.

```
cd /opt/IBM/tivoli/netcool/omnibus/db
```

```
/bin/rm -R NYC_AGG_B
```

7. Remove the ObjectServer files from var.

```
cd /opt/IBM/tivoli/netcool/omnibus/var
```

```
/bin/rm NYC_AGG_B.*
```

 **Note:** There might not be any files to remove from this directory.

8. Remove the ObjectServer log files.

```
cd /opt/IBM/tivoli/netcool/omnibus/log
```

```
/bin/rm NYC_AGG_B.*
```

9. Remove the ObjectServer property file.

```
cd /opt/IBM/tivoli/netcool/omnibus/etc
```

```
/bin/rm NYC_AGG_B.props
```

All of the required files for the NYC_AGG_B ObjectServer are removed.

The next step is to re-create the ObjectServer with the SQL files created previously.

10. Create the ObjectServer from the command line.

```
cd /home/netcool/NYC_AGG_B
```

```
nco_dbinit -alertsdata -alertsdatafile alertsdatafile.sql -applicationfile  
application.sql -automationfile automation.sql -desktopfile desktop.sql  
-securityfile security.sql -systemfile system.sql -server NYC_AGG_B
```

11. Start the ObjectServer.

```
nco_pa_start -server HOST2_PA -user netcool -password object00 -process  
BackupObjectServer
```

The NYC_AGG_B ObjectServer is created with the SQL files from the ncw_osreport utility. The ObjectServer structure is the same as before the simulated failure. The ObjectServer contains the events up to the point of the simulated failure, when the export was created. However, the primary ObjectServer was running during the time of the simulated failure. The events in the backup ObjectServer no longer match the events in the primary ObjectServer.

12. Start the ObjectServer synchronizer gateway.

```
nco_pa_start -server HOST2_PA -user netcool -password object00 -process  
BackupGateway
```

When the gateway starts, it synchronizes the contents of the primary and backup ObjectServers. The NYC_AGG_B ObjectServer is now restored.

Backing up and restoring the Web GUI

In the following exercise, you change the Web GUI environment and create an export of the modified environment. Next, you manually remove the change. Then, you use the data from the export to restore the modified environment.



Important: You perform the following exercises on **host2**.

The first step is to change the Web GUI environment. For this exercise, you create a custom filter.

1. Open a Firefox browser.

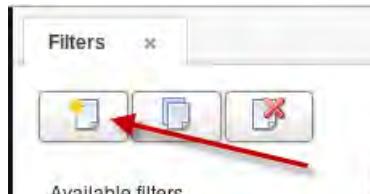


The default browser home page is Dashboard Application Services Hub.

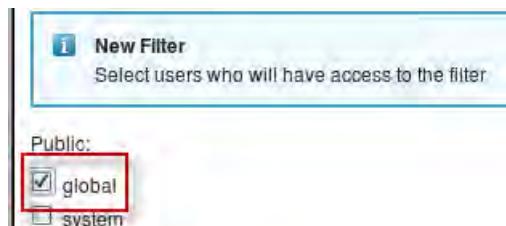
2. Log in as **ncoadmin** with password **object00**.
3. Click the icon and select **Filters**.



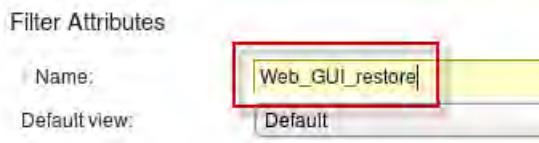
- Click the icon to create a filter.



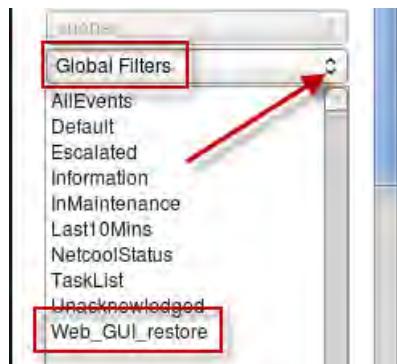
- Select **global** and click **OK**.



- Enter **Web_GUI_restore** for the filter name and click **Save and Close**.



- Change the category to **Global Filters** and verify **Web_GUI_restore** is listed.



Leave the browser as is. You return to it shortly.

The next step is to export the configuration of the Web GUI. For this exercise, you export the complete Web GUI configuration, which is referred to as a *clone*. A property file is used to specify what Web GUI components are exported. The file is configured by default to export everything in the Web GUI configuration.

- Export the Web GUI configuration as follows:

- Change to the target directory.

```
cd /opt/IBM/JazzSM/ui/bin
```

b. Run the export utility.

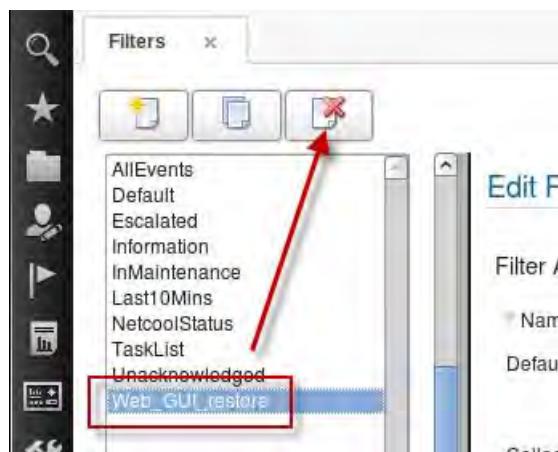
```
./consolecli.sh Export --username smadmin --password object00 --settingFile  
/opt/IBM/netcool/omnibus_webgui/integration/plugins/OMNIbusWebGUI_clone_sett  
ings.properties --excludePlugins TCExportPlugins  
. . .  
CTGWA4232I Starting CMS export...  
CTGWA4236I Archiving the files...  
CTGWA4235I CMS Export completed successfully  
CTGWA4011I The charting data was successfully exported to the directory  
/opt/IBM/JazzSM/ui/output/data.zip.  
CTGWA4017I The command completed successfully.
```

The command creates the following file:

/opt/IBM/JazzSM/ui/output/data.zip

9. Return to the Firefox browser and the Filters page.

10. Click **Web_GUI_restore** and click the red X icon to delete the filter.



11. Click **OK** to confirm the delete.

Are you sure you want to delete Filter "Web_GUI_restore"?



The filter is deleted.

12. Log out of Dashboard Application Services Hub.



Leave the browser as is. You return to it shortly.

Import the data.zip file to restore the filter.



Important: The data.zip file includes the complete Web GUI configuration, not just the filter definition.

The import utility expects the data.zip file in a specific directory.

13. Copy the data.zip file as follows:

- Create the input directory.

```
cd /opt/IBM/JazzSM/ui/  
mkdir input
```

- Change to the target directory.

```
cd /opt/IBM/JazzSM/ui/output
```

- Copy the file.

```
cp data.zip ../input
```

14. Import the Web GUI configuration as follows:

- Change to the target directory.

```
cd /opt/IBM/JazzSM/ui/bin
```

b. Run the import utility.

```
./consolecli.sh Import --username smadmin --password object00 --settingFile  
/opt/IBM/netcool/omnibus_webgui/integration/plugins/OMNIbusWebGUI_clone_sett  
ings.properties --excludePlugins TCRImportPlugins  
. . .  
CTGWA7099I Backup file saved to  
/opt/IBM/JazzSM/ui/profiles/JazzSMProfile/backups.  
CTGWA4240I CMS data source is used to work with remote launch entries and is  
an optional setting. It will not be created since required parameters are  
missing  
CTGWA4017I The command completed successfully.
```



Note: The import command syntax is not the same as the export command. One important difference is the excludePlugins line.

15. Restart Dashboard Application Services Hub.

a. Stop the server.

```
cd /opt/IBM/JazzSM/profile/bin/  
./stopServer.sh server1 -user smadmin -password object00
```

Wait for the server to stop.

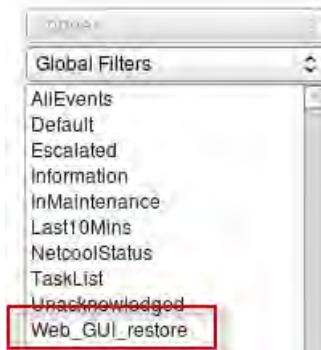
b. Start the server.

```
./startServer.sh server1
```

Wait for the server to start.

16. Return to the Firefox browser and log in as **ncoadmin** with password **object00**.

17. Open the Filters page and view the list of global filters.



The Web_GUI_restore filter appears in the list.

18. Log out of Dashboard Application Services Hub.

19. Close the Firefox browser.

The previous steps demonstrated how to clone the contents of Web GUI. The same technique can be used to clone a copy Dashboard Application Services Hub. The only change that is required is to use a different property file.

```
/opt/IBM/netcool/omnibus_webgui/integration/plugins/OMNIbusWebGUI_DASH_clone.properties
```

The commands to export the configuration are as follows:

```
cd /opt/IBM/JazzSM/ui/bin
```

```
./consolecli.sh Export --username smadmin --password object00 --settingFile  
/opt/IBM/netcool/omnibus_webgui/integration/plugins/OMNIbusWebGUI_DASH_clone_settings.properties
```

The commands to import the configuration are as follows:

```
cd /opt/IBM/JazzSM/ui/bin
```

```
./consolecli.sh Import --username smadmin --password object00 --settingFile  
/opt/IBM/netcool/omnibus_webgui/integration/plugins/OMNIbusWebGUI_DASH_clone_settings.properties
```

Exercise 4 Applying maintenance

In the following exercise, you install a Netcool/OMNIbus core fix pack.



Important: You perform the following exercises on **host1**.

1. Shut down Netcool/OMNIbus core components on host1.

- a. Enter the following command to shut down process activity on host1.

```
nco_pa_shutdown -server HOST1_PA -user netcool -password object00
```

Connected To PA Server [HOST1_PA] Shutdown Options :-

- 1) Shutdown Server leaving managed processes running.
- 2) Shutdown Server and stop all managed processes.
- 3) Exit shutdown interface.

Select Option [1-3] 2

- b. Enter **2** to shut down everything.

2. Expand the fix pack installation file.

- Change to the target directory.

```
cd /software/maintenance/
```

- Expand the installation file.

```
unzip 8.1.0-TIV-OMNibusCore-linux-x86_64-FP0001.zip
```

3. Review the fix pack readme file to determine how to install the fix pack



Important: You are using the Electronic Software Delivery Zip File method.

```
more 8.1.0.1-TIV-NCOMNibus-FP0001.README
```

```
.
```

```
.
```

```
.
```

Electronic Software Delivery(ESD) Zip File

If you have downloaded the ESD for Fix Pack 1 then:

- Create a directory to store the files and extract the contents on the zip file into this.
- If you have installed Installation Manager using either Administrator or Nonadministrator mode, run the `update_console` or `update_gui` script to launch Installation Manager into the update panel without the requirement of configuring the update repository.
- If the script fails or Installation Manager is installed in Group mode, then you must configure the update repository as in the Local Repository above, with the new repository being the extracted OMNibusRepository/composite directory.
- Select the "IBM Tivoli Netcool OMNibus" entry from the list of packages to find updates for.

4. Install the fix pack as follows.

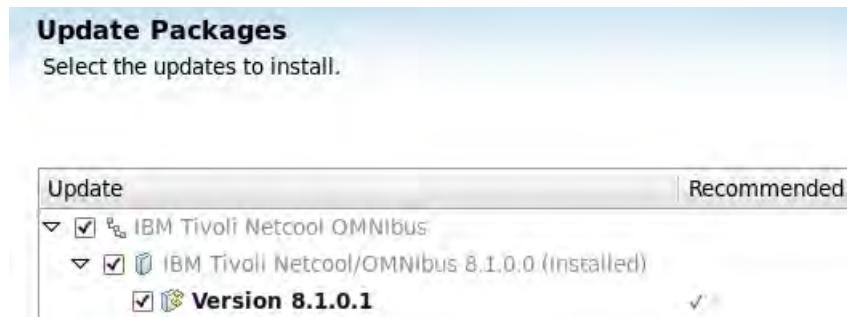
```
./update_gui.sh
```

IBM Installation Manager starts.

- Click **Next**.



- Click **Next**.



- Accept the license agreement and click **Next**.

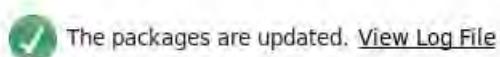
- Leave the selected features and click **Next**.

- Click **Update** to start the installation.



Note: The installation runs for several minutes.

- Verify whether the installation is successful and click **Finish**.



The following update was installed:

Update	Installation Directory
<div style="display: flex; align-items: center;"> ▼ <input checked="" type="checkbox"/> IBM Tivoli Netcool OMNibus </div>	
<div style="display: flex; align-items: center;"> <input checked="" type="checkbox"/> IBM Tivoli Netcool/OMNibus 8.1.0.1 </div>	/opt/IBM/tivoli/netcool

- Review the fix pack readme file to determine whether there are any postinstallation steps required.

more 8.1.0.1-TIV-NCOMNIBUS-FP0001.README



Note: There are no postinstallation steps for this fix pack.

- Restart the core components.

nco_pad -name HOST1_PA

7. Verify the status of the components.

```
nco_pa_status -server HOST1_PA -user netcool -password object00
```

Service Name	Process Name	Hostname	User	Status	PID
Core	MasterObjectServer	host1.tivoli.edunetcool		RUNNING	14697
	SyslogProbe	host1.tivoli.edunetcool		RUNNING	14698
	SnmpProbe	host1.tivoli.edunetcool		RUNNING	14699

8. Display the current version of Netcool/OMNIbus software.

```
nco_id
```

```
Netcool/OMNIbus 8.1.0 - May 2014
NCHOME: /opt/IBM/tivoli/netcool
IMHOME: /home/netcool/IBM/InstallationManager/eclipse
IMDATA: /home/netcool/var/ibm/InstallationManager
```

Products:

```
IBM Tivoli Netcool/OMNIbus - 8.1.0
IBM_Tivoli_Netcool_OMNIbus - 8.1.0.1
```

Fix Packs:

```
IBM_Tivoli_Netcool_OMNIbus - 8.1.0.1
```

The text 8.1.0.1 indicates that Netcool/OMNIbus version 8.1 is installed, and fix pack 1 is applied.

9. Remove the fix pack installation files to conserve disk space.

```
cd /software
```

```
/bin/rm -R maintenance
```

The process is the same for applying maintenance to Web GUI. The maintenance for Web GUI is packaged as a separate file.

Exercise 5 Modifying behavior

In the following exercise, you change some of the basic behavior of the ObjectServer and Web GUI.

Modifying ObjectServer behavior

Changing a non-dynamic property



Important: You perform the following exercises on **host2**.

In a production environment, client components might connect to the ObjectServer from outside a firewall. In this type of deployment, you must configure a static IDUC listening port number. This port number is then added to the firewall configuration. In the classroom configuration, the ObjectServer is running as a non-root user. Therefore, the IDUC port number must be greater than 1024. For this exercise, you use 9500.

1. Verify whether port 9500 is in use.

```
netstat -na | grep 9500
```

The port is not in use.

2. Modify the ObjectServer property file as follows:

- a. Change to the target directory.

```
cd /opt/IBM/tivoli/netcool/omnibus/etc
```

- b. Save a copy of the file before modification.

```
cp NYC_AGG_B.props NYC_AGG_B.props.orig
```

- c. Modify the property file.

```
gedit NYC_AGG_B.props
```

- d. Locate the last occurrence of the following line:

```
IDuc.ListeningPort: 0 # INTEGER (IDuc port to listen on.)
```



Important: The line appears more than once in the file. You need the entry with no comment character.

- e. Change the value from 0 to 9500 as shown here:

```
IDuc.ListeningPort: 9500 # INTEGER (IDuc port to listen on.)
```

- f. Save the changes and exit gedit.

3. Restart the ObjectServer.

- a. Stop the ObjectServer.

```
nco_pa_stop -server HOST2_PA -process BackupObjectServer -user netcool  
-password object00
```

- b. Start the ObjectServer.

```
nco_pa_start -server HOST2_PA -process BackupObjectServer -user netcool  
-password object00
```

The ObjectServer is now configured to use 9500 as the static IDUC listening port.

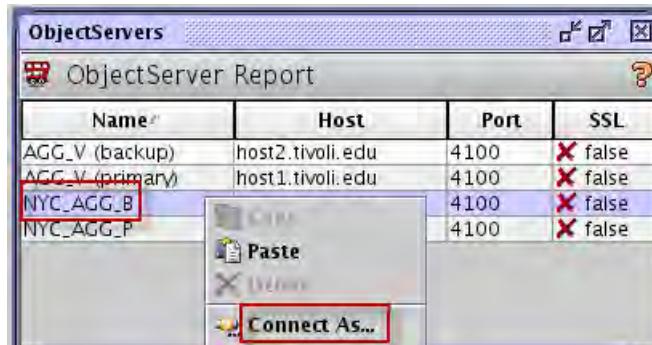
Changing a dynamic property

The IDUC listening property is one that cannot be modified dynamically. It requires an ObjectServer restart to take effect. Other property values can be modified dynamically.

1. Start the Netcool/OMNIbus Administrator utility.

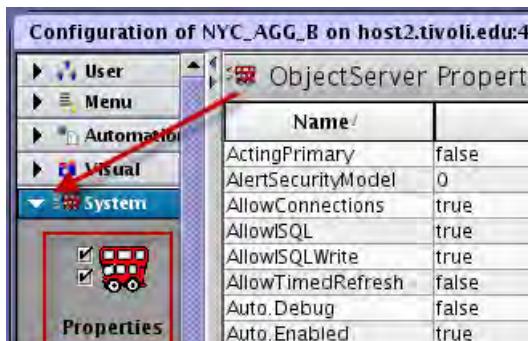
```
nco_config &
```

2. Right-click **NYC_AGG_B** and select **Connect As...**.



3. Log in as **root** with password **object00**.



4. Expand **System** and click **Properties**.

5. Scroll down in the list and observe the settings for Iduc.ListeningPort.

ObjectServer Properties					
Name	Value	Description	Editable	Immediate	
Hostname	host2.tivoli.edu	ObjectServer's hostname	<input checked="" type="checkbox"/>	false	<input checked="" type="checkbox"/>
Iduc.ListeningHostn...		Hostname to listen for Iduc ...	<input checked="" type="checkbox"/>	true	<input checked="" type="checkbox"/>
Iduc.ListeningPort	9500	Iduc port to listen on.	<input checked="" type="checkbox"/>	true	<input checked="" type="checkbox"/>
Ipct.QueueSize	1024	Size of middleware internal...	<input checked="" type="checkbox"/>	true	<input checked="" type="checkbox"/>

The value is set to 9500, which was configured in the previous step. The value in the column that is labeled Immediate is false, which is an indication that the property cannot be changed dynamically. You must restart the ObjectServer.

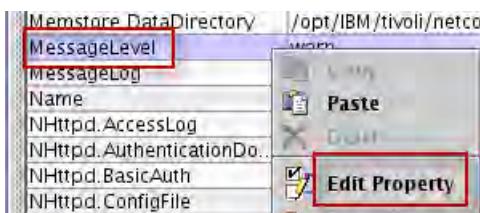
6. Scroll down in the list and observe the settings for MessageLevel.

ObjectServer Properties					
Name	Value	Description	Editable	Immediate	
LogFileFormat	true	Use stream for logging.	<input checked="" type="checkbox"/>	true	<input checked="" type="checkbox"/>
MaxLogFileSize	1024	Maximum log file size in kby...	<input checked="" type="checkbox"/>	true	<input checked="" type="checkbox"/>
Memstore.ChainLockRatio	32	Ratio of locks to chains for r...	<input checked="" type="checkbox"/>	true	<input checked="" type="checkbox"/>
Memstore.DataDirectory	/opt/IBM/tivoli/netcool/o...	Memory storage directory	<input checked="" type="checkbox"/>	true	<input checked="" type="checkbox"/>
MessageLevel	warn	Message reporting level	<input checked="" type="checkbox"/>	true	<input checked="" type="checkbox"/>
MessageLog	/opt/IBM/tivoli/netcool/o...	Path to the message log file.	<input checked="" type="checkbox"/>	true	<input checked="" type="checkbox"/>

MessageLevel determines how verbose the messages are that are produced by the ObjectServer. The default value is warn. The MessageLevel property can be modified dynamically. You change the MessageLevel to debug when you troubleshoot certain ObjectServer issues.

7. Modify the MessageLevel property as follows:

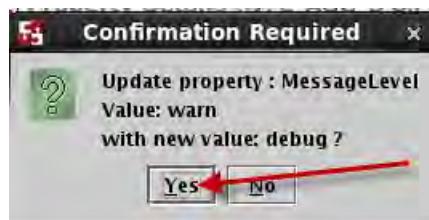
- Right-click **MessageLevel** and select **Edit Property**.



8. Change the value to **debug** and click **OK**.



9. Click **Yes** to confirm the change.



The logging level is now set to debug.

Leave the Administrator utility as is. You return to it shortly.

10. Examine the contents of the ObjectServer log file.

```
cd /opt/IBM/tivoli/netcool/omnibus/log
```

```
tail NYC_AGO_B.log
```

```
[netcool@host2 log]$ tail NYC_AGO_B.log
2014-12-02T20:11:43: Debug: D-ETC-004-018: Calling 1 callback(s) for event 9
2014-12-02T20:11:43: Debug: D-ETC-004-018: Calling 1 callback(s) for event 11
2014-12-02T20:11:43: Debug: D-ETC-004-018: Calling 1 callback(s) for event 12
2014-12-02T20:11:43: Information: I-OBX-104-016: Profiler timing submitted from
connection ID 2: 0.000050s
2014-12-02T20:11:43: Debug: D-ETC-004-034: "CLOCK tick" handler called
2014-12-02T20:11:43: Debug: D-ETC-004-019: CLOCK thread: sleeping ...
2014-12-02T20:11:44: Debug: D-ETC-004-034: "CLOCK tick" handler called
2014-12-02T20:11:44: Debug: D-ETC-004-019: CLOCK thread: sleeping ...
2014-12-02T20:11:45: Debug: D-ETC-004-034: "CLOCK tick" handler called
2014-12-02T20:11:45: Debug: D-ETC-004-019: CLOCK thread: sleeping ...
[netcool@host2 log]$
```

Log messages indicate the debug level.

11. Return to the Administrator utility and set MessageLevel back to **warn**.

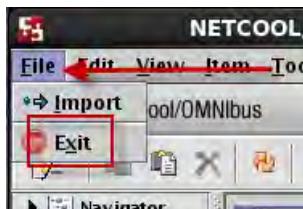
The screenshot shows the 'ObjectServer Properties' dialog. It is a table with two columns: 'Name' and 'Value'. The 'MessageLevel' row is highlighted with a red box. The 'Value' column for 'MessageLevel' contains the word 'warn'.

Name	Value
CognosAgent	true
MaxLogFileSize	1024
Memstore.ChainLockRatio	32
Memstore.DataDirectory	/opt/IBM/tivoli/netcool/o...
MessageLevel	warn
MessageLog	/opt/IBM/tivoli/netcool/o...



Important: Debug level messages should be enabled only when you troubleshoot an issue. When you run the ObjectServer with debug level messaging, it requires more ObjectServer overhead.

12. Exit the Administrator utility.



When you use the Administrator utility to modify a property value, the utility updates the property file on disk with the corresponding change. The utility writes the file to disk when you exit the utility.



2 ObjectServer administration exercises

In this unit you learn how to add extra columns to the Netcool/OMNIbus event record. You updated the ObjectServer gateway to include the extra columns. And, you create an ObjectServer.

Exercise 1 Modifying the event record

The classroom environment consists of two ObjectServers: NYC_AGG_P and NYC_AGG_B. The ObjectServers are configured as a high availability pair, and synchronized with an ObjectServer gateway: NYC_AGG_GATE. In the following steps, you add extra columns to the event record definition in both ObjectServers. You also update the gateway configuration so that the extra columns are synchronized between the ObjectServers. To facilitate this exercise, a file is provided that contains the SQL commands to add the column names.



Important: You perform the following steps on host2.

1. Open a Terminal window if necessary.
2. Make sure the NYC_AGG_P ObjectServer is available.

```
nco_ping NYC_AGG_P  
NCO_PING: Server available.
```

3. Make sure the NYC_AGG_B ObjectServer is available.

```
nco_ping NYC_AGG_B  
NCO_PING: Server available.
```

4. Change to the target directory.

```
cd /workshop/unit02
```

5. Examine the supplied SQL file.

```
more nyc_agg.sql
```

The file contains SQL commands to add a number of columns to the alerts.status table. The use of a file like this is a common technique to facilitate the addition of multiple columns with minimal effort.

6. Import the file into NYC_AGG_P.

```
nco_sql -server NYC_AGG_P -user root -password object00 < nyc_agg.sql  
(0 rows affected)  
(0 rows affected)
```

 **Note:** The messages are normal and can be ignored.

The NYC_AGG_P ObjectServer is running on host1. You modified the ObjectServer with a command-line utility that ran on host2.

7. Import the file into NYC_AGG_B.

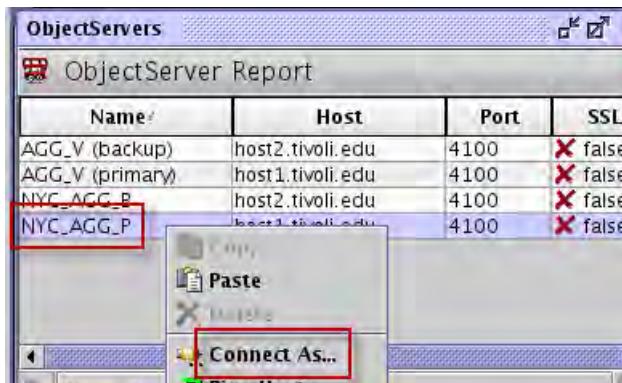
```
nco_sql -server NYC_AGG_B -user root -password object00 < nyc_agg.sql  
(0 rows affected)  
(0 rows affected)
```

8. Verify the changes to NYC_AGG_P.

a. Start the Netcool/OMNIbus Administrator utility.

```
nco_config &
```

- b. Right-click **NYC_AGG_P** and select **Connect As...**.



- c. Log in as **root** with password **object00**.
d. Expand **System** and click **Databases**.
e. Expand **alerts**, scroll down and click **status**.

Name	Data Type	Length	Precision
Acknowledged	Integer	4	X ...
AdvCorrCauseType	Integer	4	X ...
AdvCorrServer	VarChar	64	X ...
AdvCorrServerS...	Integer	4	X ...
AENMsg	VarChar	20	X ...
AgencyId	VarChar	10	X ...
Agent	VarChar	64	X ...
AggregationFirst	UTC	4	X ...
AlertGroup	VarChar	255	X ...
AlertKey	VarChar	255	X ...
Archived_Flag	Integer	4	X ...
BSM_Identity	VarChar	10...	X ...
CauseType	Integer	4	X ...
Class	Integer	4	X ...
CollectionFirst	UTC	4	X ...
ContactEmail	VarChar	128	X ...

The list of column names opens.

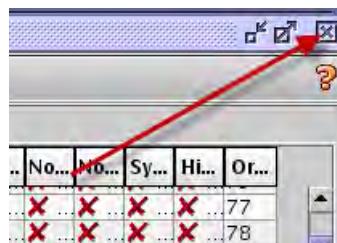
You are able to connect to the NYC_AGG_P ObjectServer, which is running on host1, with the administrator utility on host2.

- f. Examine the list of names and verify that the following names appear:

AENMsg
 AgencyId
 ContactName
 ContactEmail
 ContactPhone
 SiteAddr
 SiteCity
 SiteId
 SiteName
 SiteState
 SiteCountry
 TicketID

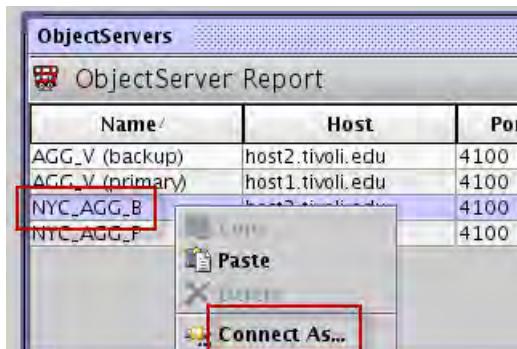
Name	Data Type	Le...	Pri...	No...	No...	Sy...	Hi...
AENMsg	VarChar	20	X	X	X	X	X
AgencyId	VarChar	10	X	X	X	X	X
Agent	VarChar	64	X	X	X	X	X
AggregationFirst	UTC	4	X	X	X	X	X
AlertGroup	VarChar	255	X	X	X	X	X
AlertKey	VarChar	255	X	X	X	X	X
Archived_Flag	Integer	4	X	X	X	X	X
BSM_Identity	VarChar	10...	X	X	X	X	X
CauseType	Integer	4	X	X	X	X	X
Class	Integer	4	X	X	X	X	X
CollectionFirst	UTC	4	X	X	X	X	X
ContactEmail	VarChar	128	X	X	X	X	X
ContactName	VarChar	128	X	X	X	X	X
ContactPhone	VarChar	128	X	X	X	X	X

- g. Click the X to close the connection to NYC_AGG_P.



- h. Click Yes to close the connection.

9. Verify the changes to NYC_AGG_B.
 - a. Right-click NYC_AGG_B and select **Connect As...**.



- b. Log in as **root** with password **object00**
 - c. Repeat the previous steps to verify the column names.
- You are able to connect to NYC_AGG_P and NYC_AGG_B from the same instance of the administrator utility.

10. Click **File** and select **Exit** to close the administrator utility.



Important: The ObjectServer modifications did not require a restart.

Exercise 2 Modifying the gateway

In a high availability configuration, the gateway synchronizes changes between the two ObjectServers. One type of change is a change to the event record. This includes the addition of a record or a change to an existing record. The gateway uses a configuration file to identify which column names are synchronized. You added extra column names in the previous exercise. Now, you update the gateway configuration file to add those column names.



Important: You perform the following steps on host2.

1. Use process activity to stop the gateway.

- a. Determine the process name.

```
nco_pa_status -server HOST2_PA -user netcool -password object00
```

Service Name	Process Name	Hostname	User	Status	PID
Core	BackupObjectServer	host2.tivoli.edunetcool	netcool	RUNNING	2185
	BackupGateway	host2.tivoli.edunetcool	netcool	RUNNING	6848
	ArchiveGateway	host2.tivoli.edunetcool	netcool	RUNNING	2187

- b. Stop the process.

```
nco_pa_stop -server HOST2_PA -user netcool -password object00 -process
BackupGateway
```

- c. Verify whether the gateway is stopped.

```
nco_pa_status -server HOST2_PA -user netcool -password object00
```

Service Name	Process Name	Hostname	User	Status	PID
Core	BackupObjectServer	host2.tivoli.edunetcool	RUNNING	2185	
	BackupGateway	host2.tivoli.edunetcool	DEAD	0	
	ArchiveGateway	host2.tivoli.edunetcool	RUNNING	2187	

2. Modify the gateway map file.

- a. Change to the target directory.

```
cd /opt/IBM/tivoli/netcool/omnibus/etc
```

- b. Save a copy of the file before changes.

```
cp AGG_GATE.map AGG_GATE.map.orig
```



Hint: You can find the name and location of the map file in the gateway property file.

```
more NYC_AGG_GATE.props
```

```
.
```

```
.
```

```
.
```

```
Gate.MapFile: '$OMNIHOME/etc/AGG_GATE.map'
```

- c. Edit the file with the gedit utility.

```
gedit AGG_GATE.map
```

- d. Locate the following section:

```
#####
# CUSTOM alerts.status FIELD MAPPINGS GO HERE
#
'Archived_Flag'      =      '@Archived_Flag',
'Delete_Flag'        =      '@Delete_Flag',|
```

- e. Add the following lines below **Delete_Flag**.

```
'AENMsg'      ='@AENMsg',
'AgencyId'    ='@AgencyId',
>ContactName' ='@ContactName',
>ContactEmail' ='@ContactEmail',
>ContactPhone' ='@ContactPhone',
'SiteAddr'     ='@SiteAddr',
'SiteCity'     ='@SiteCity',
'SiteId'       ='@SiteId',
'SiteName'     ='@SiteName',
'SiteState'    ='@SiteState',
'SiteCountry'  ='@SiteCountry',
'TicketID'     ='@TicketID',
```

```
#####
'Archived_Flag'      ='@Archived_Flag',
'Delete_Flag'        ='@Delete_Flag',
'AENMsg'              ='@AENMsg',
'AgencyId'            ='@AgencyId',
>ContactName'         ='@ContactName',
>ContactEmail'        ='@ContactEmail',
>ContactPhone'        ='@ContactPhone',
'SiteAddr'             ='@SiteAddr',
'SiteCity'             ='@SiteCity',
'SiteId'               ='@SiteId',
'SiteName'             ='@SiteName',
'SiteState'            ='@SiteState',
'SiteCountry'          ='@SiteCountry',
'TicketID'             ='@TicketID',|
```

- f. Save the file and exit gedit.

3. Use process activity to start the gateway.

- a. Start the process.

```
nco_pa_start -server HOST2_PA -user netcool -password object00 -process
BackupGateway
```

- b. Verify whether the gateway is started.

```
nco_pa_status -server HOST2_PA -user netcool -password object00
```

Service Name	Process Name	Hostname	User	Status	PID
Core	BackupObjectServer	host2.tivoli.edunetcool		RUNNING	2185
	BackupGateway	host2.tivoli.edunetcool		RUNNING	15608
	ArchiveGateway	host2.tivoli.edunetcool		RUNNING	2187

If the gateway is not running, it generally means a syntax error in the modified map file. This list shows the most common errors:

- Misspelled column name.
- Missing comma.
- Missing quotation mark.

Examine the gateway log file.

```
more /opt/IBM/tivoli/netcool/omnibus/log/NYC_AGG_GATE.log
```

Correct any issues and restart the gateway.

Exercise 3 Creating an ObjectServer

In this exercise you create an ObjectServer that is based on the best practice configuration for an aggregation ObjectServer. You use the custom SQL file included with Netcool/OMNIbus in the multitier directory.



Important: You perform the following steps on host2.

1. Create the ObjectServer.

- a. Change to the target directory.

```
cd /opt/IBM/tivoli/netcool/omnibus/extensions/multitier/objectserver
```

- b. Run the nco_dbinit utility.

```
nco_dbinit -server LON_AGG_P -customconfigfile aggregation.sql
```

The framework for the LON_AGG_P ObjectServer is created.

2. Add the ObjectServer to the interfaces file.

- a. Run the nco_xigen utility.

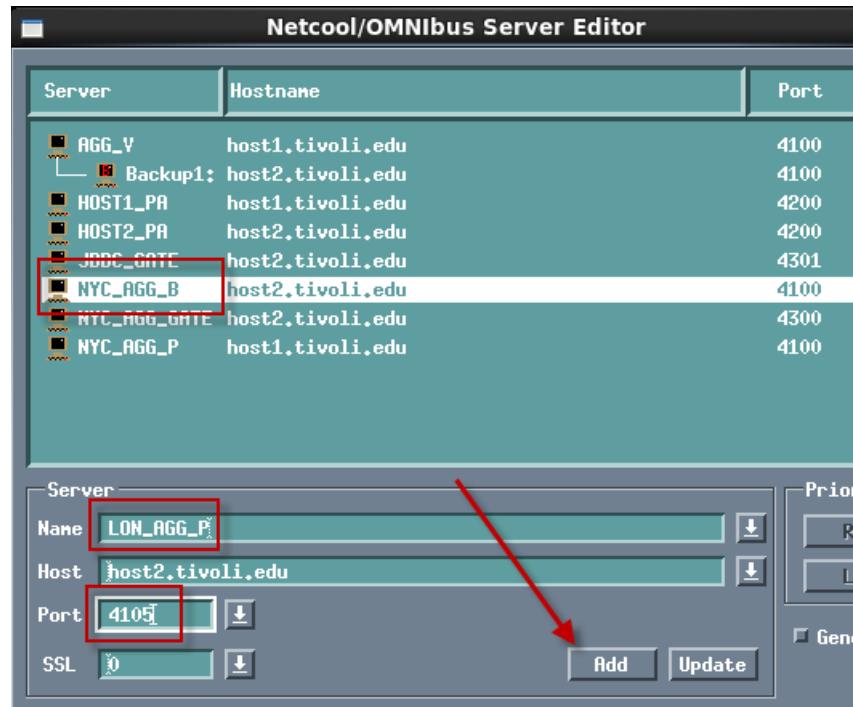
```
nco_xigen &
```

The Netcool/OMNIbus Server Editor opens.

- b. Click **NYC_AGG_B** to select it.

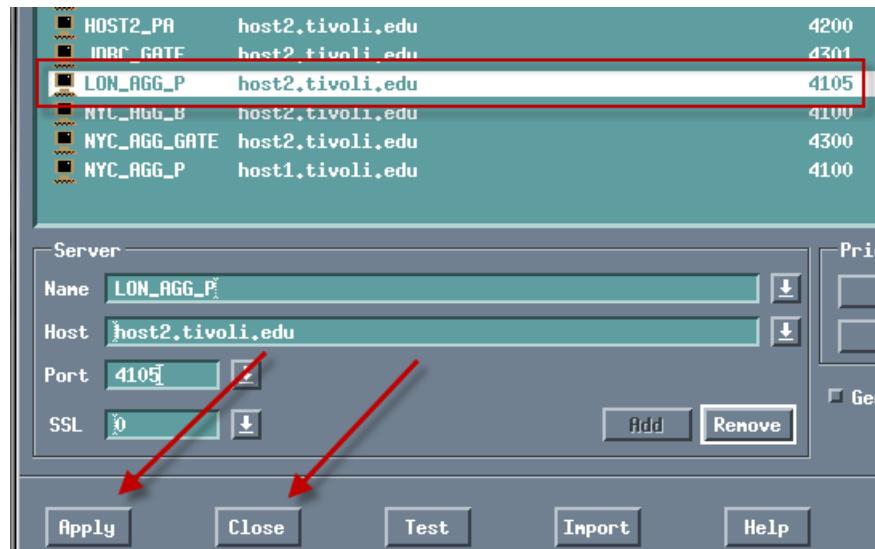
- c. In the lower portion of the window, change the ObjectServer name to **LON_AGG_P**.

- d. Change the port number to **4105** and click **Add**.



Important: Make sure you click **Add**. If you click **Update**, you change the entry for NYC_AGG_B.

- e. Verify whether the entry is correct. Click **Apply** and click **Close**.

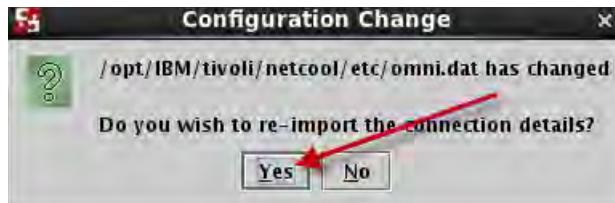


3. Add the ObjectServer to process activity management.

- a. Start the administrator utility.

```
nco_config &
```

- b. Click **Yes** to run the import wizard.

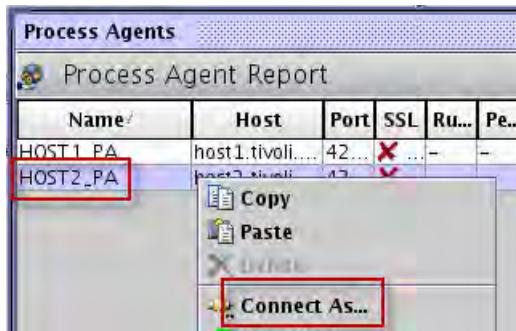


- c. Click **Finish**.

The **LON_AGG_P** ObjectServer is added to the list of ObjectServers.

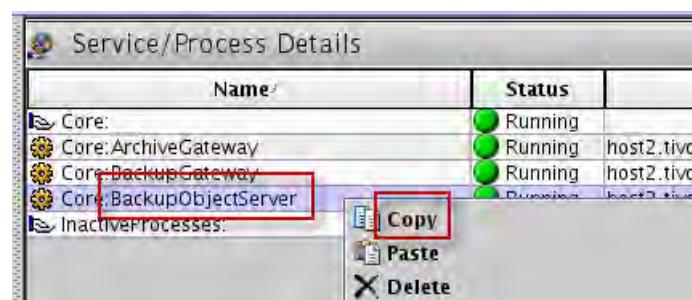
ObjectServers			
ObjectServer Report			
Name	Host	Port	
AGG_V (backup)	host2.tivoli.edu	4100	X
AGG_V (primary)	host1.tivoli.edu	4100	X
LON_AGG_P	host2.tivoli.edu	4105	X
NYC_AGG_B	host2.tivoli.edu	4100	X
NYC_AGG_P	host1.tivoli.edu	4100	X

- d. In the Process Agents box, right-click **HOST2_PA** and select **Connect As...**.



- e. Log in as **netcool** with password **object00**.

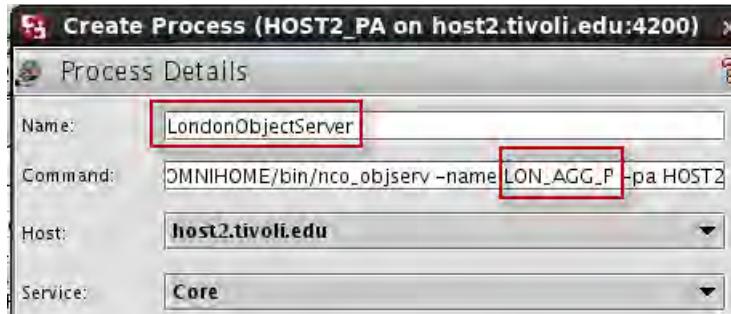
- f. Right-click **BackupObjectServer** and select **Copy**.



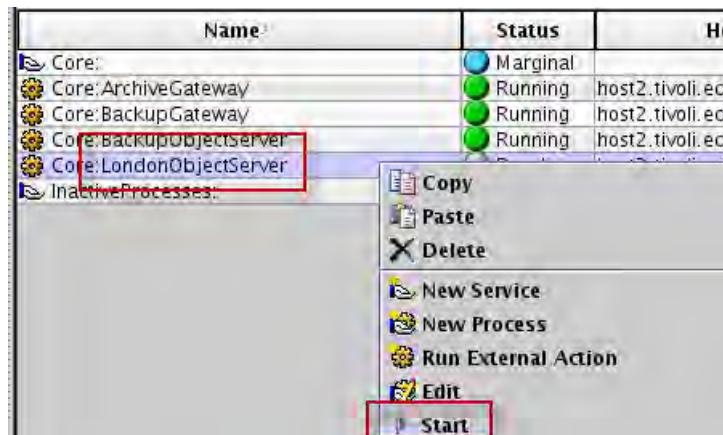
- g. Right-click and select **Paste**.

- h. Enter **LondonObjectServer** for the process name.

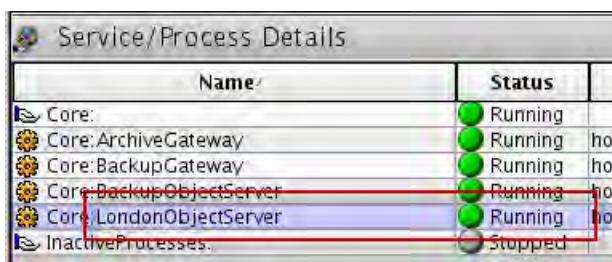
- i. Change the ObjectServer name to **LON_AGG_P** and click **OK**.



- j. Right-click **LondonObjectServer** and select **Start**.



- k. Verify whether the ObjectServer is running.



- l. Click **File** and select **Exit**.
m. Click **Yes** to confirm exit.
n. Click **Yes** to confirm saving the changes
o. Click **OK**.

The LON_AGG_P ObjectServer is created and running. The ObjectServer is managed with process activity.

When an ObjectServer is created, the *root* user is defined with no password. The following steps use a command-line utility to add a password to that user.

4. Add a password to the LON_AGG_P **root** user as follows.

- a. Connect to the ObjectServer with the nco_sql utility:

```
nco_sql -server LON_AGG_P -user root -password ''
```



Important: The value for password in the command that is shown is *two single quotation marks* (''). This syntax indicates a *blank* password.

- b. Enter the following commands that are shown in bold text:

```
1> alter user 'root' set password 'object00';
2> go
(0 rows affected)
1> quit
```

The password for the *root* user is now **object00** on the LON_AGG_P ObjectServer.

- c. Verify that the password is correct:

```
nco_sql -server LON_AGG_P -user root -password 'object00'
```

```
1> quit
```

The prompt characters (1>) indicate that the utility is able to connect to the ObjectServer with the revised password. Enter quit to exit the utility.

Add the same custom columns to the LON_AGG_P ObjectServer.

5. Change to the target directory.

```
cd /workshop/unit02
```

6. Import the file into LON_AGG_B.

```
nco_sql -server LON_AGG_P -user root -password object00 < nyc_agg.sql
(0 rows affected)
```



Note: The messages are normal and can be ignored.



3 Probes exercises

The exercises in this unit are designed to demonstrate a few of the advanced features of probes. You learn how to import a MIB file, and generate a rules file. You incorporate the rules file into the Netcool Knowledge Library. You verify that the generated rules file can successfully generate an event from an SNMP trap. You learn how to configure the peer-to-peer feature for implementing high availability. You configure the optional procedure that you use to reload the rules file for all active probes.

Exercise 1 MIB Manager

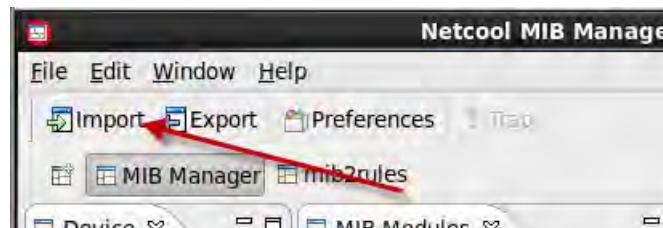
In this exercise, you import a MIB file into the MIB manager utility, and generate rules for use with the SNMP probe.



Important: You perform the following exercise on **host1**.

Generating rules

1. Open a Terminal window if necessary.
2. Start MIB manager.
nco_mibmanager &
3. Click **Import**.



- Browse to the following directory, and click Import.

/workshop/unit03



The utility processes all files within the directory. If you select **Traverse Subdirectories**, the utility processes files in any subdirectories as well.



Note: The class image directory contains a single MIB file: **adsmser.mib**.

- When the import is complete, click **OK** to close the status window.



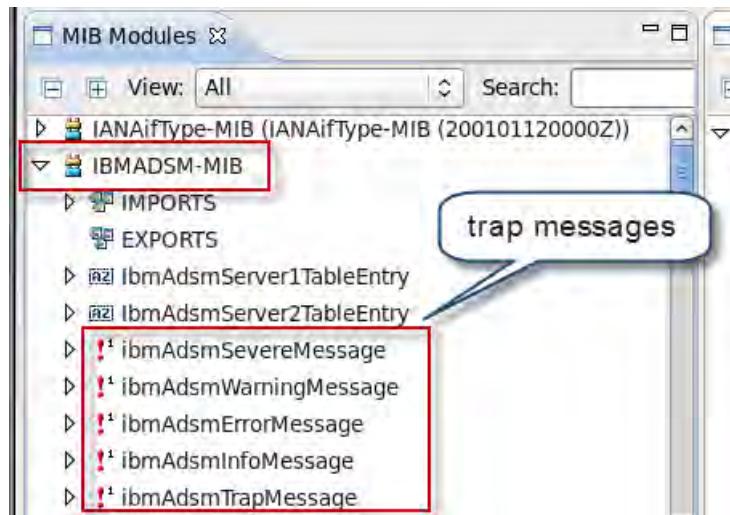
- Review the summary, and verify whether trap objects are found.

Import Complete			Imports	
Object Type	Found	Process		
MIB Modules	1	1	SNMPv2-CONF	✓
Total MIB Objects	79	79	SNMPv2-SMI	✓
TRAP-TYPE	5	5	SNMPv2-TC	✓
NOTIFICATION-TYPE	0	0		
TEXTUAL-CONVENTION	2	2		
MACRO	0	0		
OBJECT IDENTIFIER	10	10		
OBJECT-TYPE	35	35		

Not all MIB files contain trap definitions. If the file does not contain trap definitions, then you cannot use it to generate rules. The trap definitions are identified as either TRAP-TYPE or NOTIFICATION-TYPE objects.

- Click **Dismiss** to close the status window.

8. Expand **IBMADSM-MIB**.



9. Click **ibmAdsmSevereMessage** to select it.



The details for the trap appear in the lower pane.

10. Scroll down and locate the value for **Severity**.



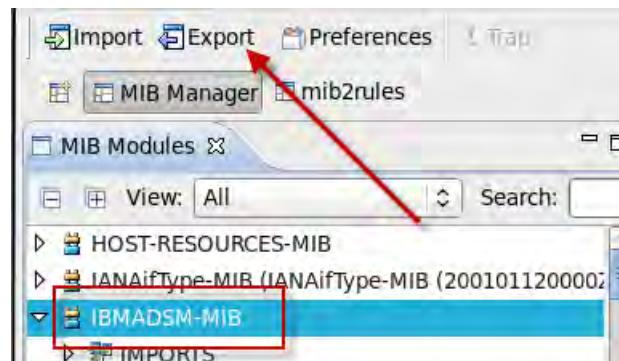
The MIB manager utility generates rules with default values for certain event record column names. You see the values for three of the column names here. In most cases, when you generate rules with the MIB manager utility, you adjust the values for Severity, and Type. The utility is not able to determine whether a trap message for any vendor represents a serious issue or not. The utility assumes that every trap is considered a minor issue.

You can change the value in the MIB manager utility before the utility generates the rules. Or you can generate the rules and then change the values.

11. Double-click Severity, and change the setting to **Major**. Click the green check mark to save the change.



12. Click **IBMASDM-MIB** to select it, and then click **Export**.



13. Browse to **/workshop/unit03**. Select **Netcool Knowledge Library version 3.x** for file type. Select **Selected Subtree(s) only**. Click **Export**.



The directory setting defines where the utility saves the output files.

14. Verify the export status, and click **OK** to close the window.



The status message indicates that five objects are exported. Five trap messages are contained in the selected MIB.

15. Select **File > Exit** to close the MIB manager utility.

Adding rules to Netcool Knowledge Library

The MIB manager utility generates rules from a vendor-specific MIB file. The Netcool Knowledge Library contains rules for a number of vendors. The process to add rules from MIB manager varies whether the vendor is included in Netcool Knowledge Library or not.

The classroom exercise uses an IBM MIB file. IBM is a supported vendor in the Netcool Knowledge Library.

1. Examine the files that MIB manager produces.

```
cd /workshop/unit03
ls -1
20141217203522-nckl_3_0
adsmser.mib
simnet.def
simnet.lookup
```

The adsmser.mib file is used to generate the rules. The MIB manager utility generates the directory with the time stamp and nckl in the name.



Note: Your directory name is not the same as this example.

2. Change to the *time stamp* directory.

```
cd 20141217203522-nckl_3_0/
ls -1
ibm
README
```

The ibm directory contains the generated files.

3. Examine the readme file.

more README

```
MIB Manager NCKL Format Rulesfiles
-----
Instructions For Use:

The rulesfile was designed for use with the NCKL 3.x format rules.
The current IBM recommended and supported rulesfile format
is NCKL. It is highly recommended all users make use of the NCKL
format if at all possible.

To use this NCKL format rulesfile simply place includes for the two
per vendor master files into your snmptrap.rules file.
Those files are named:
    <vendor>/<vendor>.m2r.master.include.lookup
    <vendor>/<vendor>.m2r.master.include.rules

If there already exists vendor specific NCKL rules for the vendor
then simply include the above two m2r.master files in

    <vendor>/<vendor>.master.include.lookup
    <vendor>/<vendor>.master.include.rules

and in this case the <vendor>/<vendor>-preclass.snmptrap.lookup
file should be merged into the existing preclass lookup file.
```

The readme file contains a brief set of instructions for how to include the files in the Netcool Knowledge Library. The IBM vendor exists, so you use the last set of steps.



Important: The Netcool Knowledge Library contains a collection of files that all belong to the IBM vendor. By default, no vendor-specific files are enabled for use. You must configure the Netcool Knowledge Library to enable the use of the IBM files.

4. Enable the use of IBM files as follows:

a. Change to location of the Netcool Knowledge Library files.

```
cd $NC_RULES_HOME
```



Note: The NC_RULES_HOME environment variable must be defined when you use the Netcool Knowledge Library.

b. Save a copy of the master rules file before changes.

```
/bin/rm snmptrap.rules.orig
cp snmptrap.rules snmptrap.rules.orig
```



Note: It is safe to remove the write-protected file.

c. Open the file for edit with the gedit utility.

```
gedit snmptrap.rules
```

- d. Remove the comment character from the following line:

```
||| #include "$NC_RULES_HOME/include-snmptrap/huawei/huawei.master.include.lookup"
||| include "$NC_RULES_HOME/include-snmptrap/IANA/IANA.master.include.lookup"
||| #include "$NC_RULES_HOME/include-snmptrap/ibm/ibm.master.include.lookup"
||| include "$NC_RULES_HOME/include-snmptrap/IEEE/IEEE.master.include.lookup"
```

- e. Remove the comment character from the following line:

```
#include "$NC_RULES_HOME/include-snmptrap/huawei/huawei.master.include.rules"
include "$NC_RULES_HOME/include-snmptrap/IANA/IANA.master.include.rules"
#included "$NC_RULES_HOME/include-snmptrap/ibm/ibm.master.include.rules"
include "$NC_RULES_HOME/include-snmptrap/IEEE/IEEE.master.include.rules"
```

- f. Remove the comment character from the following line:

```
#include "$NC_RULES_HOME/include-snmptrap/hatteras/hatteras-preclass.include.snmptrap.rules"
#include "$NC_RULES_HOME/include-snmptrap/huawei/huawei-preclass.include.snmptrap.rules"
#include "$NC_RULES_HOME/include-snmptrap/IANA/IANA-preclass.include.snmptrap.rules"
#include "$NC_RULES_HOME/include-snmptrap/ibm/ibm-preclass.include.snmptrap.rules"
#include "$NC_RULES_HOME/include-snmptrap/IEEE/IEEE-preclass.include.snmptrap.rules"
```

- g. Save the changes, and exit the gedit utility.

5. Verify whether all lines are changed:

```
cat snmptrap.rules | grep ibm
```

```
[netcool@host1 rules]$ cat snmptrap.rules | grep ibm
include "$NC_RULES_HOME/include-snmptrap/ibm/ibm.master.include.lookup"
# Example: $OPTION_EnableDetails_ibm = "1"
    include "$NC_RULES_HOME/include-snmptrap/ibm/ibm.master.include.rules"
    include "$NC_RULES_HOME/include-snmptrap/ibm/ibm-preclass.include.snmptrap.rules"
```

The comment character is gone from all lines.

6. Verify the syntax of the modified master rules file.

```
nco_p_syntax -server NYC_AGG_P -rulesfile snmptrap.rules
```

The syntax checker probe processes the master rules file, and verifies the syntax. The probe produces numerous messages as it runs. When the probe is complete, scroll back slightly in the output, and locate the following line:

```
2014-12-18T17:13:38: Debug: D-UNK-000-000: Auto-resizing lookup table 'syslogCorrScore'
' with 10271 entries from 127 to 513
2014-12-18T17:13:38: Information: I-UNK-000-000: Rules file syntax OK
2014-12-18T17:13:38: Information: I-UNK-000-000: Disconnecting...
2014-12-18T17:13:38: Debug: D-UNK-000-000: Shutting down Probewatch heartbeat thread.
2014-12-18T17:13:39: Debug: D-UNK-000-000: Probewatch heartbeat thread detected stop r
```

The message indicates that no errors are found.



Important: The process to enable any particular vendor is the same. The only part that varies is the vendor name.

The Netcool Knowledge Library is now configured to use the IBM files that are included with the package.

7. Examine the IBM vendor files in the Netcool Knowledge Library.

```
cd $NC_RULES_HOME/include-snmptrap/ibm
```

```
ls -l
```

```
ibm-METRICALARMTRAP-MIB.include.snmptrap.lookup
ibm-METRICALARMTRAP-MIB.include.snmptrap.rules
ibm-METRICALARMTRAP-MIB.sev.snmptrap.lookup
ibm-METRICALARMTRAP-MIB.user.include.snmptrap.rules
ibm-preclass.include.snmptrap.rules
ibm-preclass.snmptrap.lookup
ibm-QILABS-MIB.include.snmptrap.lookup
ibm-QILABS-MIB.include.snmptrap.rules
ibm-QILABS-MIB-notifications.adv.include.snmptrap.rules
ibm-QILABS-MIB-notifications.user.include.snmptrap.rules
ibm-QILABS-MIB-qiNotifications.adv.include.snmptrap.rules
ibm-QILABS-MIB-qiNotifications.user.include.snmptrap.rules
ibm-QILABS-MIB.sev.snmptrap.lookup
```

The directory contains a number of files. Two files in particular are:

```
ibm-preclass.include.snmptrap.rules
ibm-preclass.snmptrap.lookup
```

8. Examine the preclass rules file.

```
more ibm-preclass.include.snmptrap.rules
```

```
#####
log(DEBUG, "<<<< Entering... ibm-preclass.include.snmptrap.rules >>>>")

if(match(@Type, "2"))
{
    $SOS_AdvCorrCauseType_ibm = 4
}
else
{
    if(exists($SEV_KEY))
    {
        $SOS_AdvCorrCauseType_ibm = lookup($SEV_KEY, ibm_preclass)
    }
    else
    {
        $SOS_AdvCorrCauseType_ibm = lookup($SOS_EventId, ibm_preclass)
    }
}

log(DEBUG, "<<<< Leaving... ibm-preclass.include.snmptrap.rules >>>>")
```

The beginning of the file consists of a collection of comments. The actual rules code appears on the end of the file.

The MIB manager utility generates a file with the same name. The file that the MIB manager utility generates contains the same rules code. The file does not contain the same comments.

9. Examine the preclass lookup file.

```
more ibm-preclass.snmptrap.lookup
```

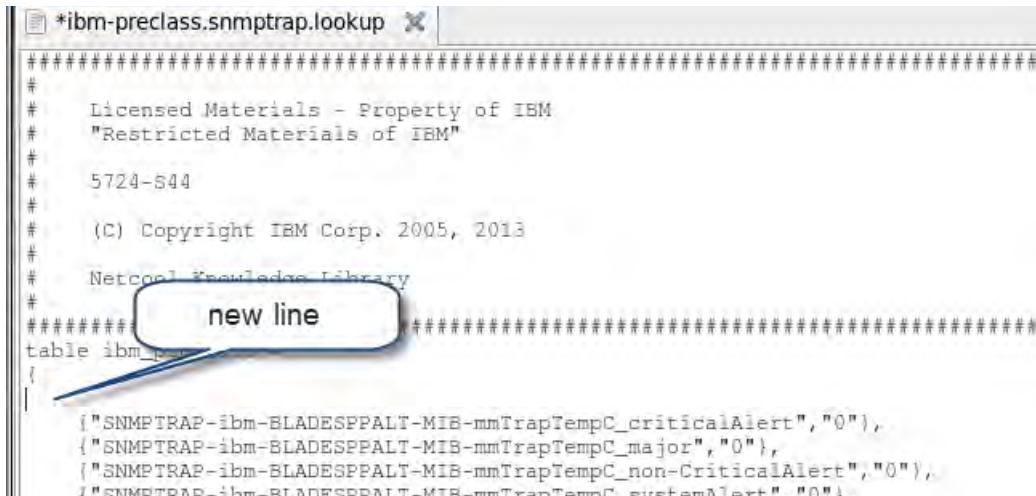
This file contains the preclassification values for the events that are contained in the existing IBM rules files.

The MIB manager utility generates a file with the same name. However, the file from MIB manager includes the preclassification values for only the rules that the utility generates. The values from the MIB manager utility must be merged into the Netcool Knowledge Library file.

10. Open the existing file for edit with the gedit utility.

```
chmod +w ibm-preclass.snmptrap.lookup  
gedit ibm-preclass.snmptrap.lookup &
```

11. Add a blank line as shown in the following screen capture:



```
#  
# Licensed Materials - Property of IBM  
# "Restricted Materials of IBM"  
#  
# 5724-S44  
#  
# (c) Copyright IBM Corp. 2005, 2013  
#  
# Netcool Knowledge Library  
#  
#####  
table ibm_p...  
{  
    {"SNMPTRAP-ibm-BLADESPPALT-MIB-mmTrapTempC_criticalAlert","0"},  
    {"SNMPTRAP-ibm-BLADESPPALT-MIB-mmTrapTempC_major","0"},  
    {"SNMPTRAP-ibm-BLADESPPALT-MIB-mmTrapTempC_non-CriticalAlert","0"},  
    {"SNMPTRAP-ibm-BLADESPPALT-MIB-mmTrapTempC_infoAlert","0"}  
}
```

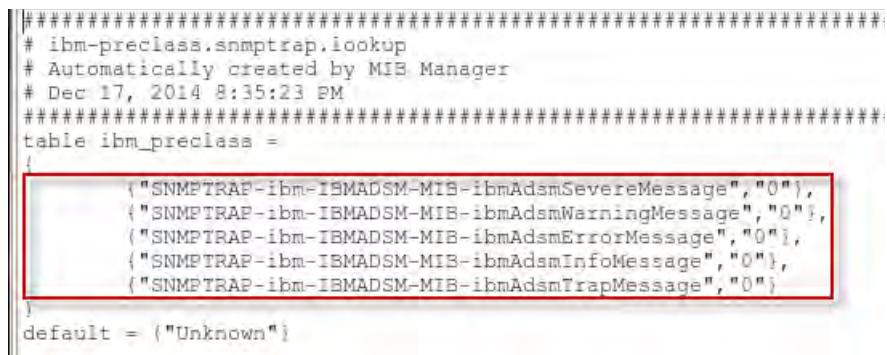
This file creates a table that contains the preclassification values for all trap messages. You must add the definitions that MIB manager created to the beginning of the table.

 **Note:** Leave the gedit window open.

12. Open the MIB manager version of the file for edit with the gedit utility.

```
cd /workshop/unit03/*nckl*/ibm  
gedit ibm-preclass.snmptrap.lookup
```

13. Copy the definitions for the five trap messages.



```
#####  
# ibm-preclass.snmptrap.lookup  
# Automatically created by MIB Manager  
# Dec 17, 2014 8:35:23 PM  
#####  
table ibm_preclass =  
{  
    {"SNMPTRAP-ibm-IBMADSM-MIB-ibmAdsmSevereMessage","0"},  
    {"SNMPTRAP-ibm-IBMADSM-MIB-ibmAdsmWarningMessage","0"},  
    {"SNMPTRAP-ibm-IBMADSM-MIB-ibmAdsmErrorMessage","0"},  
    {"SNMPTRAP-ibm-IBMADSM-MIB-ibmAdsmInfoMessage","0"},  
    {"SNMPTRAP-ibm-IBMADSM-MIB-ibmAdsmTrapMessage","0"}  
}  
default = ("Unknown")
```

14. Paste the five lines into the existing file, and add a *comma* to the end of the last line.

```

#
#####





    {"SNMPTRAP-ibm-IBMADSM-MIB-ibmAdsmWarningMessage","0"},  

    {"SNMPTRAP-ibm-IBMADSM-MIB-ibmAdsmErrorMessage","0"},  

    {"SNMPTRAP-ibm-IBMADSM-MIB-ibmAdsmInfoMessage","0"},  

    {"SNMPTRAP-ibm-IBMADSM-MIB-ibmAdsmTrapMessage","0"},|  

    {"SNMPTRAP-ibm-BLADESPPALT-MIB-mmTrapTempC_criticalAlert","0"},  

    {"SNMPTRAP-ibm-BLADESPPALT-MIB-mmTrapTempC_major","0"},  

    {"SNMPTRAP-ibm-BLADESPPALT-MIB-mmTrapTempC_non-CriticalAlert","0"},  

}

```

add comma



Important: If you do not add the comma, the rules file contains a syntax error, and cannot be used.

15. Save the changes to the existing file, and exit the gedit utility.

16. Cancel any changes to the MIB manager version of the file, and exit the gedit utility.

17. Delete the MIB manager version of the file.

```
cd /workshop/unit03/*nckl*/ibm  
rm ibm-preclass.snmptrap.lookup
```



Note: You copy all the remaining files to the Netcool Knowledge Library. You do not want to overwrite the existing file with this version.

The master lookup file that MIB manager creates contains an extra include statement. The include statement is required if the files add a new vendor to the Netcool Knowledge Library. The include statement is not required if you add files to an existing vendor.

18. Modify the master lookup file as follows:

- a. Open the file for edit with the gedit utility.

```
gedit ibm.m2r.master.include.lookup
```

- b. Add a comment character to the preclass file as shown here:

```

#####
#include "$NCF_RULES_HOME/include-snmptrap/ibm/ibm-preclass.snmptrap.lookup"  
include "$NCF_RULES_HOME/include-snmptrap/ibm/ibm-IBMADSM-MIB.sev.snmptrap.lookup"

```



Note: You can remove the line or comment out the line.

- c. Save the changes, and exit the gedit utility.

19. Copy the MIB manager files to the Netcool Knowledge Library.

```
cp * $NC_RULES_HOME/include-snmptrap/ibm
```

```
cp: cannot create regular file
`/opt/IBM/tivoli/netcool/rules/include-snmptrap/ibm/ibm-preclass.include.snmptrap.rules': Permission denied
```

 **Note:** Ignore the permission denied message. The file in the message exists in the Netcool Knowledge Library, and the file permission does not allow a write operation. The contents of both files are the same, so the MIB manager version is not required.

The last step is to add include statements to reference the MIB manager rules file, and lookup file.

20. Add the include statement for the lookup file as follows:

- Change to the Netcool Knowledge library ibm directory.

```
cd $NC_RULES_HOME/include-snmptrap/ibm
```

- Open the IBM master lookup file for edit with the gedit utility.

```
chmod +w ibm.master.include.lookup
gedit ibm.master.include.lookup
```

- Add the include statement at the beginning of the file as shown here:

```
# Added the following line to include rules from MIB Manager
include "$NC_RULES_HOME/include-snmptrap/ibm/ibm.m2r.master.include.lookup"
#
#####
#
# Added the following line to include rules from MIB Manager
include "$NC_RULES_HOME/include-snmptrap/ibm/ibm.m2r.master.include.lookup"
#
include "$NC_RULES_HOME/include-snmptrap/ibm/ibm-preclass.snmptrap.lookup"
```

 **Note:** The include statement can go anywhere in the file. If you place the statement at the beginning of the file, it is easy to locate in the future.

- Save the changes and exit the gedit utility.

21. Add the include statement for the rules file as follows:

- Open the IBM master rules file for edit with the gedit utility.

```
chmod +w ibm.master.include.rules
gedit ibm.master.include.rules
```

- b. Add the include statement at the beginning of the file as shown here:

```
# Added the following line to include rules from MIB Manager
include "$NC_RULES_HOME/include-snmptrap/ibm/ibm.m2r.master.include.rules"
#
#####
#
# Added the following line to include rules from MIB Manager
include "$NC_RULES_HOME/include-snmptrap/ibm/ibm.m2r.master.include.rules"

include "$NC_RULES_HOME/include-snmptrap/ibm/ibm-METRICALARTRAP-MIB.include.snmpt
include "$NC_RULES_HOME/include-snmptrap/ibm/ibm-IBM-SYSTEM-RAID-MIB.include.snmpt
```



Note: The include statement can go anywhere in the file. If you place the statement at the beginning of the file, it is easy to locate in the future.

- c. Save the changes and exit the gedit utility.

22. Verify the syntax of the modified master rules file.

```
cd $NC_RULES_HOME
nco_p_syntax -server NYC_AGG_P -rulesfile snmptrap.rules
2014-12-18T20:02:44: debug: D-UNK-000-000: Auto-resizing lookup table 'syslogwor
rScore' with 10271 entries from 127 to 513
2014-12-18T20:02:44: Information: I-UNK-000-000: Rules file syntax OK
2014-12-18T20:02:44: Information: I-UNK-000-000: Disconnecting ...
2014-12-18T20:02:44: Debug: D-UNK-000-000: Shutting down Probewatch heartbeat th
read.
2014-12-18T20:02:45: Debug: D-UNK-000-000: Probewatch heartbeat thread detected
stop request
2014-12-18T20:02:45: Debug: D-ETC-004-051: THREAD MGR: thread probewatchheartbea
```

If the scan finds syntax errors, review the modifications that you made to the following files:

- ibm.m2r.master.include.lookup
- ibm.master.include.lookup
- ibm.master.include.rules
- ibm-preclass.snmptrap.lookup

Typical mistakes are usually spelling, and missing special characters.

Validating the rules

The true test of the rules file changes is the ability to correctly interpret an SNMP trap. You can use the MIB manager utility to generate a trap message.

The classroom image includes an instance of the SNMP probe. The probe is configured to use the Netcool Knowledge library rules. Process activity is configured to manage the probe. You can use process activity to stop, and start the probe.

- Find the process name for the probe.

```
nco_pa_status -server HOST1_PA -user netcool -password object00
```

Service Name	Process Name	Hostname	User	Status	PID
Core	MasterObjectServer	host1.tivoli.edunetcool		RUNNING	2067
	SyslogProbe	host1.tivoli.edunetcool		RUNNING	2068
	SnmpProbe	host1.tivoli.edunetcool		RUNNING	2069

- Stop the process.

```
nco_pa_stop -server HOST1_PA -user netcool -password object00 -process SnmpProbe
```

- Start the process.

```
nco_pa_start -server HOST1_PA -user netcool -password object00 -process SnmpProbe
```

- Verify whether the probe is running.

```
nco_pa_status -server HOST1_PA -user netcool -password object00
```

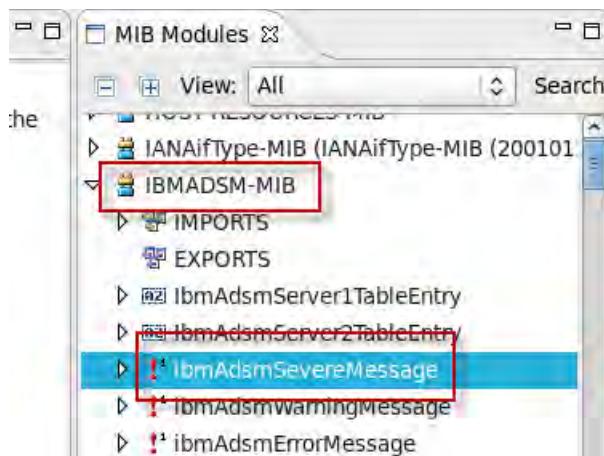
Service Name	Process Name	Hostname	User	Status	PID
Core	MasterObjectServer	host1.tivoli.edunetcool		RUNNING	2067
	SyslogProbe	host1.tivoli.edunetcool		RUNNING	2068
	SnmpProbe	host1.tivoli.edunetcool		RUNNING	25731

- Generate a test trap as follows:

- Start the MIB manager utility.

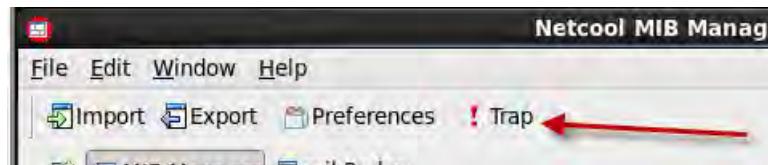
```
nco_mibmanager &
```

- Expand **IBMADSM-MIB**, and then click **ibmAdsmSevereMessage**.



- Click the **ibmAdsmSevereMessage** entry in the OID View frame.

d. Click Trap.



e. Enter some text in the ibmAdsmMessageText box, and click Execute.



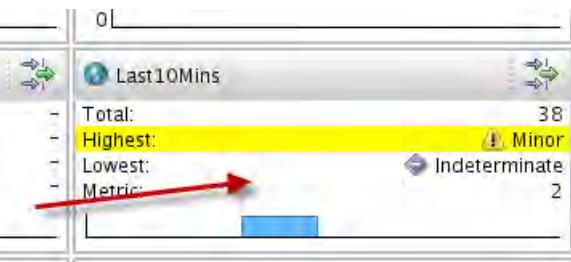
f. Click Cancel to close the box.



6. Open a Firefox browser.

7. Log in to Dashboard Application Services Hub as **ncoadmin** with password **object00**.8. Click the icon, select **Event Dashboard**.

9. Click the box that is labeled **Last10Mins**.



The Active Event List opens.

10. Locate the event.

The screenshot shows a table titled 'Last10Mins@OMNIBUS - Active Event List (host2.tivoli.edu:16311)'. The table has columns: Sev, Ack, Node, Alert Group, and Summary. One row is selected and highlighted with a red box. A red arrow points from the text in step 10 to the 'Node' column of the selected row.

Sev	Ack	Node	Alert Group	Summary
!	No	127.0.0.1	ibmAdsmSevere...	ibmAdsmSevereMessage: IBM ADSM Severe Message.
!	No	host1.tivoli.edu	Connectionstatus	PROBE: syslog connected from host host1.tivoli.edu (ID: 2).
!			DBStatus	Journal count (alerts.journal): 8
!	No	host2.tivoli.edu	ConnectionStatus	PROBE: mtrapd connected from host host2.tivoli.edu (ID: 1).

The event has a severity of **Major**. You modified the severity setting for this event in MIB manager before generating the rules.



Important: When you use the Netcool Knowledge Library with the SNMP probe, if the probe receives a trap, and there is no corresponding rule, the probe creates an event with a Summary of **Unknown Enterprise OID**.

If time permits, feel free to generate traps for the remaining messages.

11. Close the Active Event List window.
12. Log out of Dashboard Application Services Hub.
13. Exit the MIB manager utility.

Exercise 2 Probe high availability

In this exercise, you implement probe high availability by configuring the peer-to-peer mode of operation. You use the Simnet probe for this exercise.



Important: You perform part of following exercise on **host1**, and part on **host2**.

1. Select the **host1** image.
2. Open a Terminal window if necessary.
3. Modify the Simnet definition file as follows:



Note: The class image includes a custom definition file. The definition file contains a list of devices, and a type of error. The probe uses the file to create the artificial events. The file is not required for peer-to-peer mode.

- a. Change to the probe directory.

```
cd $OMNIHOME/probes/linux2x86
```

- b. Replace the existing file with the modified file.

```
cp /workshop/unit03/simnet.def .
```

4. Configure the Simnet probe for peer-to-peer mode as follows:

- a. Change to the probe directory.

```
cd $OMNIHOME/probes/linux2x86
```

- b. Save a copy of the property file before modifications.

```
cp simnet.props simnet.props.orig
```

- c. Open the property file for edit with the gedit utility.

```
gedit simnet.props
```

- d. Add the following lines to the end of the file:

Mode	:	'master'
PeerHost	:	'host2.tivoli.edu'
Peerport	:	9999

Mode	:	'master'
PeerHost	:	'host2.tivoli.edu'
Peerport	:	9999

- e. Save the changes, and exit the gedit utility.

5. Examine the events.

- a. Open a Firefox browser.

- b. Log in as **ncoadmin** with password **object00**.

- c. Open the **Event Dashboard**.

- d. Select **Last10Mins**.
- e. Click the icon to select all events, right-click, and select **Delete**.



All events for the last 10 minutes are removed.

Leave the Active Event List window open.

6. Start the probe.
`nco_p_simnet -server NYC_AGG_P &`
7. Return to the Active Event List window.
8. Scroll the view to the right, and locate the **Manager** column.

Sev	Ack	nt	Type	ExpireTime	Agent	Manager
!	No		Problem	Not Set	LinkMon	Simnet Probe
!	No		Type Not Set	Not Set	MachineMon	Simnet Probe
!	No		Type Not Set	Not Set	MachineMon	Simnet Probe
!	No		Type Not Set	Not Set	MachineMon	Simnet Probe
!	No		Problem	Not Set	LinkMon	Simnet Probe
!	No		Problem	Not Set	LinkMon	Simnet Probe

The event list contains events from the Simnet Probe.

Leave the event list open.

9. Switch to the **host2** image.

To facilitate the verification of high availability, you change two Simnet files.



Note: The changes are not required for peer-to-peer mode. The changes merely make the events from this probe unique in the event list.

10. Modify the Simnet rules as follows:

- Change to the probe directory.

```
cd $OMNIHOME/probes/linux2x86
```

- Save a copy of the rules file before modifications.

```
cp simnet.rules simnet.rules.orig
```

- Open the property file for edit with the gedit utility.

```
gedit simnet.rules
```

- Locate the following line:

```
}
else
{
    @Manager      = "Simnet Probe"
    @Class        = 3300
    @Node         = $Node
    @Agent        = $Agent
    @AlertGroup   = $Group
```

- Change the value for Manager to **Simnet2 Probe**.

```
}
else
{
    @Manager      = "Simnet2 Probe"
    @Class        = 3300
```

- Save the changes, and exit the gedit utility.

11. Modify the Simnet definition file as follows:



Note: The class image includes a custom definition file. The definition file contains a list of devices, and a type of error. The probe uses the file to create the artificial events.

- Change to the probe directory.

```
cd $OMNIHOME/probes/linux2x86
```

- Replace the existing file with the modified file.

```
cp /workshop/unit03/simnet.def .
```

12. Configure the Simnet probe for peer-to-peer mode as follows:

- Change to the probe directory.

```
cd $OMNIHOME/probes/linux2x86
```

- Save a copy of the property file before modifications.

```
cp simnet.props simnet.props.orig
```

- Open the property file for edit with the gedit utility.

```
gedit simnet.props
```

- d. Add the following lines to the end of the file:

```
Mode : 'slave'
PeerHost : 'host1.tivoli.edu'
Peerport : 9999
```

```
#####
Mode : 'slave'
PeerHost : 'host1.tivoli.edu' host1.tivoli.edu'
Peerport : 9999
```

- e. Save the changes, and exit the gedit utility.

13. Start the probe.

```
nco_p_simnet -server NYC_AGG_P &
```

14. Return to the **host1** image.

15. Return to the Active Event List window.

Last10Mins@OMNIBUS - Active Event List (host2.tivoli.edu:16311) -						
Sev	Ack	nt	Type	ExpireTime	Agent	Manager
!	No		Type Not Set	Not Set	MachineMon	Simnet Probe
!	No		Type Not Set	Not Set	MachineMon	Simnet Probe
!	No		Type Not Set	Not Set	MachineMon	Simnet Probe
!	No		Problem	Not Set	LinkMon	Simnet Probe
!	No		Type Not Set	Not Set	MachineMon	Simnet Probe
!	No		Problem	Not Set	LinkMon	Simnet Probe
!	No		Problem	Not Set	LinkMon	Simnet Probe

The event list does not contain any events with Manager of Simnet2 Probe.

16. Stop the Simnet probe on **host1**.

```
pkill nco_p_simnet
```

17. Return to the Active Event List window.

18. Repeat the steps to remove all events.

Last10Mins@OMNIBUS - Active Event List (host2.tivoli.edu:16311) -						
Sev	Ack	Count	Type	ExpireTime	Agent	Manager
!	No	1	Type Not Set	Not Set	MachineMon	Simnet2 Probe
!	No	3	Type Not Set	Not Set	MachineMon	Simnet2 Probe
!	No	37	Problem	Not Set	LinkMon	Simnet Probe
!	No	1	Information	330	OMNIBus SelfM...	OMNIBus Self ...
!	No	2	Type Not Set	Not Set	MachineMon	Simnet2 Probe

After a short time, events appear with Manager of **Simnet2 Probe**.

19. Switch to the **host2** image.

20. Examine the end of the Simnet log file.

```
tail $OMNIHOME/log/simnet.log
2014-12-18T22:11:11: Warning: W-UNK-000-000: [P2P] : All events will now be
forwarded
2014-12-18T22:11:12: Warning: W-UNK-000-000: [P2P] : All events will now be
forwarded
2014-12-18T22:11:13: Warning: W-UNK-000-000: [P2P] : All events will now be
forwarded
2014-12-18T22:11:14: Warning: W-UNK-000-000: [P2P] : All events will now be
forwarded
2014-12-18T22:11:15: Warning: W-UNK-000-000: [P2P] : All events will now be
forwarded
2014-12-18T22:11:16: Warning: W-UNK-000-000: [P2P] : All events will now be
forwarded
2014-12-18T22:11:17: Warning: W-UNK-000-000: [P2P] : All events will now be
forwarded
```

The messages indicate that the probe is now sending events to the ObjectServer.

21. Stop the Simnet probe on **host2**.

```
pkill nco_p_simnet
```

 **Note:** You do not stop the probe for any reason that is related to peer-to-peer. You stop the probe because it is no longer needed in this exercise, and you change the configuration in the next exercise.

22. Switch to the **host1** image.

23. Start the Simnet probe.

```
nco_p_simnet -server NYC_AGG_P &
```

24. Return to the Active Event List, and remove all events.

After a short time, events appear with Manager of Simnet Probe.

25. Close the Active Event List window.

26. Log out of Dashboard Application Services Hub.

27. Close the Firefox browser.

Exercise 3 Probe remote administration

In this exercise, you implement the optional feature for reloading multiple probe rules files. This facility is useful when you change the rules files for several probes, and you want to implement all the changes at the same time.



Important: You perform the part of following exercise on **host1**, and part on **host2**.

Modifying the ProbeWatch configuration

The optional reload feature includes a probe rules file. The rules file creates an ObjectServer event when the probe rereads its rules file. The event contains the status of the reread operation. When the event reaches the ObjectServer, a trigger locates the event. The trigger extracts the status from the event, and writes the status to a log file.

The host1 image contains probes, and the host2 image contains probes. You must make changes on both images.

1. Switch to the **host1** image.
2. Change to the location of the rules file.

```
cd $OMNIHOME/extensions/roi
```

The rules file uses a special command to generate the ProbeWatch event. The command must be modified before it is used.

3. Open the rules file for edit with the gedit utility.

```
chmod +w probewatch.include  
gedit probewatch.include
```

4. Locate the following line:

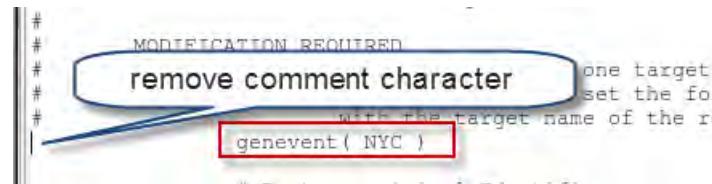
```
#  
#      MODIFICATION REQUIRED  
#      genevent requires at least one target to be set in the rules file  
#      once that has been set the following line needs to be updated  
#      with the target name of the required destination ObjectServer  
#      genevent( <DefaultOS> )
```



Hint: The line is near the end of the file.

The genevent command creates the ObjectServer event. The value in parentheses identifies the ObjectServer.

- Remove the comment character. Change the line as shown here.



Note: The value **NYC** refers to *some* ObjectServer. You define the actual ObjectServer later.

- Save the changes and exit the gedit utility.
- Modify the Simnet rules to incorporate the file as follows.
 - Change to the probe configuration directory.
cd \$OMNIHOME/probes/linux2x86
 - Save a copy of the rules file before modification.
cp simnet.rules simnet.rules.orig
 - Open the rules file for edit with the gedit utility.
gedit simnet.rules
 - Locate the ProbeWatch section, and remove all lines between the open, and close *curly braces*.

```
if( match( @Manager, "ProbeWatch" )
{
    switch(@Summary)
    {
        case "Running ...":
            @Severity = 1
            @AlertGroup = "probestat"
            @Type = 2
        case "Going Down ...":
            @Severity = 5
            @AlertGroup = "probestat"
            @Type = 1
        default:
            @Severity = 1
    }
    @AlertKey = @Agent
    @Summary = @Agent + " probe on " + @Node + ":" + @Summary
}
else
```

remove

Every probe rules file includes a ProbeWatch section. You replace the commands in this section, with commands to incorporate the revised probe watch file.

- e. Add the include statement as shown here.

```
include "/opt/IBM/tivoli/netcool/omnibus/extensions/roi/probewatch.include"
```

```
#####
if( match( @Manager, "ProbeWatch" ) )
{
    include "/opt/IBM/tivoli/netcool/omnibus/extensions/roi/probewatch.include"
}
else
```

- f. Add the registertarget command as shown here.

```
NYC = registertarget( %Server, "", "alerts.status" )
```

```
#####
#
NYC = registertarget( %Server, "", "alerts.status" )

if( match( @Manager, "ProbeWatch" ) )
{
    include "/opt/IBM/tivoli/netcool/omnibus/extensions/roi/probewatch.include"
}
else
```

The registertarget command defines the actual ObjectServer that receives events that the genevent command creates. The reference to **%Server** specifies that the command uses the ObjectServer that is specified on the probe Server property. When configured in this manner, two probes that share the same file can reference different ObjectServers. One probe can send events to ObjectServer A, and the other sends events to ObjectServer B.

The other option for the registertarget command is to hardcode an ObjectServer name.

- g. Save the changes, and exit the gedit utility.

8. Verify the syntax of the modified rules file as follows:

```
nco_p_syntax -server NYC_AGG_P -rulesfile simnet.rules
```

```
2014-12-19T15:23:58: Debug: D-ETC-004-050: THREAD MGR: thread probewatchheartbea
t-thread (0x24337b0) running
2014-12-19T15:23:58: Information: I-UNK-000-000: Rules file syntax OK
2014-12-19T15:23:58: Information: I-UNK-000-000: Disconnecting ...
2014-12-19T15:23:58: Debug: D-UNK-000-000: Shutting down Probewatch heartbeat th
read.
```

The class image includes a Syslog probe, and an SNMP probe. These probes use the Netcool Knowledge Library rules files.

9. Modify the Syslog rules to incorporate the file as follows.

- a. Change to the Netcool Knowledge Library directory.

```
cd $NC_RULES_HOME
```

- b. Save a copy of the rules file before modification.

```
cp syslog.rules syslog.rules.orig
```

- c. Open the rules file for edit with the gedit utility.

```
chmod +w syslog.rules  
gedit syslog.rules
```

- d. Locate the ProbeWatch section, and remove all lines between the open, and close *curly braces*.

```
if(match(@Manager, "ProbeWatch"))
{
    $ProbeName = @Agent
    $ProbeStatus = @Summary

    @Agent = "ProbeWatch"

    @Node = hostname()

    @AlertGroup = "Probe Status"
    @AlertKey = "Probe: " + $ProbeName + ", Host: " + hostname() + ", Objec
    @Summary = "Probe " + $ProbeStatus + " (" + @AlertKey + ")"
    switch($ProbeStatus)
    {
        case "Running ...":
            @Severity = 1
            @Type = 2
        case "Going Down ...":
            @Severity = 5
            @Type = 1
        default:
            @Severity = 2
            @Type = 1
    }
    @Identifier = @Node + " " + @AlertKey + " " + @AlertGroup + " " + @Type
    $ProbeStatus
}
else
```



Note: The ProbeWatch section is further down in the file, and contains more lines than the Simnet probe. Remove all of the lines.

- e. Add the include statement as shown here.

```
include "/opt/IBM/tivoli/netcool/omnibus/extensions/roi/probewatch.include"
```

```
if(match(@Manager, "ProbeWatch"))
{
    include "/opt/IBM/tivoli/netcool/omnibus/extensions/roi/probewatch.include"
}
else
{
    log(DEBUG, "<<<< Entering... syslog.rules >>>>")
```

You include the same file as the one for the Simnet probe.

- f. Scroll up in the rules file, and add the registertarget command as shown here.

```
NYC = registertarget( %Server, "", "alerts.status" )
```

```
# 1.0 - Initial Release.  
#  
#NYC = registertarget( iServer, "", "alerts.status" )  
  
table syslogSrcType = "$NC_RULES_HOME/syslog-SrcType.lookup"  
default = "Unknown"  
  
table syslogDestTargets = "$NC_RULES_HOME/include/syslog/Targets.ncx"   
destTargets =
```

The registertarget command must be the *first non-commented line* in the rules file. The Syslog rules file contains a number of table statements. The registertarget must appear before these statements.

- g. Save the changes, and exit the gedit utility.
10. Verify the syntax of the modified rules file as follows:

```
nco_p_syntax -server NYC_AGG_P -rulesfile syslog.rules
```

```
| Class' with 10280 entries from 127 to 514
| 2014-12-19T15:43:43: Debug: D-UNK-000-000: Auto-resizing lookup table 'sy
| rScore' with 10271 entries from 127 to 513
| 2014-12-19T15:43:43: Information: I-UNK-000-000 Rules file syntax OK
| 2014-12-19T15:43:43: Information: I-UNK-000-000 Disconnecting ...
| 2014-12-19T15:43:43: Debug: D-UNK-000-000: Shutting down Probewatch heart
| read.
```

11. Restart the Syslog probe to reread the modified rules file.

- a. Stop the probe.

```
nco_pa_stop -server HOST1_PA -user netcool -password object00 -process
SyslogProbe
```

- b. Start the probe.

```
nco_pa_start -server HOST1_PA -user netcool -password object00 -process
SyslogProbe
```

- c. Verify the status.

```
nco_pa_status -server HOST1_PA -user netcool -password object00
```

Service Name	Process Name	Hostname	User	Status	PID
Core	MasterObjectServer	host1.tivoli.edunetcool		RUNNING	2060
	SyslogProbe	host1.tivoli.edunetcool		RUNNING	17571
	SnmpProbe	host1.tivoli.edunetcool		RUNNING	17932

The probe is running with the modified rules file.

12. Repeat the previous steps to apply the same changes to snmptrap.rules.

- a. Save a copy of the rules file before modification.

```
cp snmptrap.rules snmptrap.rules.orig
```

- b. Open the rules file for edit with the gedit utility.

```
chmod +w snmptrap.rules
gedit snmptrap.rules
```

- c. Locate the ProbeWatch section, and remove all lines between the open, and close *curly braces*.

3 Probes exercises

Exercise 3 Probe remote administration

- Add the include statement as shown here.

```
include "/opt/IBM/tivoli/netcool/omnibus/extensions/roi/probewatch.include"

if(match(@Manager, "ProbeWatch"))
{
    include "/opt/IBM/tivoli/netcool/omnibus/extensions/roi/probewatch.include"
}
else
{
    log(DEBUG, "<<<< Entering... snmptrap.rules >>>>")
}
```

- Scroll up in the rules file, and add the registertarget command as shown here.

```
NYC = registertarget( %Server, "", "alerts.status" )
```

```
#####
NYC = registertarget( %Server, "", "alerts.status" )
#####
# Register the Object Server name as DefaultOS. Uncomment the BackupOS line
# and update the backup Server name if you wish to divert events
# to backup your Object Server.
DefaultOS = registertarget( %Server, %ServerBackup, "alerts.status")
```

- Save the changes, and exit the gedit utility.

13. Verify the syntax of the modified rules file as follows:

```
nco_p_syntax -server NYC_AGG_P -rulesfile snmptrap.rules
```

```
Class' with 10280 entries from 127 to 514
2014-12-19T15:58:42: Debug: D-UNK-000-000: Auto-resizing lookup table 'sys
rScore' with 10271 entries from 127 to 513
2014-12-19T15:58:42: Information: I-UNK-000-000: Rules file syntax OK
2014-12-19T15:58:42: Information: I-UNK-000-000: Disconnecting ...
2014-12-19T15:58:42: Debug: D-UNK-000-000: Shutting down Probewatch heartb
read.
```

14. Restart the SNMP probe to reread the modified rules file.

- Stop the probe.

```
nco_pa_stop -server HOST1_PA -user netcool -password object00 -process
SnmpProbe
```

- Start the probe.

```
nco_pa_start -server HOST1_PA -user netcool -password object00 -process
SnmpProbe
```

- Verify the status.

```
nco_pa_status -server HOST1_PA -user netcool -password object00
```

Service Name	Process Name	Hostname	User	Status	PID
Core	MasterObjectServer	host1.tivoli.edunetcool		RUNNING	2060
SyslogProbe		host1.tivoli.edunetcool		RUNNING	17571
SnmpProbe		host1.tivoli.edunetcool		RUNNING	2062

The probe is running with the modified rules file.

15. Switch to the **host2** image.

16. Repeat the steps to modify the genevent command in the probewatch.include file.

17. Repeat the steps to configure the Simnet probe on host2 to use the probe watch file.



Note: Do not start the probe on host2.

Configuring process activity

When you use the *rules reload* feature, process activity must manage all probes. You must add the Simnet probes to process activity.

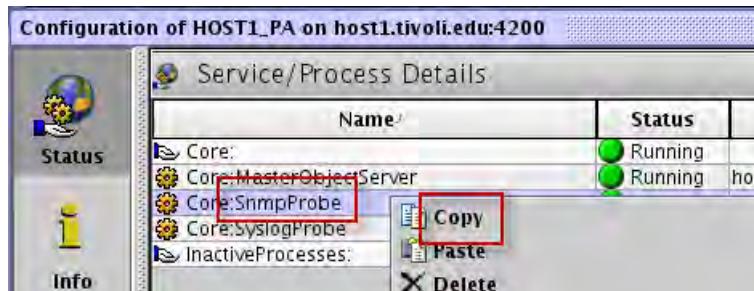
1. Switch to the **host1** image.
2. Stop the Simnet probe that is running.

```
pkkill nco_p_simnet
```

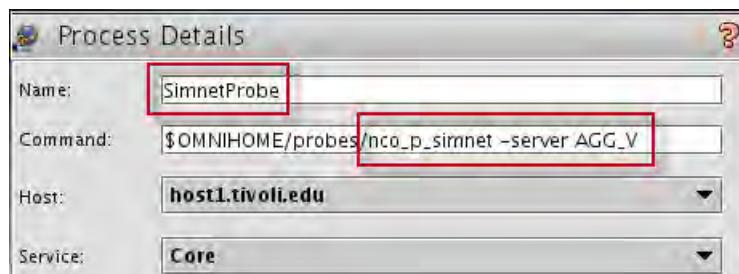
3. Start the Netcool/OMNIbus Administrator utility.

```
nco_config &
```

4. Connect to the **HOST1_PA** process agent as **netcool** with password **object00**.
5. Right-click **SnmpProbe**, and select **Copy**.

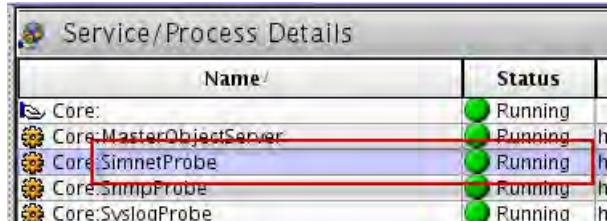


6. Right-click, and select **Paste**.
7. Enter **SimnetProbe** for the name. Change the command to **nco_p_simnet**. Add **-server AGG_V**. Click **OK**.



Note: You send events to the *virtual* ObjectServer.

8. Right-click **SimnetProbe**, and select **Start**.

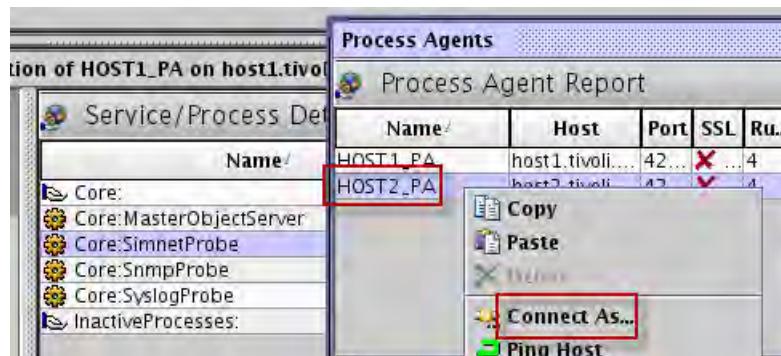


Process activity now controls the Simnet probe.

You must configure the process agent on host2 to manage the local Simnet probe.

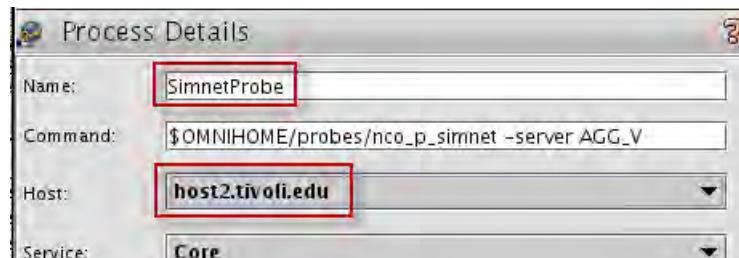
9. Right-click **SimnetProbe**, and select **Copy**.

10. Locate the process agent box, and connect to HOST2_PA as **netcool** with user **object00**.

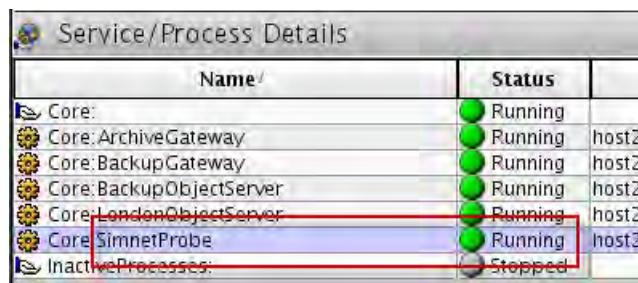


11. Right-click anywhere in the list of processes, and select **Paste**.

12. Enter **SimnetProbe** for the name. Select **host2.tivoli.edu** for the host. Click **OK**.

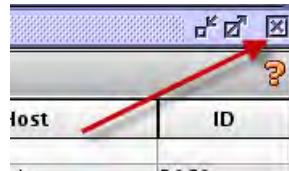


13. Right-click **SimnetProbe**, and select **Start**.



With a few mouse clicks, you copy an entry from one process agent, and paste the entry into another process agent. You make a few changes, and start the process.

14. Click the **X** to close the **HOST2_PA** process agent box.



15. Click **Yes** to confirm the close.

16. Click **Yes** to save the configuration.

17. Click **OK** to close the confirmation window.

The configuration changes are written to \$OMNIHOME/etc/nco_pa.conf on **host2**. This process is an example of the ability to remotely control probes.

18. Repeat the previous steps to close the **HOST1_PA** process agent box, and save the changes.

Leave the Netcool/OMNIBus Administrator utility open.

Installing the rules reload feature

The **reload** feature is implemented as a collection of ObjectServer triggers, and procedures. Netcool/OMNIBus includes a file of SQL commands that create these objects.

1. Change to the location of the SQL file.

```
cd $OMNIHOME/extensions/roi
```

2. Import the file into **NYC_AGG_P**.

```
nco_sql -server NYC_AGG_P -user root -password object00 < probemanagement.sql
(0 rows affected)
```

3. Return to the Netcool/OMNIBus Administrator.

4. Connect to the **NYC_AGG_P** ObjectServer as **root** with password **object00**.

5. Examine the list of trigger groups.

Trigger Groups	
Name	Enabled
AdvCorr	✓ true
audit_config	✗ false
automatic_backup_system	✓ true
compatibility_triggers	✓ true
connection_watch	✓ true
default_triggers	✓ true
fallback_triggers	✓ true
gateway_triggers	✓ true
iduc_triggers	✓ true
oslc	✓ true
primary_only	✓ true
probe_management	✓ true
proxier_triggers	✓ true

The file created the **probe_management** trigger group.

6. Examine the list of triggers. Click **Group** to sort the entries by group name. Scroll down and locate the entries for the **probe_management** group.

Name	Group	Kind	Priority	De
hash_not_ack	primary_only	Temporal	1	
generic_clear	primary_only	Temporal	1	
mail_on_critical	primary_only	Temporal	1	
resync_complete	primary_only	Signal	1	
probeevent_insert	probe_management	Database	2	
probeevent_reinsert	probe_management	Database	2	
profiler_group_report	profiler_triggers	Signal	2	
profiler_report	profiler_triggers	Signal	1	

The file created the `probeevent_insert`, and `probeevent_reinsert` database triggers.

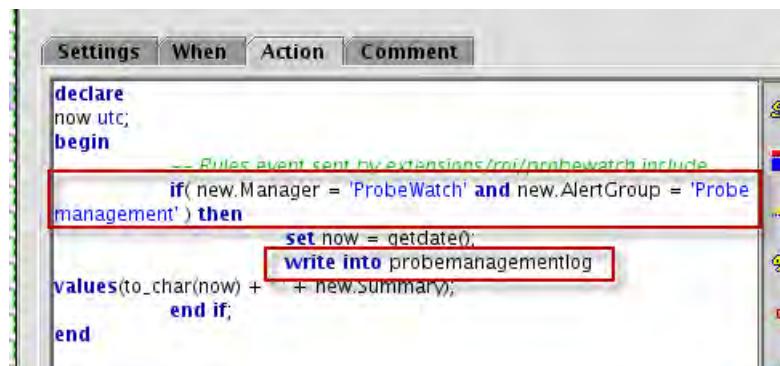
7. Right-click `probeevent_insert`, and select **Edit Trigger**.

Database Trigger Details

Name:	probeevent_insert
Group:	probe_management
<input type="button" value="Settings"/> <input type="button" value="When"/> <input type="button" value="Action"/> <input type="button" value="Comment"/>	
On	<input type="button" value="alerts"/> <input type="button" value="status"/>
Run	Priority: <input type="button" value="2"/> <div style="display: flex; justify-content: space-around;"> <input type="radio"/> Delete <input checked="" type="radio"/> Insert <input type="radio"/> Reinsert <input type="radio"/> Update </div>
<input type="radio"/> Pre database action <input checked="" type="radio"/> Post database action	

The trigger activates based on an **INSERT** into the `alerts.status` table.

8. Click Action.



The screenshot shows a software interface for configuring a trigger. At the top, there are tabs labeled 'Settings', 'When', 'Action', and 'Comment'. The 'Action' tab is selected. Below the tabs, there is a code editor containing the following PL/SQL-like script:

```
declare
now utc;
begin
-- Rule.event sent by extensions/moi/probeawatch include
if( new.Manager = 'ProbeWatch' and new.AlertGroup = 'Probe
management' ) then
set now = getdate();
write into probemanagementlog
values(to_char(now) + ' ' + new.summary);
end if;
end
```

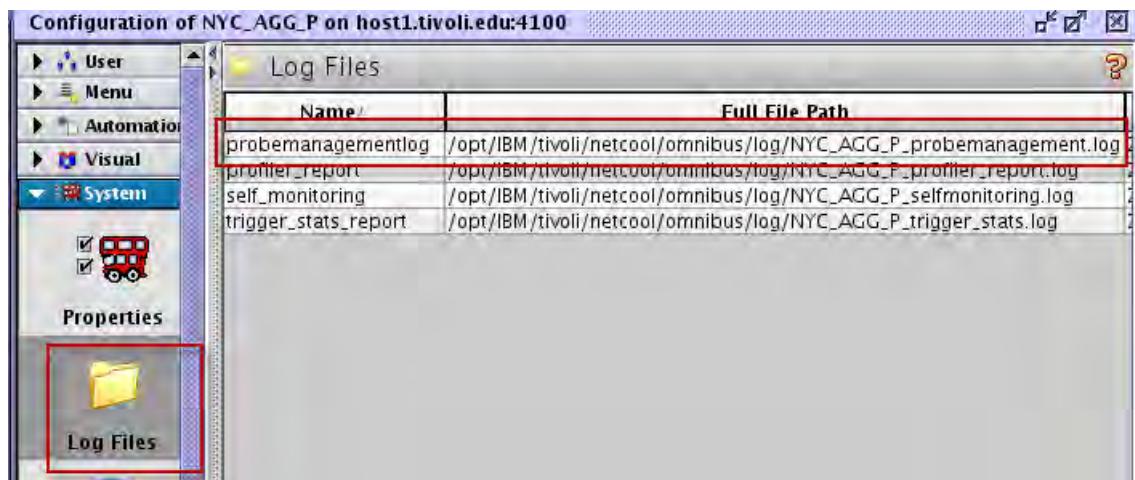
Two specific lines of code are highlighted with red boxes: 'if(new.Manager = 'ProbeWatch' and new.AlertGroup = 'Probe management') then' and 'write into probemanagementlog'. These highlights indicate the logic that triggers the log entry.

The trigger activates when the ObjectServer receives a new event. The trigger checks the event to determine whether it comes from the *ProbeWatch* feature of a probe. If it does, the trigger writes information to a log file.

The probeevent_reinsert procedure runs the same action commands. The probeevent_reinsert trigger activates based on an event deduplication.

9. Click Cancel.

10. Examine the list of log files.

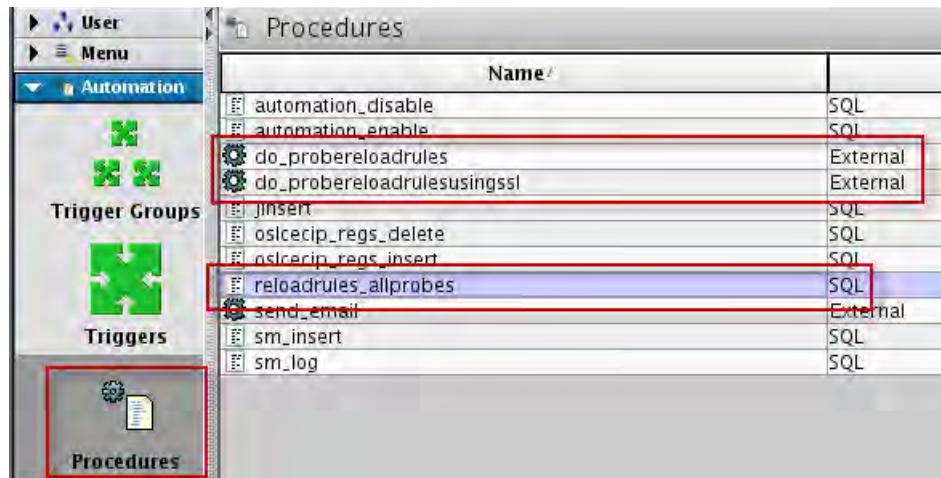


The screenshot shows the 'Log Files' section of the configuration interface for the 'NYC_AGG_P' probe. On the left, there is a navigation tree with 'User', 'Menu', 'Automation', 'Visual', and 'System' sections. Under 'System', there is a 'Properties' section with a 'Log Files' icon, which is highlighted with a red box. The main pane displays a table titled 'Log Files' with two columns: 'Name' and 'Full File Path'. The table contains the following data:

Name	Full File Path
probemanagementlog	/opt/IBM/tivoli/netcool/omnibus/log/NYC_AGG_P_probemanagement.log
profiler_report	/opt/IBM/tivoli/netcool/omnibus/log/NYC_AGG_P_profiler_report.log
self_monitoring	/opt/IBM/tivoli/netcool/omnibus/log/NYC_AGG_P_selfmonitoring.log
trigger_stats_report	/opt/IBM/tivoli/netcool/omnibus/log/NYC_AGG_P_trigger_stats.log

The entry that is shown here defines the name and location of the log file.

11. Examine the list of procedures.



The file creates one SQL procedure and two external procedures. One external procedure communicates with the probe over HTTP. The other external procedure uses HTTPS.

12. Right-click **reloadrules_allprobes**, and select **Edit Procedure**.

```

Actions:
begin
    set now = getdate();
    write into probemanagementlog values(to_char(now) + ' ' + 'Reload all probe
rules request');

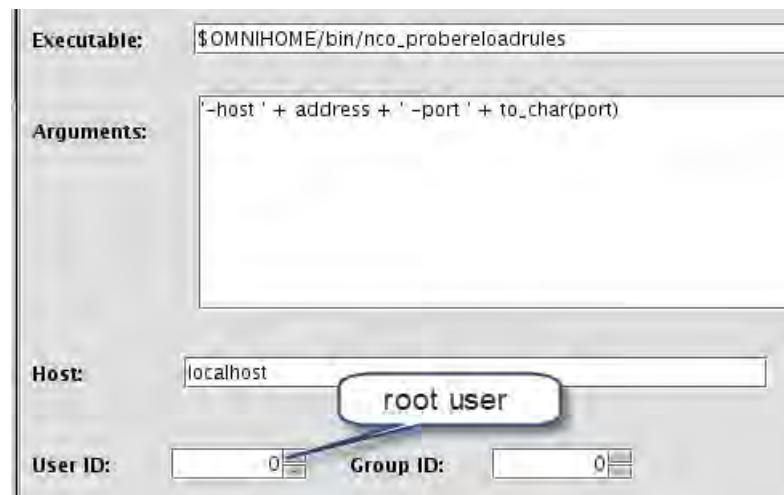
    -- refresh_all_registered_probes
    for each row probe in registry.probes
        begin
            -- A status of 1 means the probe is running
            if( probe.Status = 1 )
            then
                if( probe.HTTPS_port != 0 )
                then
                    execute do_probereloadrulesusingssl(
                        probe.Hostname, probe.HTTPS_port );
                set now = getdate();
                write into probemanagementlog values(
                    to_char(now) + ' Sent HTTPS reload rule request to ' + probe.Name + ' on ' +
)

```

You run the **reloadrules_allprobes** procedure when you want to reload the rules on all probes. When the procedure runs, it creates an entry in the probe management log file. It reads the entries in the probe registry ObjectServer table. It checks whether each entry contains an HTTP port value or HTTPS port value. An entry with a port value indicates that the probe supports remote access. If the probe supports HTTPS, the trigger runs the **do_probereloadrulesusingssl** external procedure. If the probe supports HTTP, the trigger runs the **do_probereloadrules** external procedure.

13. Click Cancel.

14. Right-click **do_probereloadrules**, and select **Edit Procedure**.



The procedure sends a request to the process activity agent. The process activity agent runs \$OMNIHOME/bin/nco_probereloadrules with the specified parameter list. The process agent runs the command on **localhost** as the **root** user. The command sends a reload request to a probe over HTTP.

15. Change the User ID value to **500** and change the Group ID value to **501**. Click **OK** to modify the procedure.



The process agent now runs the command as the **netcool** user.



Important: The process agent on the class image is running as the **netcool** user. If the process agent runs as a non-root user, the agent cannot run a command as the **root** user.

16. Repeat the previous steps and change the User ID and Group ID values for the other procedure.

17. Import the file into **NYC_AGG_B**.

```
nco_sql -server NYC_AGG_B -user root -password object00 < probemanagement.sql
(0 rows affected)
```

18. Return to the Netcool/OMNIbus Administrator utility and connect to **NYC_AGG_B**.

19. Repeat the previous steps, and modify the User ID and Group ID values for the two procedures.
20. Close the Netcool/OMNIbus Administrator utility.

Using the feature

To reload the rules for all probes, you connect to the ObjectServer, and run the procedure.

1. Run the procedure.

```
nco_sql -server NYC_AGG_P -user root -password object00
1> execute reloadrules_allprobes
2> go
(0 rows affected)
1> quit
```

2. Examine the probe management log file.

```
cd $OMNIHOME/log
more NYC_AGG_P_probemanagement.log1
```

```
-----
Fri Dec 19 16:50:55 2014 Reload all probe rules request
Fri Dec 19 16:50:55 2014 Sent HTTP reload rules request to mttrapd on host1.tivoli.edu:4198
Fri Dec 19 16:50:55 2014 [HTTP interface not active for simnet on host1.tivoli.edu]
Fri Dec 19 16:50:55 2014 syntax on host2.tivoli.edu is not running
Fri Dec 19 16:50:55 2014 syntax on host1.tivoli.edu is not running
Fri Dec 19 16:50:55 2014 [HTTP interface not active for simnet on host2.tivoli.edu]
Fri Dec 19 16:50:55 2014 Sent http reload rules request to syslog on host1.tivoli.edu:4199
...
Fri Dec 19 16:50:55 2014 Reload all rules summary: managed probes 2, unmanaged probes 2,
probes not running 2
Fri Dec 19 16:51:46 2014 mttrapd probe on host1.tivoli.edu:4198 : Rules file reread upon
HTTP request successful ...
Fri Dec 19 16:51:57 2014 syslog probe on host1.tivoli.edu:4199 : Rules file reread upon
HTTP request successful ...
```

Some of the messages in the log are created by the `execute reloadrules_allprobes` procedure. The messages from the procedure appear first. The last two messages come from the probes. The two Simnet probes are not configured for remote access. The Snmp and Syslog probes are configured for remote access. Each of these probes successfully reread their rules file.

3. Configure the Simnet probe on **host1** for remote access as follows:

- a. Change to the probe configuration directory.

```
cd $OMNIHOME/probes/linux2x86
```

- b. Save a copy of the property file before modification.

```
cp simnet.props simnet.props.orig
```

- c. Open the property for edit with the gedit utility.

```
gedit simnet.props
```

- d. Add the lines that are shown here to the end of the file.

```
NHttpd.EnableHTTP      :      TRUE
NHttpd.ListeningHostname   :      "host1.tivoli.edu"
NHttpd.ListeningPort       :      4190
```

```
#####
Mode          : 'master'
PeerHost      : 'host2.tivoli.edu'
Peerport      : 9999
NHttpd.EnableHTTP      :      TRUE
NHttpd.ListeningHostname   :      "host1.tivoli.edu"
NHttpd.ListeningPort       :      4190
```

- e. Save the changes, and exit the gedit utility.

4. Restart the probe for the changes to take effect.

- a. Stop the probe.

```
nco_pa_stop -server HOST1_PA -user netcool -password object00 -process
SimnetProbe
```

- b. Start the probe.

```
nco_pa_start -server HOST1_PA -user netcool -password object00 -process
SimnetProbe
```

- c. Verify the status.

```
nco_pa_status -server HOST1_PA -user netcool -password object00
```

Service Name	Process Name	Hostname	User	Status	PID
Core	MasterObjectServer	host1.tivoli.edunetcool		RUNNING	2060
	SyslogProbe	host1.tivoli.edunetcool		RUNNING	17571
	SnmpProbe	host1.tivoli.edunetcool		RUNNING	17932
	SimnetProbe	host1.tivoli.edunetcool		RUNNING	25894

The probe is now configured to support remote access.

In a subsequent unit, you create an event view that uses several custom event column names. You added the column names to the ObjectServer in a previous unit. The columns currently do not contain any values. Configure the Simnet probe to use a lookup table to populate the columns.



Note: The lookup table is not a requirement for remote administration. However, when you configure it now, it provides a convenient technique to verify remote administration.

5. Configure the Simnet probe to use the lookup table as follows:

- a. Save a copy of the rules file before modification.

```
cp simnet.rules simnet.rules.orig
```

- b. Open the property for edit with the gedit utility.

```
gedit simnet.rules
```

- c. Add the lines that are shown here to the beginning of the file.

```
table nyc = "/workshop/unit03/simnet.lookup"
default =
{ "UNKNOWN", "UNKNOWN", "UNKNOWN", "UNKNOWN", "UNKNOWN", "UNKNOWN", "UNKNOWN", "UNKNOWN" }

#####
#NYC = registertarget( %Server, "", "alerts.status" )
table nyc = "/workshop/unit03/simnet.lookup"
default = { "UNKNOWN", "UNKNOWN", "UNKNOWN", "UNKNOWN", "UNKNOWN", "UNKNOWN", "UNKNOWN" }
if( match( @Manager, "ProbeWatch" ) )
{

```

Table definitions follow any registertarget definitions in a rules file. The *default* statement provides values for each column name if the table does not contain a matching value.

- d. Scroll down in the file, and add the following lines:

```
[@AgencyId,@SiteId,@SiteName,@SiteAddr,@SiteCity,@SiteState,@SiteCountry]
= lookup(@Node, nyc)
update(@AgencyId)
Update(@SiteId)

@Severity      = $Severity
@IdIdentifier = $Node + $Agent + $Severity + $Group

[ @AgencyId,@SiteId,@SiteName,@SiteAddr,@SiteCity,@SiteState,@SiteCountry] = lookup(@Node, nyc)
update(@AgencyId)
update(@SiteId)

if (nmatch($Summary, "Port failure"))
{
```

The lookup statement searches the **nyc** table for an entry with the same value as the Node column. If an entry is found, the seven column names in the list are populated with values from the table. If no entry is found, the seven column names are populated with the values from the *default* statement.

- e. Save the changes, and exit the gedit utility.

6. Verify the syntax of the modified rules file.

```
nco_p_syntax -server NYC_AGG_P -rulesfile simnet.rules
```

Make sure that no syntax errors exist.

7. Switch to the **host2** image.
8. Configure the Simnet probe on **host2** for remote access as follows:

- a. Change to the probe configuration directory.

```
cd $OMNIHOME/probes/linux2x86
```

- b. Save a copy of the property file before modification.

```
cp simnet.props simnet.props.orig
```

- c. Open the property for edit with the gedit utility.

```
gedit simnet.props
```

- d. Added the lines that are shown here to the end of the file.

```
NHttpd.EnableHTTP      :      TRUE
NHttpd.ListeningHostname   :      "host2.tivoli.edu"
NHttpd.ListeningPort     :      4190
```

```
#####
Mode          : 'slave'
PeerHost      : 'host1.tivoli.edu'
Peerport      : 9999
#####
NHttpd.EnableHTTP      :      TRUE
NHttpd.ListeningHostname   :      "host2.tivoli.edu"
NHttpd.ListeningPort     :      4190|
```

- e. Save the changes, and exit the gedit utility.

9. Restart the probe for the changes to take effect.

- a. Stop the probe.

```
nco_pa_stop -server HOST2_PA -user netcool -password object00 -process
SimnetProbe
```

- b. Start the probe.

```
nco_pa_start -server HOST2_PA -user netcool -password object00 -process
SimnetProbe
```

- c. Verify the status.

```
nco_pa_status -server HOST2_PA -user netcool -password object00
```

Service Name	Process Name	Hostname	User	Status	PI
Core	BackupObjectServer	host2.tivoli.edunetcool		RUNNING	
	BackupGateway	host2.tivoli.edunetcool		RUNNING	
	ArchiveGateway	host2.tivoli.edunetcool		RUNNING	
	LondonObjectServer	host2.tivoli.edunetcool		RUNNING	
	SimnetProbe	host2.tivoli.edunetcool		RUNNING	

The probe is now configured to support remote access.

10. Configure the Simnet probe to use the lookup table as follows:

- a. Save a copy of the rules file before modification.

```
cp simnet.rules simnet.rules.orig
```

3 Probes exercises

Exercise 3 Probe remote administration

- b. Open the property for edit with the gedit utility.

```
gedit simnet.rules
```

- c. Add the lines that are shown here to the beginning of the file.

```
table nyc = "/workshop/unit03/simnet.lookup"
default =
{ "UNKNOWN", "UNKNOWN", "UNKNOWN", "UNKNOWN", "UNKNOWN", "UNKNOWN", "UNKNOWN", "UNKNOWN" }

#####
NYC = registerTarget( %Server, "", "alerts.status" )






```

- d. Scroll down in the file, and add the following lines:

```
[@AgencyId,@SiteId,@SiteName,@SiteAddr,@SiteCity,@SiteState,@SiteCountry]
= lookup(@Node, nyc)
update(@AgencyId)
Update(@SiteId)

@Severity      = $Severity
@Identifier    = $Node + $Agent + $Severity + $Group

[ @AgencyId,@SiteId,@SiteName,@SiteAddr,@SiteCity,@SiteState,@SiteCountry] = lookup(@Node, nyc)
update(@AgencyId)
update(@SiteId)

if (nmatch($Summary, "Port failure"))
{
```

- e. Save the changes, and exit the gedit utility.

11. Verify the syntax of the modified rules file.

```
nco_p_syntax -server NYC_AGG_P -rulesfile simnet.rules
```

Make sure that no syntax errors exist.

12. Switch to the **host1** image.

The two Simnet probes are configured for remote access. The two Simnet probes are configured to use a lookup table. The lookup table does not take effect until the probes reread their rules files.

13. Run the procedure to reread the rules files.

```
nco_sql -server NYC_AGG_P -user root -password object00
1> execute reloadrules_allprobes
2> go
(0 rows affected)
1> quit
```

14. Examine the probe management log file.

```
cd $OMNIHOME/log
tail NYC_AGG_P_probemanagement.log1
```

```
Fri Dec 19 19:55:13 2014 Sent HTTP reload rules request to mttrapd on host1.tivoli.edu:4198
Fri Dec 19 19:55:13 2014 Sent HTTP reload rules request to simnet on host1.tivoli.edu:4190
Fri Dec 19 19:55:13 2014 syntax on host2.tivoli.edu is not running
Fri Dec 19 19:55:13 2014 syntax on host1.tivoli.edu is not running
Fri Dec 19 19:55:13 2014 Sent HTTP reload rules request to simnet on host2.tivoli.edu:4190
Fri Dec 19 19:55:13 2014 Sent HTTP reload rules request to syslog on host1.tivoli.edu:4199
Fri Dec 19 19:55:13 2014 Reload all rules summary: managed probes 4, unmanaged probes 0, probes not running 2
Fri Dec 19 19:55:14 2014 simnet probe on host1.tivoli.edu:4190 : Rules file reread upon HTTP request successful ...
Fri Dec 19 19:55:44 2014 mttrapd probe on host1.tivoli.edu:4198 : Rules file reread upon HTTP request successful ...
Fri Dec 19 19:56:03 2014 syslog probe on host1.tivoli.edu:4199 : Rules file reread upon HTTP request successful ...
```

The procedure sends reload requests to all four probes. Three of the probes reply with a reload successful. There is no message for the Simnet probe on **host2**. The Simnet probe on host2 does reload its rules file. However, the probe is configured for peer-to-peer mode. When the Simnet probe on host1 is running, the Simnet probe on host2 does not send events to the ObjectServer. The event contains the reload successful status.

15. Open a Firefox browser if necessary.

16. Log in as **ncoadmin** with password **object00**.

17. Open the Event Dashboard, and select the monitor box for **Last10Mins**.

18. Right-click any Simnet event, and select **Information**. Locate the **AgencyId** column.

Alert Status for Serial Number 28751	
Fields	Detail
Field	Value
AdvCorrServer...	
AdvCorrServer	0
AgencyId	bma
Agent	MachineMon
AggregationFirst	12/23/14 3:09:07 PM
AlertGroup	Systems
AlertKey	

The lookup command in the modified Simnet rules file populates the column. This value verifies that the Simnet probe did reread the rules file. If the probe did not reread the file, the column would be empty.

19. Close the Information window.

20. Close the Active Event List window.

21. Log out of Dashboard Application Services Hub.

22. Close the Firefox browser.



4 Automations exercises

In this unit, you learn to use some basic SQL commands to explore event records. You create some triggers to implement a rudimentary notification system.



Note: You can use **host1** or **host2** to complete the exercises in this unit.

Exercise 1 Basic SQL commands

In the following exercise, you use some of the basic SQL commands. You use the nco_sql utility and the SQL workbench.

Working with the nco_sql utility

1. Open a Terminal window if necessary.
2. Define the EDITOR environment variable.

```
EDITOR=gedit  
export EDITOR
```

3. Start the utility with the following command:

```
nco_sql -server NYC_AGG_P -user root -password object00  
1>
```

The **1>** indicates a successful connection to the ObjectServer.

4. Select the critical events, and display the Node column:

```
1> select Node from alerts.status where Severity=5;
```

```
2> go
```

```
Node
```

```
host2.tivoli.edu
host1.tivoli.edu
host1.tivoli.edu
host1
host1.tivoli.edu
host1.tivoli.edu
```

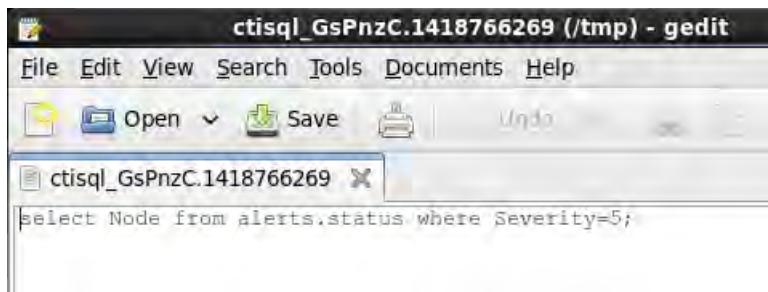
```
(6 rows affected)
```



Note: Your result might not be the same.

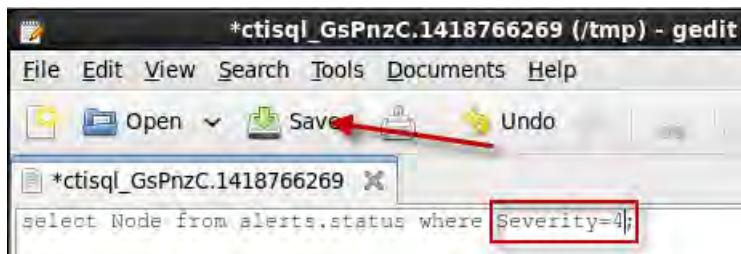
5. Use the gedit editor to modify the previous command:

```
1> gedit
```



The previous command opens in the gedit editor. You can use the editor to modify the command.

6. Change the severity value to 4, and save the changes.



7. Select **File > Quit** to exit the gedit utility.

```
1> select Node from alerts.status where Severity=4;
```

The updated command opens in the nco_sql utility.

8. Enter **go**, and press **Enter** to run the command.

```
1> select Node from alerts.status where Severity=4;  
2> go  
Node  
-----  
host1.tivoli.edu  
Washington  
Moscow  
host1.tivoli.edu  
Sydney  
127.0.0.1  
link4  
London  
  
(8 rows affected)
```

 **Note:** Your result might not be the same.

Incorporating an editor with the nco_sql utility makes it convenient to retrieve, and modify the previous command. You can retrieve only the last command.

 **Note:** You can use the same technique with any editor, such as vi. Set the EDITOR environment variable to use any editor.

9. Display more than one column name in the result set.

```
1> select Node,Tally from alerts.status where Severity=5;  
2> go  
Node                               Tally  
-----  
host2.tivoli.edu                      1  
host1.tivoli.edu                      1  
host1.tivoli.edu                      1  
host1                           2  
host1.tivoli.edu                      1  
host1.tivoli.edu                      1  
  
(6 rows affected)
```

10. Find the number of events for each Node.

```
1> select Node,count(*) from alerts.status group by Node;
2> go
Node                                COL_1
-----
host1                               1070
host1.tivoli.edu                     37
host2.tivoli.edu                     15
Washington                           2
Berlin                               1
```

In this example, there are more than 1000 events for host1.

11. Find the types of events for host1.

```
1> select Manager,count(*) from alerts.status where Node='host1' group by
Manager;
2> go
Manager                                COL_1
-----
Syslog Probe on host1.tivoli.edu      1054
ConnectionWatch                         2
syslog                                  12
OMNIbus Self Monitoring @NYC_AGG_P     1
```

(4 rows affected)

In this example, most events come from the Syslog probe.

12. Delete the Syslog events for host1.

```
1> delete from alerts.status where Node='host1' and Manager like 'Syslog Probe'
;
2> go
(1054 rows affected)
```

13. Verify whether the events are deleted.

```
1> select Manager,count(*) from alerts.status where Node='host1' group by
Manager;
2> go
Manager                                COL_1
-----
ConnectionWatch                         1
syslog                                 12
OMNIbus Self Monitoring @NYC_AGG_P      2
```

(3 rows affected)

The correct events are deleted.



Note: The delete operation cannot be reversed. Make sure that you know what you delete.

14. Exit the nco_sql utility.

```
1> quit
```

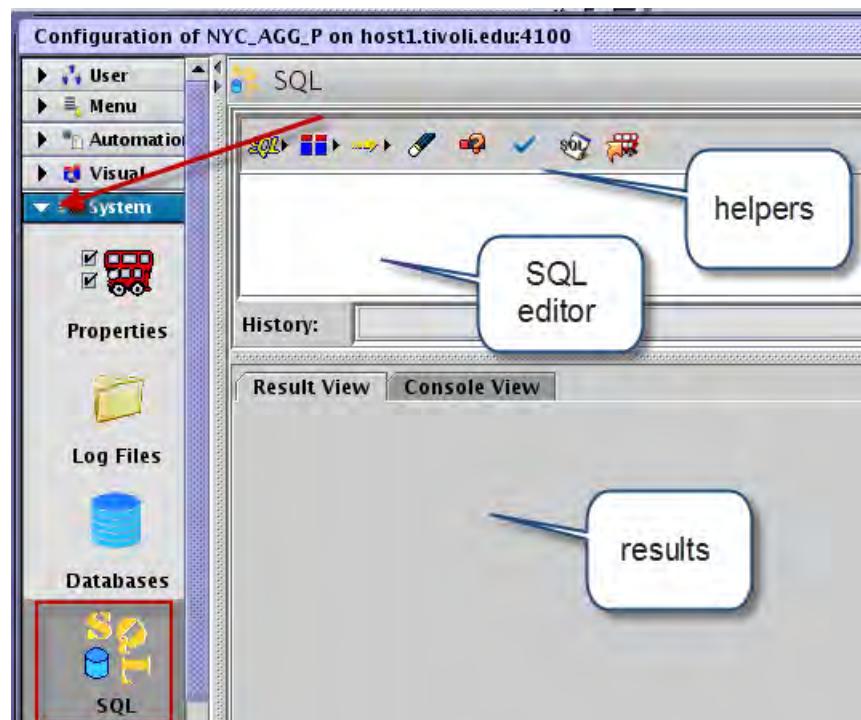
Working with the SQL workbench

1. Start the Netcool/OMNIbus Administrator.

```
nco_config &
```

2. Connect to **NYC_AGG_P**, and log in as **root** with password **object00**.

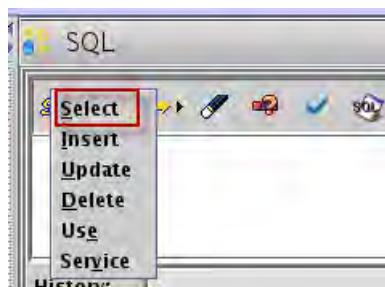
3. Expand **System**, and select **SQL**.



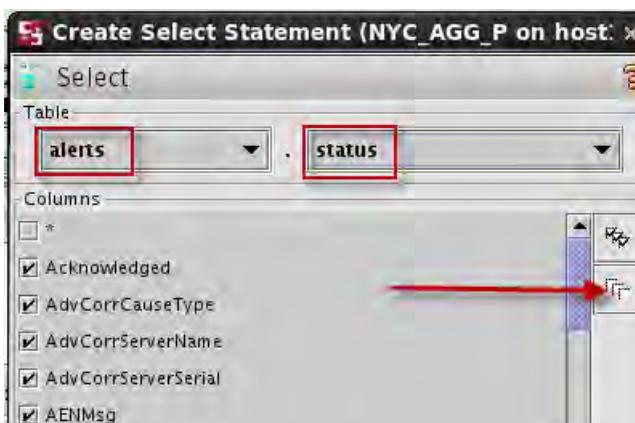
The workbench contains three functional areas. The white space is the SQL editor pane. You enter SQL commands and the output opens in the results pane. The menu bar contains a number of helpers. You use each of the buttons to help create and run an SQL command.

4. Create a simple SQL select command as follows:

- a. Click the **SQL** helper, and then click **Select**.

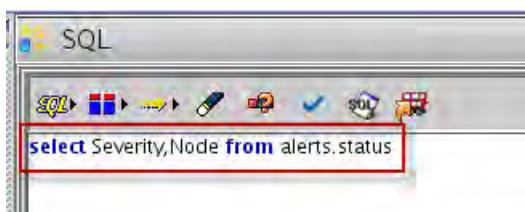


- b. Select **alerts**, and then select **status**. Click the icon to remove the check marks from all of the column names.



All column names are selected by default.

- c. Select **Node**; then select **Severity**, and click **OK**.

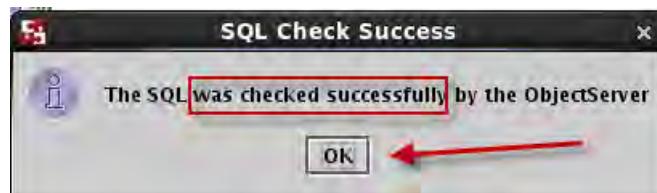


The helper generates the SQL command.

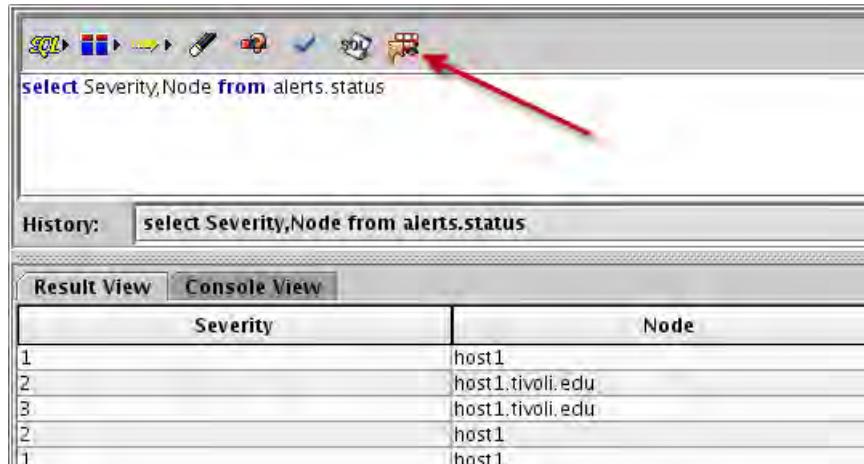
- d. Click the *check mark* icon to verify the syntax.



- e. Click **OK** to close the window.



- f. Click the *omnibus* icon to run the command.



Severity	Node
1	host1
2	host1.tivoli.edu
3	host1.tivoli.edu
2	host1
1	host1

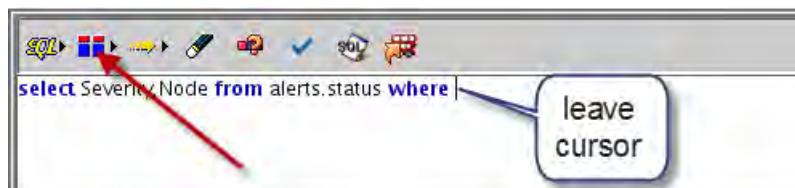
The output opens in the results pane. The command remains in the editor pane.



Hint: Click the *pencil eraser* icon to clear the command from the editor pane.

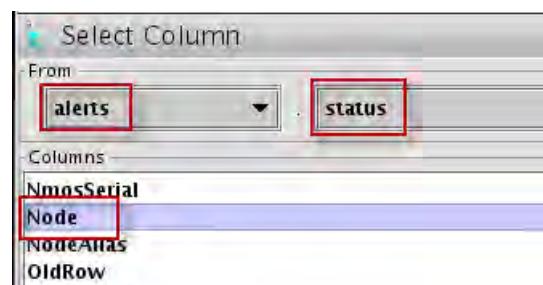
5. Modify the command to select events for host2.

- a. Enter **where**, and click the *column name helper*.



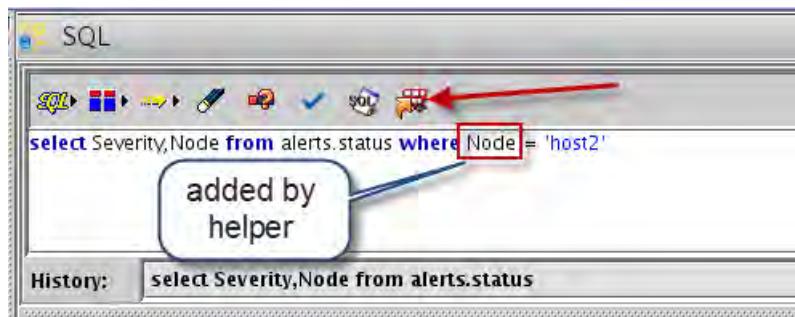
Make sure that you have a space after the where clause. Leave the cursor after the space.

- b. Select **Node**, and click **OK**.



When you click OK, the text for Node is added to the command at the point where you left the cursor.

- c. Enter = 'host2', and click the *omnibus* icon.



To create a complex condition, you can enter more text, for example:

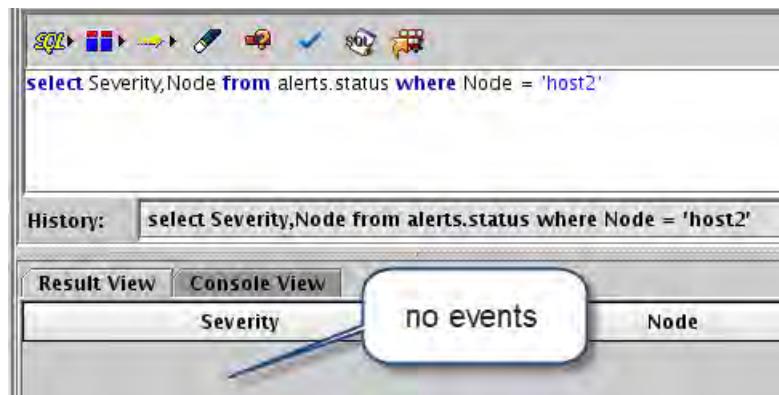
where Node = 'host2' **and AlertGroup = 'XYZ'**

You can use the column helper to create the text for each column name.



Important: The ObjectServer is case-sensitive. Column names contain upper, and lowercase letters. The column helper is convenient to ensure the correct spelling, and capitalization of a column name.

The command does not locate any events because no events contain host2 in the Node column.



- d. Change the *equal sign* to the text **like**, and run the command.

The screenshot shows the IBM Workbench interface. The SQL editor pane contains the following query:

```
select Severity,Node from alerts.status where Node like 'host2'
```

The History pane below it shows the same query. The Result View tab is selected, displaying a table with two columns: Severity and Node. The data is as follows:

Severity	Node
2	host2.tivoli.edu
5	host2.tivoli.edu
2	host2.tivoli.edu

The command finds several records.

6. Click the arrow, and examine the command history.

The screenshot shows the History pane of the IBM Workbench interface. It lists several previous SQL commands. A red arrow points to the dropdown arrow icon at the top right of the History pane, which is used to expand the list.

- select Severity,Node from alerts.status
- select Severity,Node from alerts.status where Node like 'host2'
- Result View select Severity,Node from alerts.status where Node = 'host2'
- select Severity,Node from alerts.status

The workbench retains a history of the commands. You can select any command from the list, and the text opens in the SQL editor pane. Click the *omnibus* icon to run the command.

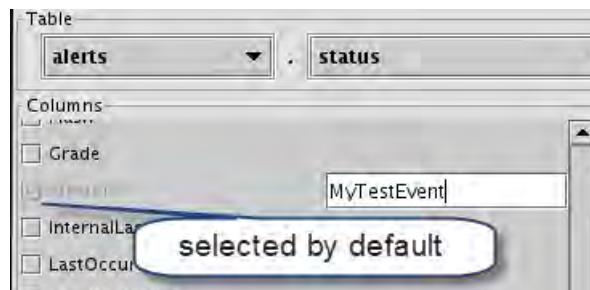
7. Insert an event record as follows:

- Click the *pencil eraser* icon to clear the editor pane.
- Click the *SQL helper* icon, and select **Insert**. Select **alerts**, and then select **status**.

The screenshot shows the Insert dialog box. The Table dropdown menu has "alerts" selected, and the Sub-table dropdown menu has "status" selected. Below the tables, a list of columns is shown, each preceded by a checkbox. A tooltip window is overlaid on the dialog, stating "no column names selected".

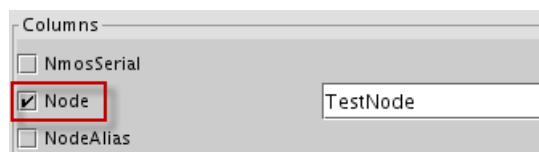
No column names are selected by default. You must select the columns that you want for the new record.

- c. Scroll down, and locate **Identifier**. Enter **MyTestEvent**.



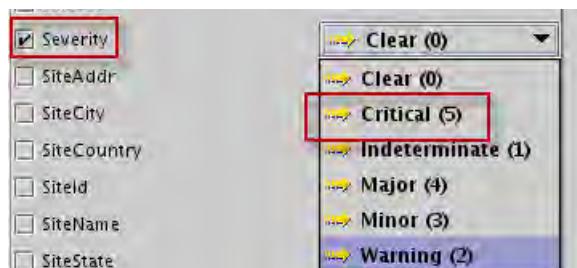
The Identifier column is selected by default. The column name is gray, and you cannot remove the check mark. The Identifier column is the *key* for the event record, and must contain a value.

- d. Scroll down, and select the **Node** column. Enter **TestNode** for the value.



When you select the column, a box opens. You enter text in the box.

- e. Scroll down, and select the **Severity** column. Click the arrow to display the values.



The Severity column is defined as an Integer. The list of values contains text because the conversion table contains values for Severity. The integer value for each severity appears after the corresponding text in the list.

- f. Select Critical for the Severity column.



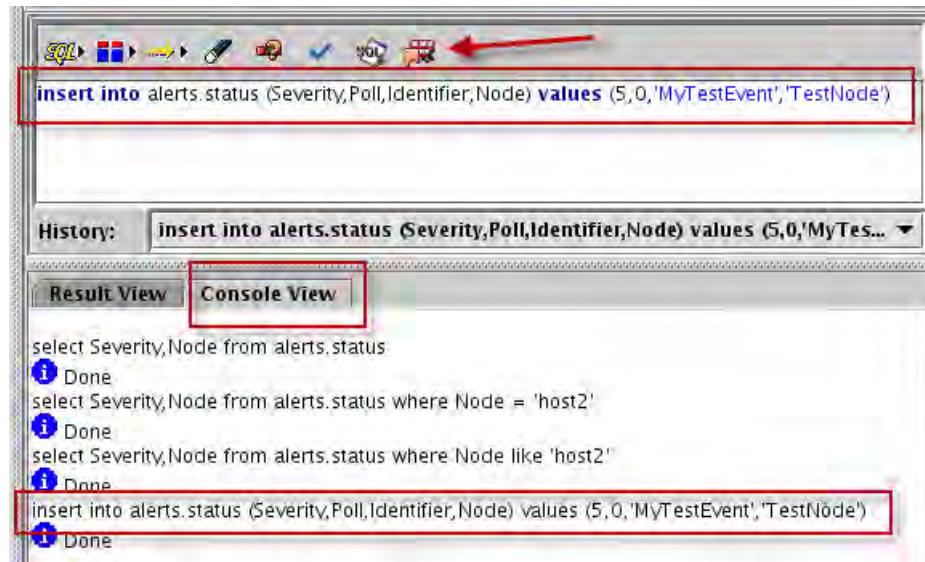
Hint: The default value for severity is 0, or Clear. An ObjectServer automation removes clear events periodically. If you do not select some other value, the new event is created as clear, and the automation might remove the event during this exercise.

- g. Scroll up, and select the **Poll** column.



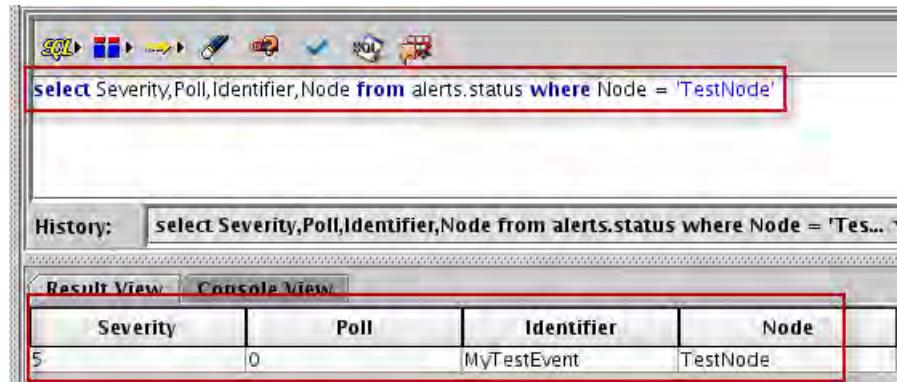
The Poll column is defined as an Integer. However, no values for Poll appear in the conversion table. You enter the value as an integer.

- h. Click **OK**. Click the *omnibus* icon to run the command.



When you run the command, the output pane switches to **Console View**. The console output opens.

- i. Clear the editor window.
j. Create a select command to locate the test event. Click the *omnibus* icon to run the command, and verify that the event is created.



8. Right-click **History**, and select **Clear History**.



The history of commands is now empty.

Leave the Netcool/OMNIbus Administrator utility open. You use it later in the exercise.

Exercise 2 Automations

In the following exercise, you work with temporal and database triggers.

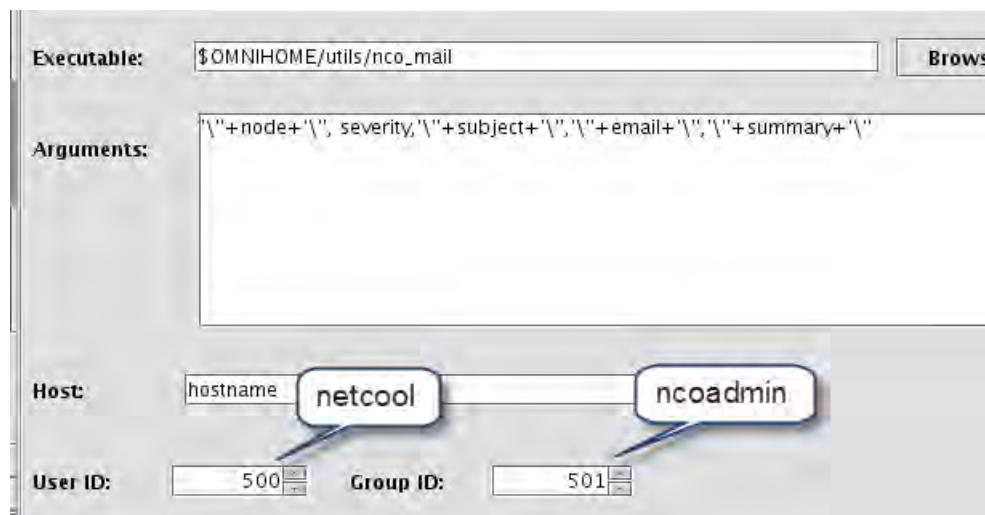
Working with temporal triggers

You added custom columns to the event record, and configured the Simnet probe to populate some of those columns. In this exercise, you create a temporal trigger. The trigger searches for a *customer* event with a critical severity. If the trigger finds an event, the trigger calls a procedure, and the procedure sends an email to notify a user of a customer with a critical situation.

Netcool/OMNIbus includes a sample trigger, and procedure for this behavior. You copy the sample trigger, and modify it.

You create the behavior in both ObjectServers.

1. Expand **Automation**, and select **Procedures**.
2. Right-click **send_email**, and select **Edit Procedure**.
3. Change User ID to **500**. Change Group ID to **501**.



Important: Process activity runs as the **netcool** user in the class images. You must change the User ID in the procedure to 500, which is the value for the netcool user.

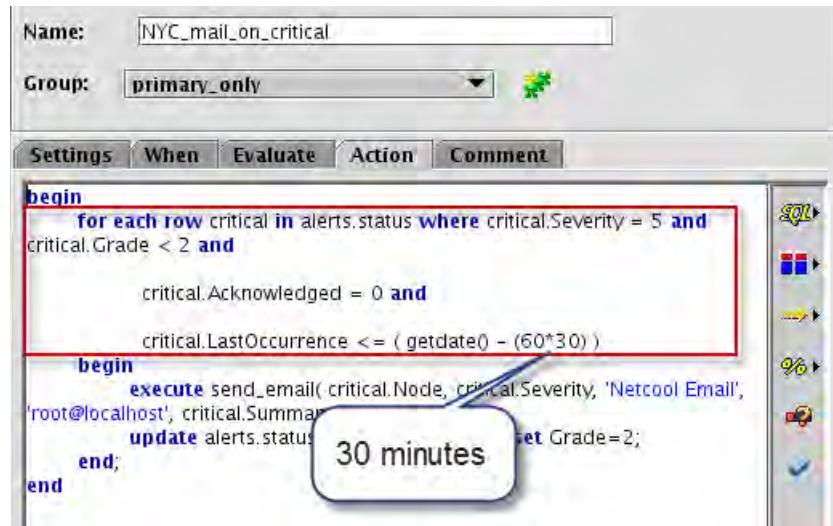
4. Click **OK** to save the changes.
5. Select **Triggers**.
6. Scroll down and locate the trigger that is called **mail_on_critical**.

7. Right-click **mail_on_critical**, and select **Copy**.
8. Right-click, and select **Paste**.
9. Enter **NYC_mail_on_critical** for the name. Change the frequency to **30** seconds. Select **Enabled**.



Important: A low frequency setting causes an increase in processor usage. You should select a frequency that is not too low for what you want to accomplish.

10. Click the **Action** tab.



The default trigger searches for critical events, which are not acknowledged, and are more than 30 minutes old. In a production environment, 30 minutes is probably reasonable. For the purposes of this exercise, make the age requirement lower. You use a lower age requirement so that you see the result of the automation sooner. You also must add a condition to test for a *customer* event. You can test the value of AgencyId to meet this requirement.



Important: Notice the test for the **Grade** column. This condition is used to ensure that the automation selects the same event record only once. When the automation locates an event, the automation calls the procedure to send the email, and sets the value of Grade to 2. This value ensures that the automation does not select the same event again.

11. Add the test for the **AgencyId** as shown here. Change the test for **LastOccurrence** as shown here.

```

Settings When Evaluate Action Comment
begin
    for each row critical in alerts.status where critical.Severity = 5 and
critical.Grade < 2 and

        critical.Acknowledged = 0 and
        critical.AgencyId <> "and"
        critical.LastOccurrence <= ( getdate() - (60*1) )
begin
    execute send_email( critical.Node, critical.Severity, 'Netcool Email',
'root@localhost', critical.Summary,
update alerts.status
end;
end;

```

The screenshot shows the 'Action' tab of the Automation Settings window. The script code is displayed. A red box highlights the condition 'critical.AgencyId <> "and"'. Another red box highlights the date calculation 'getdate() - (60*1)'.

12. Change the email address as shown here.

```

critical.LastOccurrence <= ( getdate() - (60*1) )
begin
    execute send_email( criti
cal.Node, critical.Severity, 'Netcool Email', 'netcool@localhost', critical.Summary,
'localhost');
    update alerts.status via critical.Identifier set Grade=2;
end;

```

The screenshot shows the 'Action' tab of the Automation Settings window. The email address 'netcool@localhost' is highlighted with a red box.

13. Add the line to create a journal entry as shown here. Click the *green check mark* to validate the syntax. Click **OK** to save the new trigger.

```

critical.LastOccurrence <= ( getdate() - (60*1) )
begin
    execute send_email( criti
cal.Node, critical.Severity, 'Netcool Email', 'netcool@localhost', critical.Summary,
'localhost');
    update alerts.status via critical.Identifier set Grade=2;
    call jinsert(critical.Serial, %user.user_id, getdate, 'Email sent
via NYC_mail_on_critical trigger');

```

The screenshot shows the 'Action' tab of the Automation Settings window. A red box highlights the new line 'call jinsert(critical.Serial, %user.user_id, getdate, 'Email sent via NYC_mail_on_critical trigger');

14. Leave the Netcool/OMNIbus administrator utility open.

15. Open a Firefox browser.

16. Log in to Dashboard Application Services Hub as **ncoadmin** with password **object00**.

17. Open the **Event Dashboard** page.

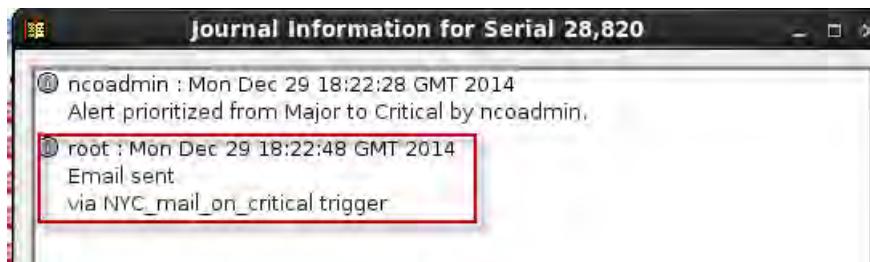
18. Click the box for **Last10Mins**.

The simulated customer events do not contain any critical events.

19. Right-click any *customer* event, select **Prioritize > Critical**.



20. After a short time, right-click the same event, and select **Journal**.



The reprioritize tool creates the first journal entry. The trigger creates the second journal entry.

21. Click **OK** to close the journal window.

22. Close the Active Event List window.

23. Log out of Dashboard Application Services Hub

24. Close the Firefox browser.

25. Open a Terminal window if necessary.

26. Check the email messages as follows:

mail

```
---- ---- ---- ---- ----  
[netcool@host1 linux2x86]$ mail  
Heirloom Mail version 12.4 7/29/08. Type ? for help.  
"/var/spool/mail/netcool": 1 message 1 new  
:N 1 netcool@host1.tivoli Mon Dec 29 18:22 20/635 "Netcool Email"
```

The netcool user has one new email message with a subject of **Netcool Email**.

27. Enter **1**, and press **Enter**.

```
Delivered-To: netcool@localhost.tivoli.edu
To: netcool@localhost.tivoli.edu
Subject: Netcool Email
Date: Mon, 29 Dec 2014 18:22:50 +0000 (UTC)
From: netcool@host1.tivoli.edu
Status: R

This message refers to node nsm123-b1.bma.gov which has the following problem;

Diskspace alert

The Severity is 5

Sent by the Netcool/OMNIbus Automation system
```

NYC_mail_on_critical trigger locates the event record and calls the **send_email** procedure. The **send_email** procedure creates the email message. The trigger updates the Grade column in the event record to prevent another email.

28. Enter **q**, and press **Enter** to exit the mail utility.

29. Return to the Netcool/OMNIbus Administrator utility.

30. Right-click **NYC_AGG_B**, and select **Connect As**.

31. Log in as **root** with password **object00**.

32. Expand **Automation**, and select **Procedures**.

33. Edit the **send_email** procedure, and change User ID to **500** and Group ID to **501**. Click **OK** to save the changes.

34. Select **Triggers**.

35. Copy the **NYC_mail_on_critical** trigger from **NYC_AGG_P**, and paste the trigger into **NYC_AGG_B**.

36. Enter **NYC_mail_on_critical** for the name, and click **OK** to save the trigger.



The remaining settings are retained from the copy operation.



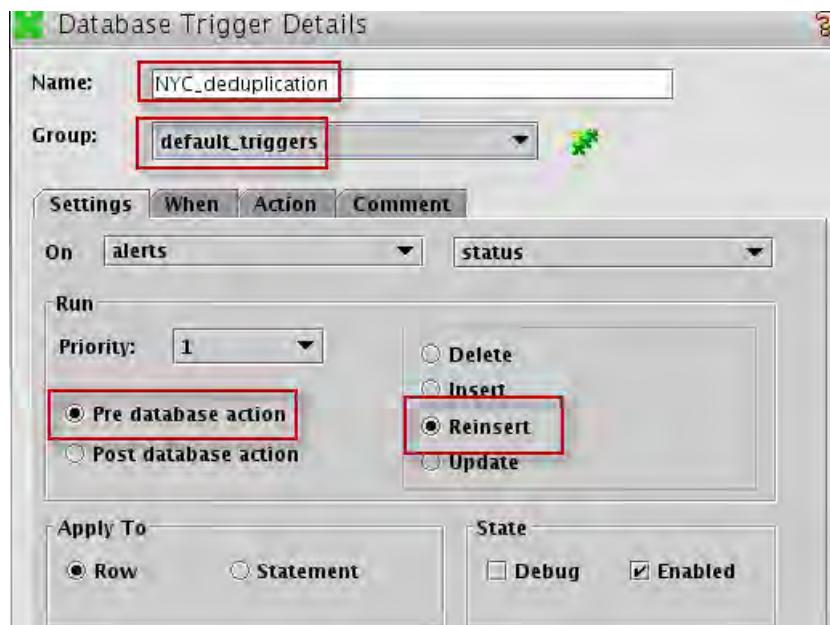
Important: The Group must be set to **primary_only**. The primary_only trigger group is disabled on the backup ObjectServer when the primary ObjectServer is active. The membership in the primary_only trigger group prevents the NYC_mail_on_critical trigger from generating email messages when the primary ObjectServer is active.

Working with database triggers

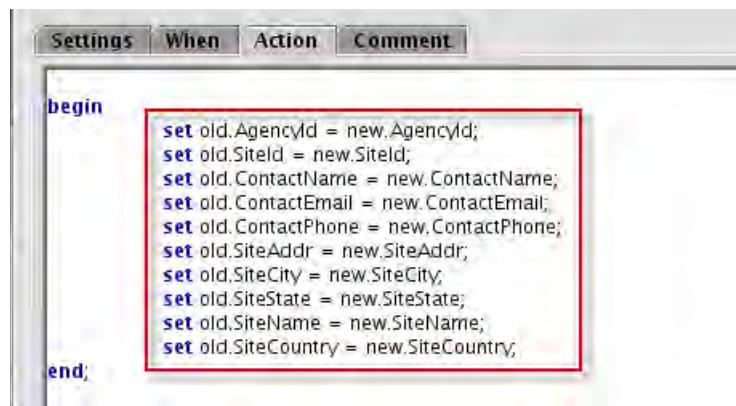
The deduplication trigger is designed to update some columns when a reinsert database action occurs. You added more columns to the event record in a previous exercise. You want the new columns to update based on a reinsert action. In the following steps, you create a database trigger to ensure that the columns update whenever a reinsert action occurs.

You create the behavior in both ObjectServers.

1. Return to the Netcool/OMNIbus administrator utility.
2. Select the **NYC_AGG_P** ObjectServer.
3. Select **Triggers**.
4. Right-click, and select **Add Database Trigger**.
5. Enter **NYC_deduplication** for the name. Select **default_triggers** for the group. Select **Reinsert**.



6. Click the **Action** tab. Enter the commands as shown here. Click the *green check mark* to validate the syntax. Click **OK** to save the trigger.



```
begin
  set old.AgencyId = new.AgencyId;
  set old.SiteId = new.SiteId;
  set old.ContactName = new.ContactName;
  set old.ContactEmail = new.ContactEmail;
  set old.ContactPhone = new.ContactPhone;
  set old.SiteAddr = new.SiteAddr;
  set old.SiteCity = new.SiteCity;
  set old.SiteState = new.SiteState;
  set old.SiteName = new.SiteName;
  set old.SiteCountry = new.SiteCountry;
end;
```

7. Right-click **NYC_deduplication**, and select **Copy**.
8. Open the **NYC_AGG_B** ObjectServer, and select **Triggers**.
9. Right-click, and select **Paste**.
10. Enter **NYC_deduplication** for the name. Click **OK** to save the trigger.
11. Select **File > Exit** to close the Netcool/OMNIbus Administrator utility.



5 Web GUI administration exercises

In the following exercises, you perform some of the common Web GUI administrative functions. You create filters, and views that incorporate custom columns that you added in the previous exercise. You create some desktop tools, a simple map, and a gauge. You use the Web GUI API client to display configuration information that is related to the Web GUI server.

Exercise 1 Creating filters, and views

In a previous exercise, you added extra column names to the ObjectServer event record. In the following steps, you create a view, which incorporates some of those column names. In addition, you use some of the columns to configure event grouping.



Important: You perform the following exercises on **host2**.

Updating grouping columns

Before you can use the columns for event grouping, you must configure Web GUI to allow those column names.

1. Open a Terminal window if necessary.
2. Change to the target directory.
`cd /opt/IBM/netcool/omnibus_webgui/etc`
3. Save a copy of the file before changes.
`cp server.init server.init.orig`
4. Modify the file as follows:
 - a. Open the file for edit with the gedit utility.
`gedit server.init`

- b. Locate the following line:

```
columngrouping.allowedcolumns=Acknowledged,AlertGroup,Class,Customer,Location,Node,NodeAlias,NmosCauseType,NmosManagedStatus,Severity,Service
```

- c. Modify the line as shown:

```
columngrouping.allowedcolumns=Acknowledged,AlertGroup,AgencyId,SiteId,Class,Customer,Location,Node,NodeAlias,NmosCauseType,NmosManagedStatus,Severity,Service
```

Add **AgencyId** and **SiteId** to the list. Their location in the list is not relevant.

- d. Save the file, and exit gedit.

5. Stop Dashboard Application Services Hub.

```
cd /opt/IBM/JazzSM/profile/bin/  
./stopServer.sh server1 -user smadmin -password object00
```

Wait for the server to stop.

6. Start Dashboard Application Services Hub.

```
./startServer.sh server1
```

Wait for the server to start.

Creating a view

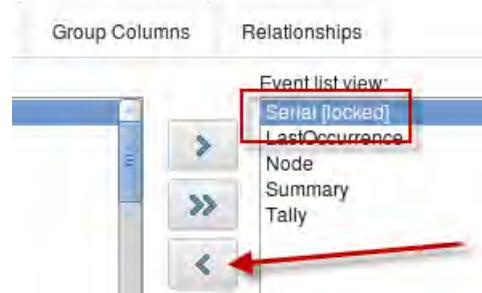
1. Open a Firefox browser if necessary.
2. Log in to Dashboard Application Services Hub as **ncoadmin** with password **object00**.
3. Click the icon to open the Administration feature, and select **Views**.



4. Click the icon to create a view.



5. Select **global**, and click **OK**.
6. Enter **NYC** for the name.
7. Click **Serial**, and then click the *left arrow* icon to remove the column from the view.



8. Click **AgencyId**, and then click the *right arrow* icon to add the column to the view.



9. Add the following columns to the view:

ContactEmail
ContactName
ContactPhone
SiteAddr
SiteCity
SiteCountry
SiteId
SiteName
SiteState

10. Click **AgencyId** and click the *up arrow* icon. Click the *up arrow* until the column is at the top of the list.

Event list view:
AgencyId
LastOccurrence
Node
Summary
Tally
ContactEmail
ContactName
ContactPhone
SiteAddr
SiteCity

11. Select *Lock column* to prevent the AgencyId column from scrolling.

Event list view:
AgencyId [locked]
LastOccurrence
Node
Summary
Tally
ContactEmail
ContactName
ContactPhone
SiteAddr
SiteCity

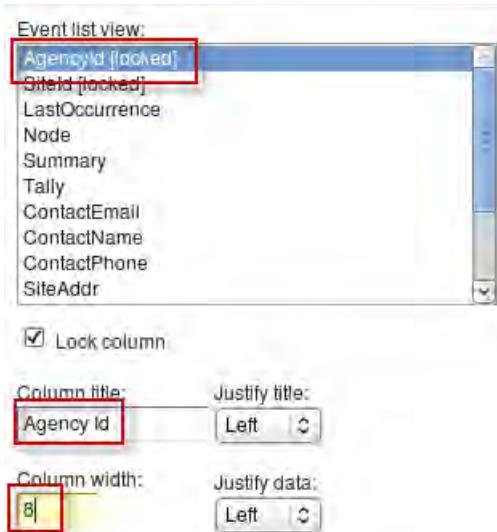
Lock column

12. Move the **SitId** column below AgencyId, and lock the column.

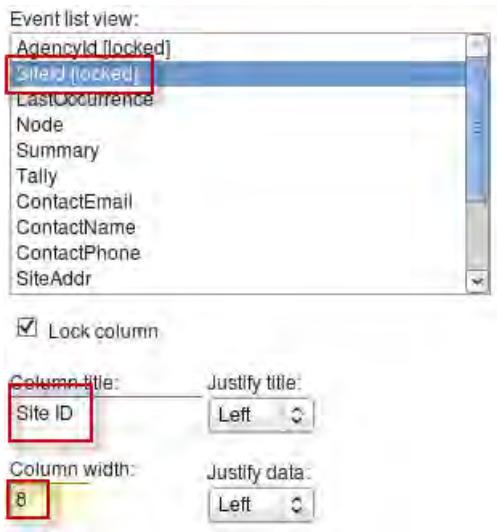
Event list view:
AgencyId [locked]
SitId [locked]
LastOccurrence
Node
Summary
Tally
ContactEmail
ContactName
ContactPhone
SiteAddr

Lock column

13. Click **AgencyId**, change the column title to **Agency ID**, and the column width to **8**.



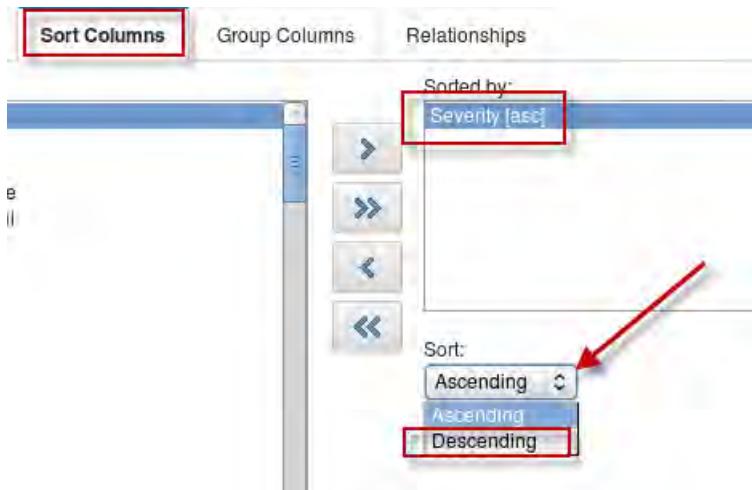
14. Click **SiteId**, change the column title to **Site ID**, and the column width to **8**.



15. Configure the remaining columns as shown here:

LastOccurrence	Last Occurrence	10
Node	Node	10
Summary	Summary	40
Tally	Tally	10
ContactEmail	Contact Email	10
ContactName	Contact Name	10
ContactPhone	Contact Phone	10
SiteAddr	Site Addr	10
SiteCity	Site City	10
SiteCountry	Site Country	10
SiteName	Site Name	10
SiteState	Site State	10

16. Click **Sort Columns**. Click **Severity**, and select **Descending**.



17. Click **Group Columns**. Add **AgencyId** and **Siteld**.



18. Click **Save and Close**.

19. Change the display to **Global Views**, and verify whether **NYC** appears in the list.



20. Click the **X** to close the Views page.

Creating filters

The AgencyId column name appears in the ObjectServer event record. In the previous exercise, you configured a probe to populate the AgencyId, and SitId columns. The probe is configured to use the following values for AgencyId:

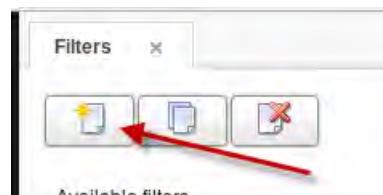
bma
dca
dpp
iea
oaca

In the following steps, you create filters to capture events for each of the possible AgencyId values.

1. Click the icon to open the Administration feature, and select **Filters**.



2. Click the icon to create a filter.



3. Select **global**, and click **OK**.
4. Enter **NYC_BMA** for the name. Click the arrow and select **NYC** for the view.

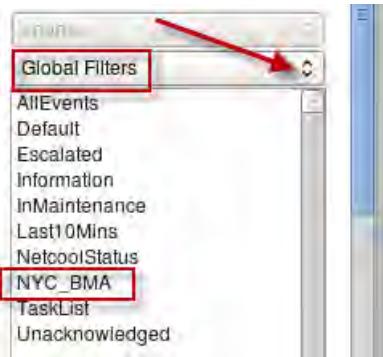
Name:	NYC_BMA
Default view:	NYC

5. Click the arrow, and select **AgencyId** for field. Click the arrow and select the *equal sign* for comparator. Enter **bma** for the value. Click **Save and Close**.



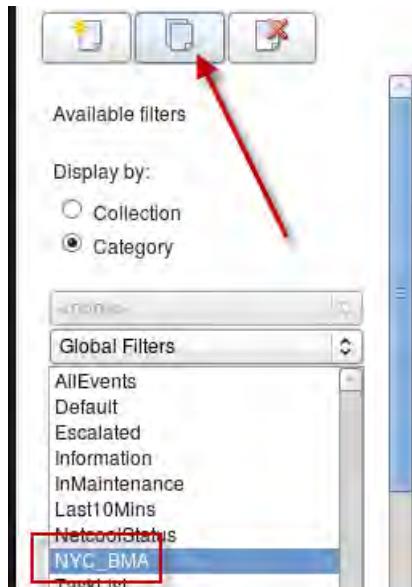
Important: Because you use the equal comparison, the value must be entered as **bma**. The equal comparison denotes an exact match.

6. Click the arrow and select **Global Filters**.



The NYC_BMA filter is created.

7. Click **NYC_BMA** to select it, and click the icon to copy the filter.



8. Select **global**, and click **OK**.

9. Change the name to **NYC_DPP**.



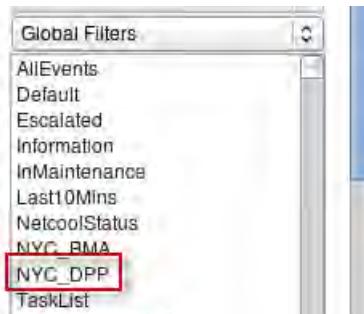
The default view of NYC is retained with the copy operation.

10. Change the text in the value field to **dpp**, and click **Save and Close**.



The AgencyId field and equal comparator are retained with the copy operation.

11. Observe the list of filters.



12. Repeat the copy operation, and create filters for the remaining agencies:

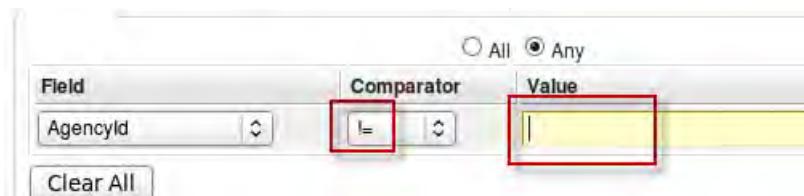
oaca
iea
dca

Create one last filter to capture events for all agencies.

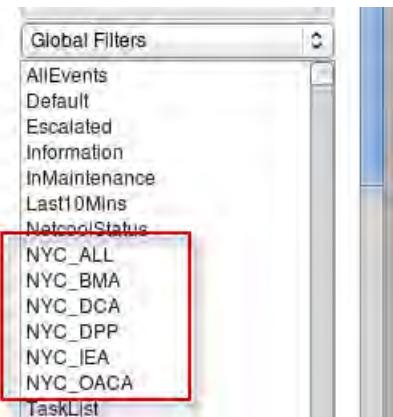
13. Copy the **NYC_BMA** filter, select **global**, and click **OK**.

14. Enter **NYC_ALL** for the name.

15. Change the comparator to *not equal* (**!=**), and remove the text from the value field. Click **Save and Close**.

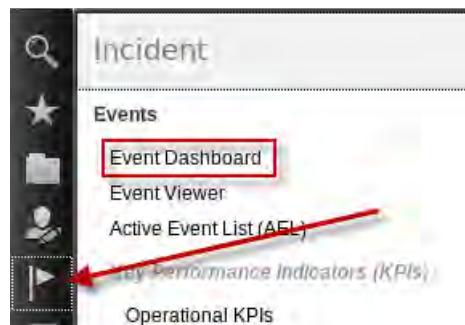


16. The complete list of filters appears as follows:

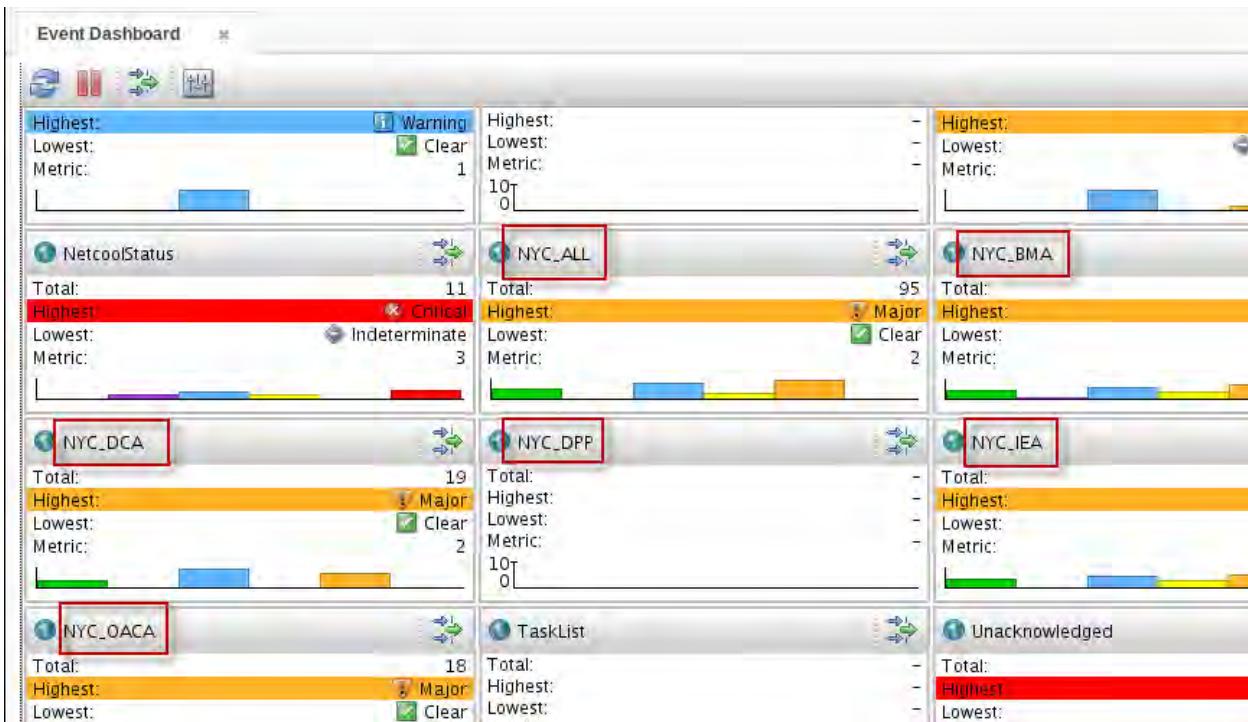


17. Click the X to close the Filters page.

18. Click the icon, and select Event Dashboard.



19. Observe the monitor boxes for the filters. Click the box that is labeled NYC_ALL.



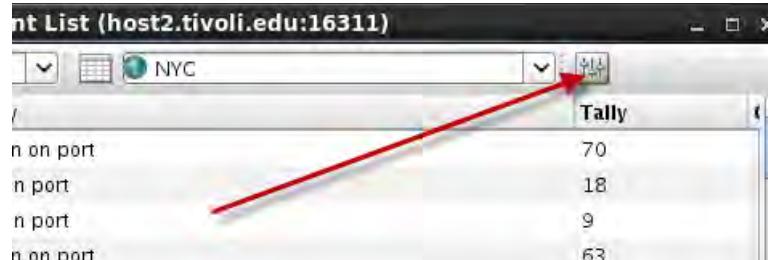
20. Observe the Active Event List.

Agency Id	Site ID	Last Occurrence	Node	Summary
oaca	oaca002	12/29/14 8:1...	2605-ce1-s.o...	Link Up on port
oaca	oaca001	12/29/14 8:1...	2605-ce1-s.o...	Link Up on port
dca		12/29/14 8:1...	NYC-dca.gov	Link Up on port
iea		12/29/14 8:1...	2605-ce1-s.ie...	Link Up on port
dca		12/29/14 8:1...	MOS-dca.gov	Link Down on port
bma	bma002	12/29/14 8:1...	2605-ce1-s.b...	Link Up on port
dca	dca010	12/29/14 8:1...	BRU-dca.gov	Link Up on port

The event list is configured with the NYC_ALL filter and NYC view. The AgencyId and SiteId columns are locked, and do not scroll.

The individual event records do not show the color based on severity by default. Change the property setting to add the color.

21. Expand the size of the Active Event List window until the *property icon* is visible. Click the icon to open the property settings.



22. Click the **Event List** tab. Select **Show Colors**. Click **Save**. Click **Close**.



The color of each event record now reflects the severity of the event.

Agency Id	Site ID	Last Occurrence	Node	Summary
UNKNOWN	UNKNOWN	12/19/14 9:4...	link6	Link Down on port
UNKNOWN	UNKNOWN	12/23/14 3:0...	Washington	Machine has gone offline
UNKNOWN	UNKNOWN	12/23/14 3:0...	Moscow	Machine has gone offline
bma	bma002	12/29/14 7:5...	nsm123-b1.b...	Diskspace alert
dca	dca010	12/29/14 8:2...	BRU-PE1-dca...	Link Down on port

23. Close the event list window.

24. Click the X to close the Event Dashboard.
 25. Log out of Dashboard Application Services Hub.
- Leave the browser open. You use it again shortly.

Exercise 2 Creating prompts, and tools

In the following steps, you create a few simple tools.



Important: You perform the following exercises on **host2**.

Configuring command tool support for the Event Viewer

By default, the Event Viewer does not support command tools. You can install a custom Tivoli Netcool/OMNIbus Web GUI Tool Launch plug-in to run command tools from the Event Viewer. The following steps illustrate the process that is required to install the plug-in on a Firefox browser. The steps for Internet Explorer are different.

The Tivoli Netcool/OMNIbus Web GUI Tool Launch plug-in for Firefox is an NPAPI plug-in. The required plug-in library files are stored on the Web GUI server and downloaded to the client browser as part of the installation procedure. Because Web GUI uses an SSL connection to the server, and typically the server uses a self-signed certificate, the default setting of Firefox does not allow the plug-in to be downloaded and installed directly. To change the setting, take the following actions:

1. Change the URL in the Firefox browser to **about:config**, and press **Enter**.
2. Click the button to continue.



This might void your warranty!

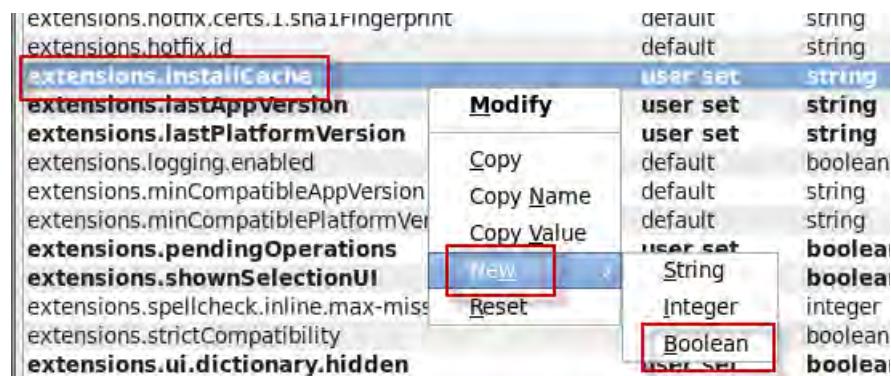
Changing these advanced settings can be harmful to the stability, security, and performance of this application. You should only continue if you are sure of what you are doing.

Show this warning next time

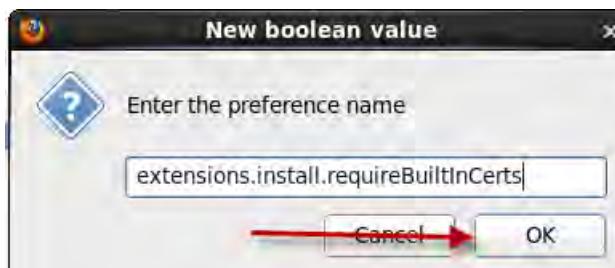
I'll be careful, I promise!

3. Scroll down in the list, and locate **extensions.installCache**.

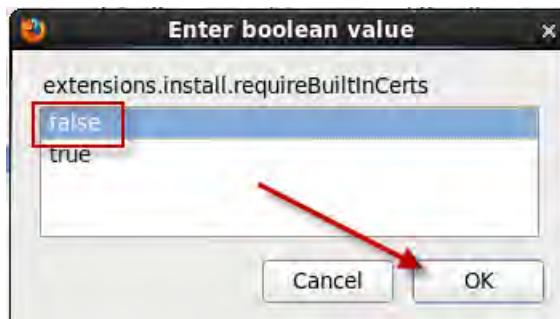
- Right-click **extensions.installCache**, and select **New > Boolean**.



- Enter **extensions.install.requireBuiltInCerts** for the name, and click **OK**.



- Select **false**, and click **OK**.



- Verify that the value is **false**.



The warning icon is located at the top left of the table area.

extensions.notnx.cert.checkAttributes	default	boolean	true
extensions.hotfix.certs.1.sha1Fingerprint	default	string	CA:C4:7D:BF:6
extensions.hotfix.id	default	string	firefox-hotfix@
extensions.install.requireBuiltInCerts	user set	boolean	false
extensions.installCache	user set	string	{ "name": "a
extensions.lastAppVersion	user set	string	17.0.3
extensions.lastPlatformVersion	user set	string	17.0.3
extensions.logging.enabled	default	boolean	false

Important: In your environment, the property might exist. If the property exists, set the value to false.

- Close the Firefox browser.

- Open the Firefox browser.

The Web GUI server is configured to disable command tools for the Event Viewer by default. You must change a Web GUI property to enable command tools for the Event Viewer.

10. Open a Terminal window if necessary

11. Change to the target directory.

```
cd /opt/IBM/netcool/omnibus_webgui/etc
```

12. Save a copy of the file before changes.

```
cp server.init server.init.orig
```

13. Open the file for edit with the gedit utility.

```
gedit server.init
```

14. Locate the following property.

```
eventviewer.tools.command:false
```



Hint: The property setting is near the end of the file.

15. Change the property to **true**.

```
eventviewer.tools.command:true
```

16. Save the file, and exit gedit.

17. Stop Dashboard Application Services Hub.

```
cd /opt/IBM/JazzSM/profile/bin  
.stopServer.sh server1 -user smadmin -password object00
```

Wait for the server to stop.

18. Start Dashboard Application Services Hub.

```
.startServer.sh server1
```

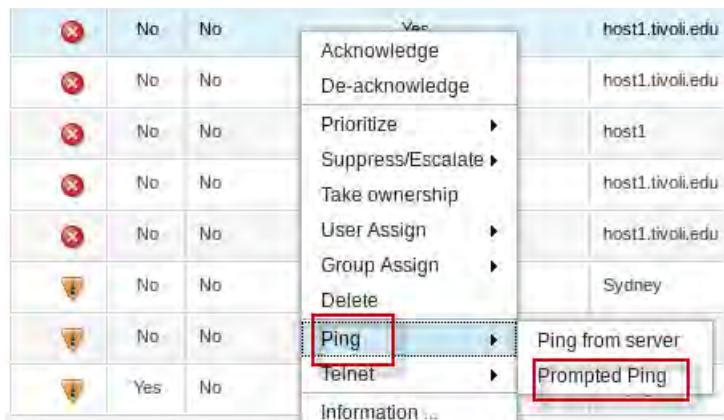
Wait for the server to start.

The following steps are used to install the plug-in.

1. Log in to Dashboard Application Services Hub as **ncoadmin** with password **object00**.
2. Click the icon, and select **Event Viewer**.



3. Right-click any event, and select **Ping > Prompted Ping**.



Note: If the browser does not allow pop-ups, configure the browser for pop-ups, and rerun the tool.

4. Enter **host1** in the prompt window, and click **Run**.

A screenshot of a prompt window. It has a label 'Name of the host' followed by an input field containing the value 'host1'. The input field is highlighted with a red box.

A Firefox window opens with an error message.

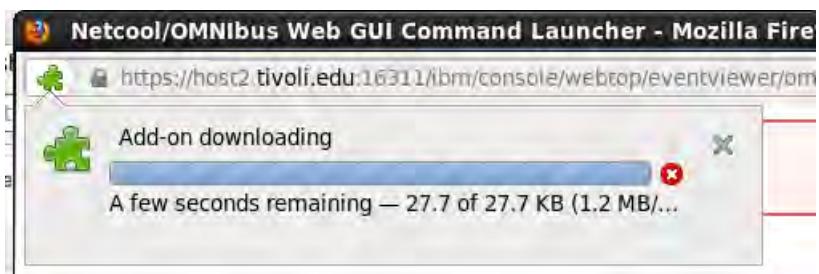
5. Click **Install plugin for 64-bit browser**.



6. Click **Allow**.



7. Observe the message about the download.



After a short time, the message goes away. A new message appears.

8. Click **Install Now**.



9. Click **Restart Now**.



The Firefox browser closes, and starts. The following message appears.

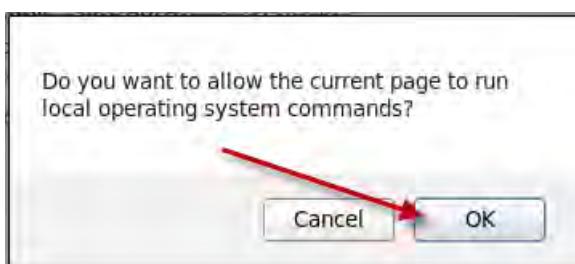
10. Close the message window.



11. Right-click an event, and select **Ping > Prompted Ping**.

12. Enter **host1** for the prompt value, and click **Run**.

13. Click **OK**.



A Terminal window opens, and the ping results appear.

14. Close the Terminal window.

```
PING host1.tivoli.edu (192.168.100.160) 56(84) bytes of data.  
64 bytes from host1.tivoli.edu (192.168.100.160): icmp_seq=1 ttl=64 time=10.1 ms  
64 bytes from host1.tivoli.edu (192.168.100.160): icmp_seq=2 ttl=64 time=0.245 ms  
s  
64 bytes from host1.tivoli.edu (192.168.100.160): icmp_seq=3 ttl=64 time=0.366 ms  
s  
64 bytes from host1.tivoli.edu (192.168.100.160): icmp_seq=4 ttl=64 time=0.268 ms  
s  
64 bytes from host1.tivoli.edu (192.168.100.160): icmp_seq=5 ttl=64 time=0.308 ms  
s  
64 bytes from host1.tivoli.edu (192.168.100.160): icmp_seq=6 ttl=64 time=0.291 ms  
s  
64 bytes from host1.tivoli.edu (192.168.100.160): icmp_seq=7 ttl=64 time=0.221 ms
```

The Firefox browser is configured with the plug-in. Web GUI is configured to enable command tool support for the Event Viewer. When the same tool runs again, the user enters the prompt value, and clicks **Run**.

Leave the Event Viewer page open. You use it again shortly.

Enabling command tools for Linux

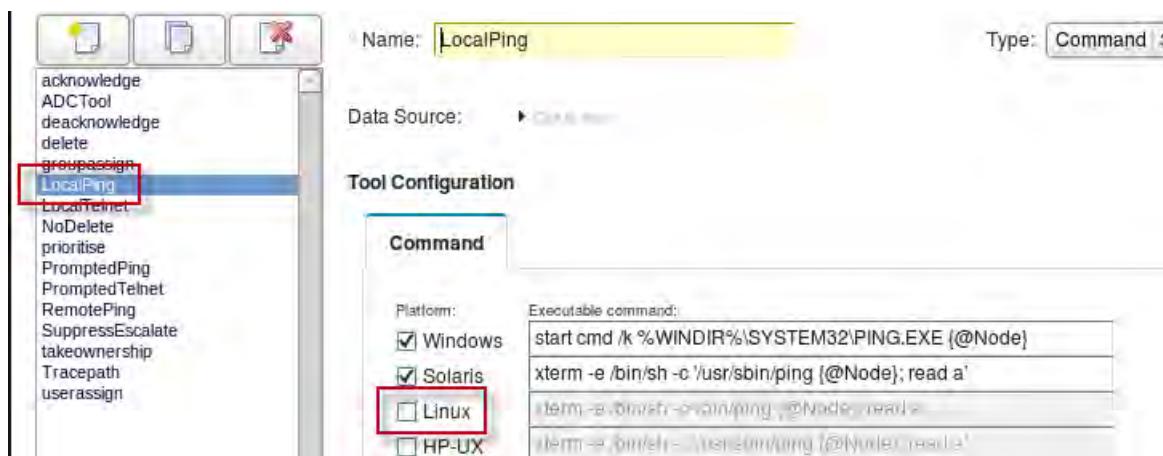
Tivoli Netcool/OMNIbus Web GUI includes several command tools. Most of those tools are not enabled for use on Linux operating systems. The following steps enable the tools for use on the Linux classroom images.

1. Click the icon, and select Tool Configuration.



The list of tools opens.

2. Click LocalPing.



The tool is not enabled for Linux.

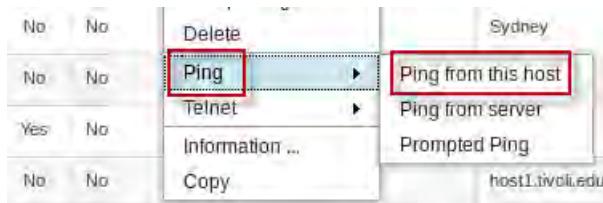
3. Enable Linux, and click Save.



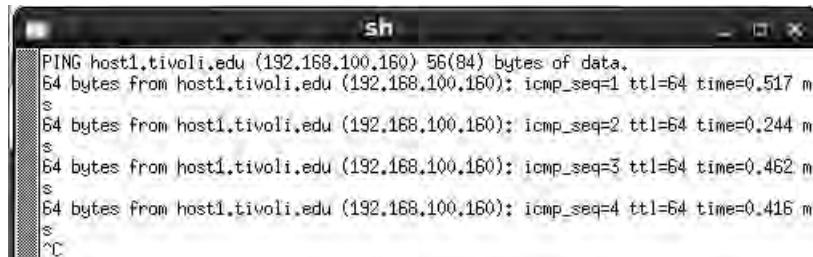
4. Click OK.



5. Click the tab to return to the Event Viewer page.
6. Right-click any event for host1 or host2, and select **Ping > Ping from this host**.



7. A Terminal window opens, and the results from the ping appear. Close the Terminal window.

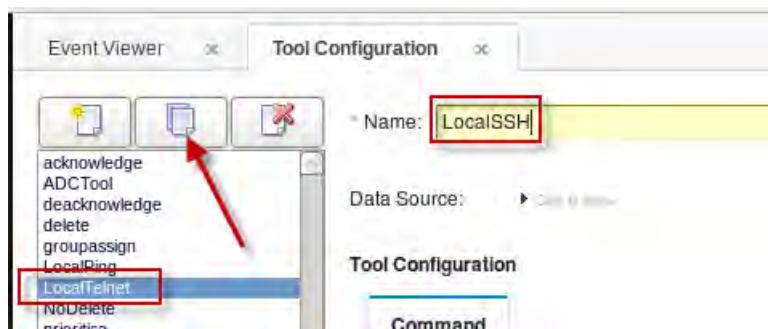


Note: The LocalPing tool extracts the host name from the Node column in the selected event. Your results might be different based on the event that you select.

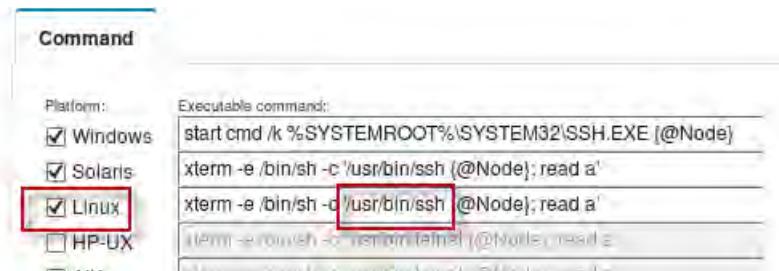
Creating a command tool

Tivoli Netcool/OMNIbus Web GUI includes a sample Telnet tool. Many companies no longer allow the use of non secure tools, such as Telnet. Instead, the companies require the use of SSH.

1. Click **LocalTelnet**, and click the icon to copy the tool. Change the name to **LocalSSH**.



2. Enable **Linux**, and change the command to use **/usr/bin/ssh**. Click **Save**.



3. Click **OK**.

Adding a tool to a menu

Add the SSH tool to a menu.

1. Click the icon, and select **Menu Configuration**.

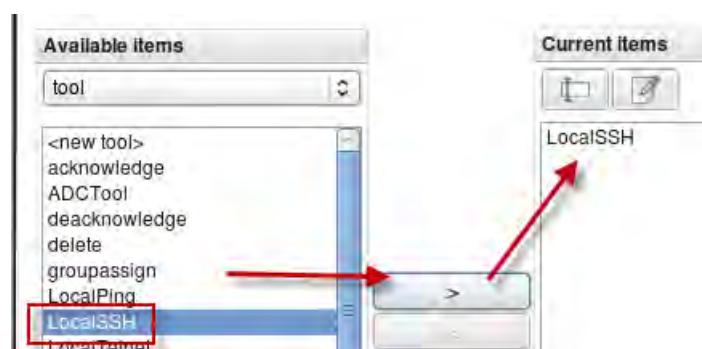


The list of menus opens. Create a menu for the SSH tool.

2. Click **New** on the bottom of the page.
 3. Enter **SSH** for the name, and enter **SSH** for the label.



4. Click **LocalSSH**, and then click the *right arrow* to add the tool to the menu. Click **Save**.



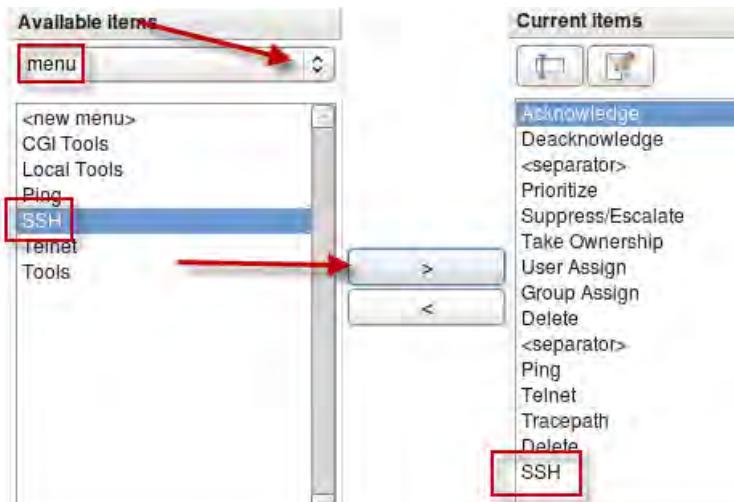
5. Click **OK**.



Add the menu to the **alerts** menu.

6. Click **alerts**, and click **Modify**.

7. Click the arrow, and select **menu**. Click **SSH**, and click the *right arrow* to add the menu to the menu. Click **Save**.



8. Click **OK**.

9. Click the tab to return to the Event Viewer page.

10. Right-click any event for host1 or host2, and select **SSH > LocalSSH**.



A Terminal window opens with the SSH session.

11. Enter **yes** to continue. Enter **object00** for the password.



A screenshot of a terminal window titled "netcool@host1:~". The window displays the following text:
The authenticity of host 'host1.tivoli.edu (192.168.100.160)' can't be established.
RSA key fingerprint is 1c:2c:83:be:ca:fd:a4:86:14:29:1c:29:7c:b5:af:55.
Are you sure you want to continue connecting (yes/no)? **yes** [REDACTED]
Warning: Permanently added 'host1.tivoli.edu,192.168.100.160' (RSA) to the list of known hosts.
netcool@host1.tivoli.edu's password: [REDACTED]
[netcool@host1 ~]\$ [REDACTED]

Note: The tool is configured to extract the host name from the Node column in the selected event record. If you receive a host not found error, select a different event.

12. Close the Terminal window.

Creating an SQL tool, and prompt

A dynamic choice prompt provides a great deal of flexibility in tool design. The values that appear when the user runs the tool are retrieved from an ObjectServer table. You can change the values in the table, and the user sees the new values the next time they use the tool. There is no need to change the tool or the prompt.

Flashing is a technique that can be used to bring attention to a specific event. A column in the event record that is called Flash, accepts a value of 0 or 1. A value of 1 causes the event to flash on, and off. For this exercise, you create a custom table in the ObjectServer for use with a dynamic choice prompt. You use the prompt in a tool to set flashing on, or set flashing off. To facilitate the exercise, you use a supplied file that contains the commands to create the table, and its contents.

Note: Event flashing is not supported in the Event Viewer. Event flashing is supported only in the Active Event List.

1. Change to the location of the supplied file.

```
cd /workshop/unit05
```

2. Examine the contents of the SQL file.

```
more prompttable.sql
```

```
-- Create a table for Flashtool and insert values
create table custom.SetFlash persistent
(FlashValue integer primary key, ShowValue varchar (20));
go
insert into custom.SetFlash (FlashValue, ShowValue) values (0, 'Set Flashing
OFF');
insert into custom.SetFlash (FlashValue, ShowValue) values (1, 'Set Flashing
ON');
go
```

The file creates a table that is called **SetFlash** in the **custom** database. The table contains an integer column, called **FlashValue**, and a character column called **ShowValue**. The table contains two records. One record contains a value of **0** for FlashValue and a value of **Set Flashing Off** for the ShowValue. The second record contains a value of **1** for FlashValue and a value of **Set Flashing On** for the ShowValue.

3. Create the table in the **NYC_AGG_P** ObjectServer.

```
nco_sql -server NYC_AGG_P -user root -password object00 < prompttable.sql
```

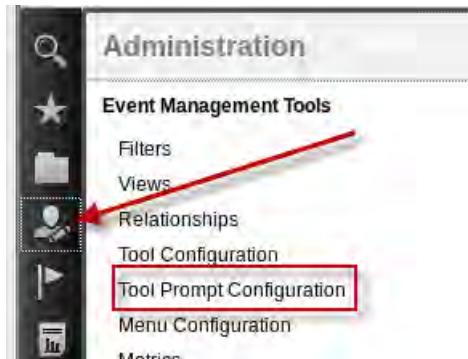
4. Create the table in the **NYC_AGG_B** ObjectServer.

```
nco_sql -server NYC_AGG_B -user root -password object00 < prompttable.sql
```



Important: You create a Web GUI tool that uses the prompt that is based on an ObjectServer table. Web GUI has access to both ObjectServers. Therefore, you must create the table that is used for the prompt in both ObjectServers.

5. Click the icon, and select **Tool Prompt Configuration**.



The list of prompts opens.

Copy an existing dynamic choice prompt.

6. Click **userassign**, and then click the icon to copy the prompt definition.

Name	Type
FlashPrompt	DynamicChoice
groupassign	DynamicChoice
hostname	String
priority	DynamicChoice
suppress	DynamicChoice
userassign	DynamicChoice

7. Enter **FlashPrompt** for the name.

8. Observe the value for SQL Query:

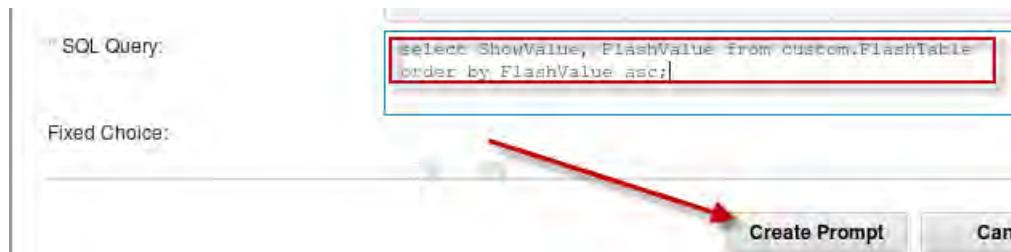
```
select Name, UID from master.names order by Name asc;
```

The userassign prompt is based on the **master.names** table. The columns in this table are **Name**, and **UID**. The first column in the select statement, **Name**, is what the user sees in the prompt window. When the user clicks one of the entries for Name in the prompt, the corresponding **UID** value is passed to the tool.

9. Change the value of SQL Query to the following text:

```
select ShowValue, FlashValue from custom.SetFlash order by FlashValue asc;
```

10. Click **Create Prompt**.



11. Click the X on the tab to close the prompt configuration page.

12. Click the tab to return to the Tool Configuration page.

Use a copy of an existing SQL tool to create your tool.

13. Click **acknowledge**, and then click the icon to create a copy of the tool. Change the tool name to **SetFlash**.



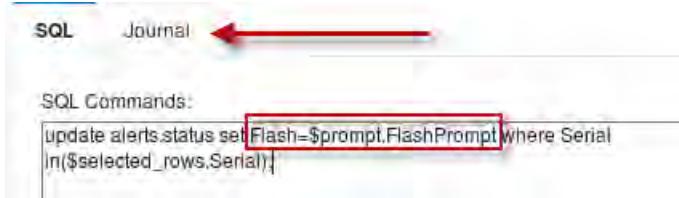
14. Observe the SQL command:

```
update alerts.status set Acknowledged=1 where Serial in($selected_rows.Serial);
```

15. Change the SQL command to the following text:

```
update alerts.status set Flash=$prompt.FlashPrompt where Serial  
in($selected_rows.Serial);
```

16. Click **Journal**.



17. Change the text for the journal message, and click **Save**.



18. Click **OK**.

19. Click the tab to return to the **Menu Configuration** page.

20. Click **alerts**, and then click **Modify**.

21. Click **SetFlash** and click the *right arrow* to add it to the menu. Click **Save**.

22. Click **OK**.

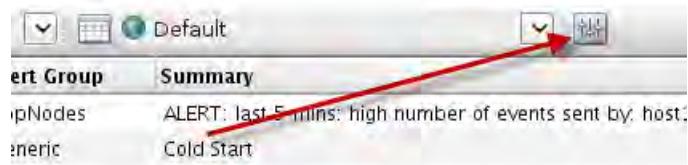
Testing the tool

Each user can enable, or disable the flashing capability in their Active Event List. The capability is disabled by default.

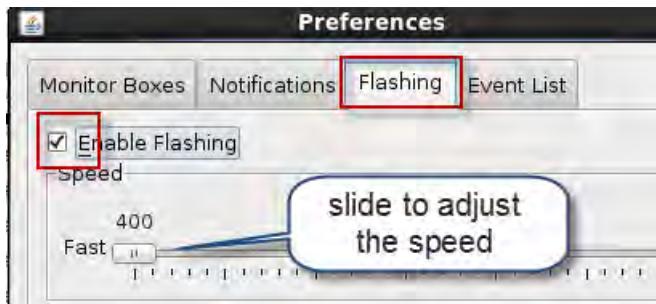
1. Click the icon, and select **Active Event List**.



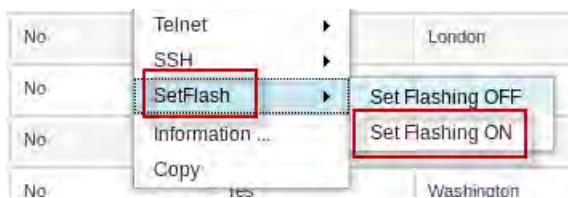
2. Click the icon to open the property editor.



3. Click **Flashing**, and select **Enable Flashing**. Click **Save**.



4. Right-click any event, and select **SetFlash > Set Flashing ON**.

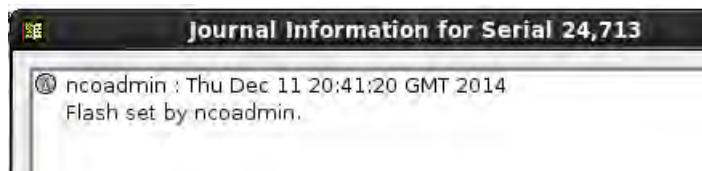


The event record flashes.

 Hint: Click some other event record to see the flashing better.

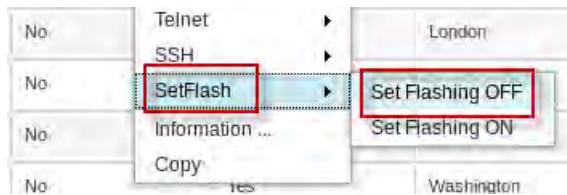
Yes	host1	TopNodes
Yes	127.0.0.1	Generic
Yes	host1.tivoli.edu	ProbeStatus
Yes	Washington	Systems
No	Moscow	Customs

5. Right-click the same event record, and select **Journal**.



6. Close the Journal window.

- Right-click the same event, and select **SetFlash > Set Flashing OFF**.



The event record no longer flashes.

- Close all open pages.

Exercise 3 Creating maps

In the following steps, you create a simple map.



Important: You perform the following exercises on **host2**.

- Click the icon, and select **Map Creation**.



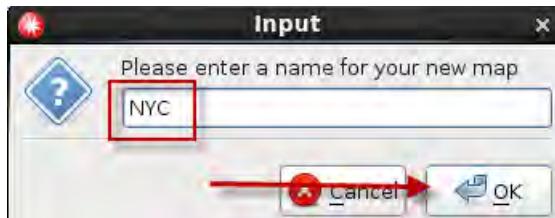
The list of existing maps opens.

- Wait for the map editor to initialize before proceeding.



The map editor uses a Java applet, and it might take a few moments to load.

3. Click **New**.
4. Enter **NYC** for the name, and click **OK**.



Important: A map name cannot contain spaces or special characters.

5. Click the arrow, and select **usa.png** for the background image.

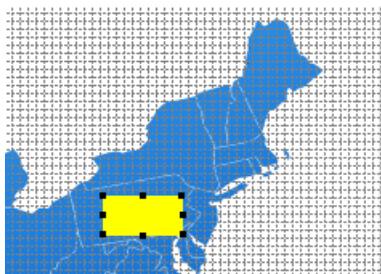


6. Drag the corners of the editor to increase the page size until the map fits in the view.
7. Click the *red square* icon on the bottom of the page to select it.



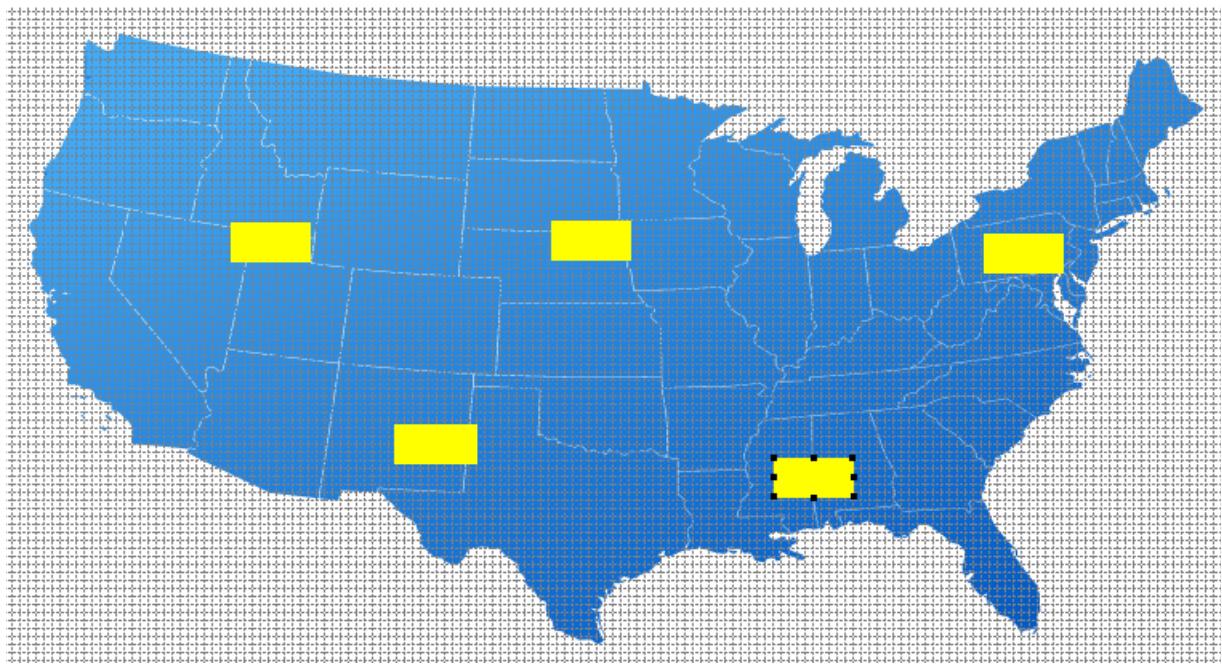
The *red square* represents an active icon. The *gray square* is an inactive icon.

8. Click anywhere on the map image.



A yellow rectangle appears on the map.

9. Place four more rectangles on the map in various locations. Drag the rectangles to change position.

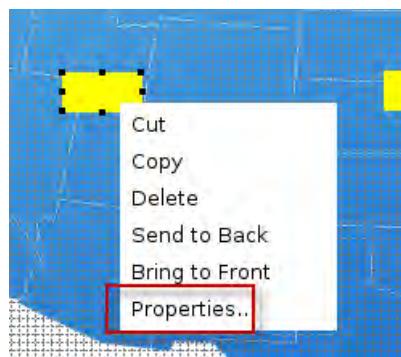


10. Click the arrow icon.

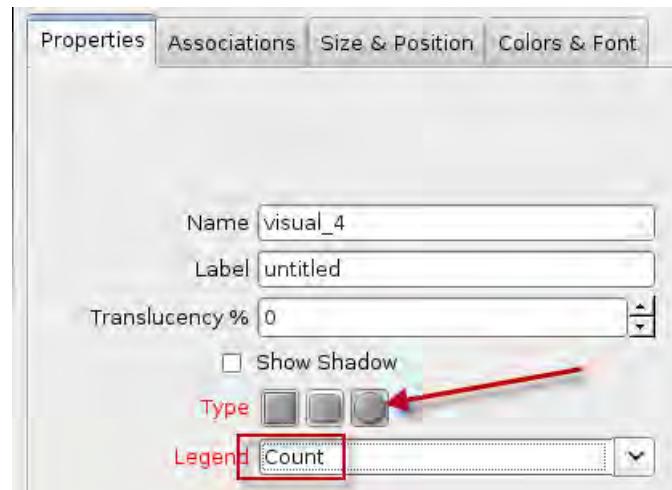


You click the *arrow* icon to clear the *red square*. If you do not select the *arrow*, you create a new rectangle every time you click the map.

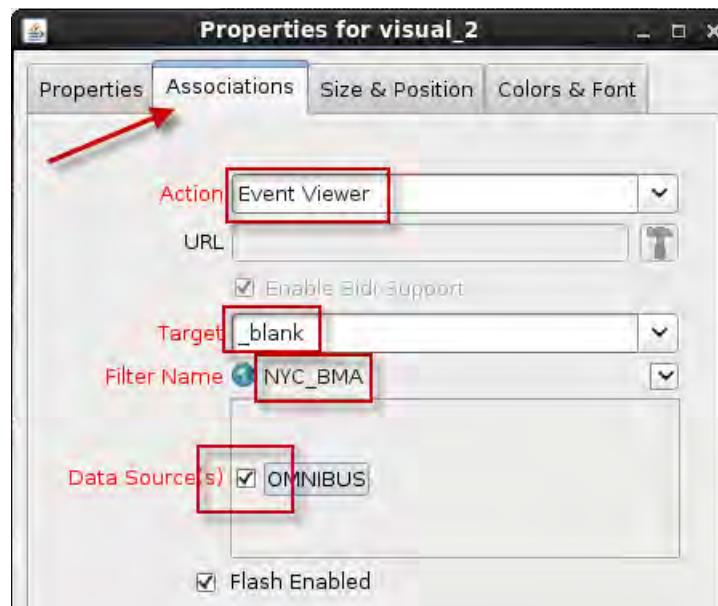
11. Right-click an icon, and select **Properties**.



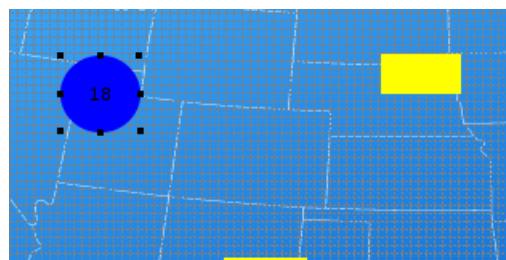
12. Select the *circle* icon, and select **Count** for the Legend.



13. Click **Associations**. Select **Event Viewer** for the action. Select **_blank** for the target. Select **NYC_BMA** for the filter. Select **OMNIBUS** for the data source. Click **OK**.



14. Drag the corner of the icon until it resembles a circle.



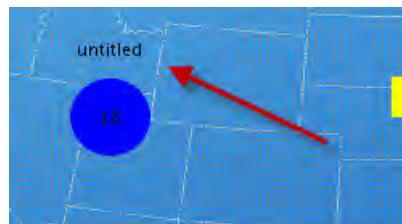
After the properties are configured, the icon turns blue.

15. Click the **A** icon on the bottom of the screen.



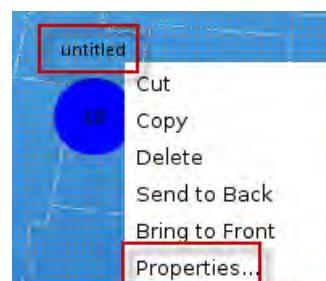
The **A** icon is used to place text on the map.

16. Click the map some place near the blue icon.

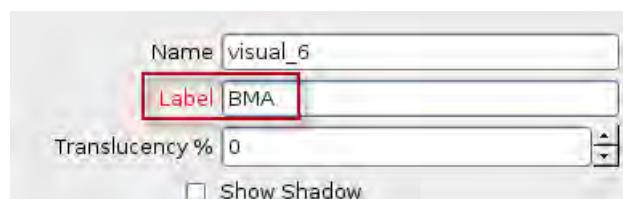


A small block of letters appears on the map.

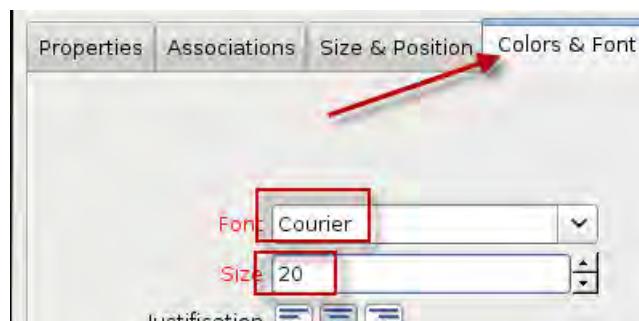
17. Right-click **untitled**, and select **Properties**.



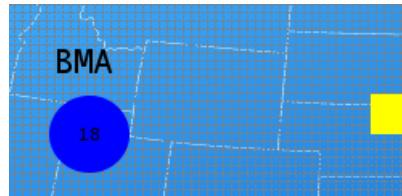
18. Enter **BMA** for the label.



19. Click **Colors & Font**. Select **Courier** for the font. Select **20** for the size. Click **OK**.



The text **BMA** appears on the map. The text box can be moved like other icons.



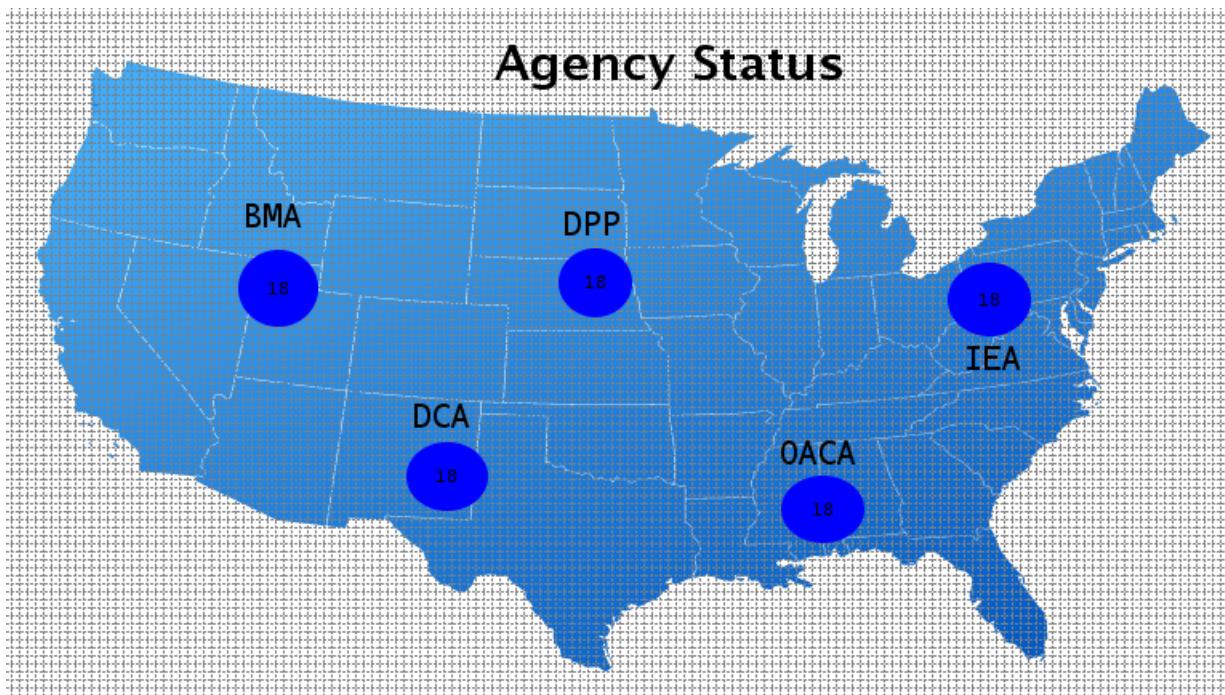
20. Repeat the previous steps to configure the properties for the remaining icons. Configure the icons with the following filters:

NYC_DCA
NYC_DPP
NYC IEA
NYC_OACA

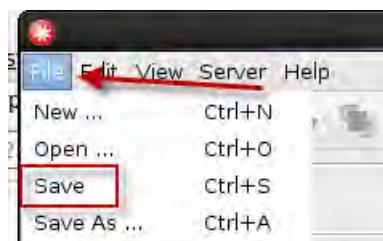
21. Use the text box to add a label for each icon.

22. Use the text box to add a title for the map.

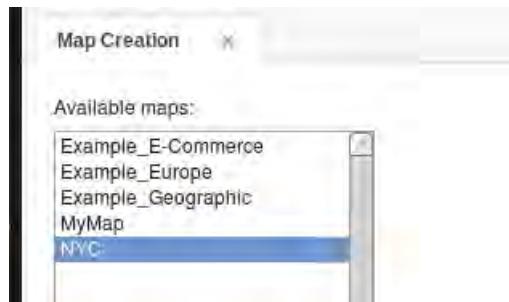
The complete map appears as follows:



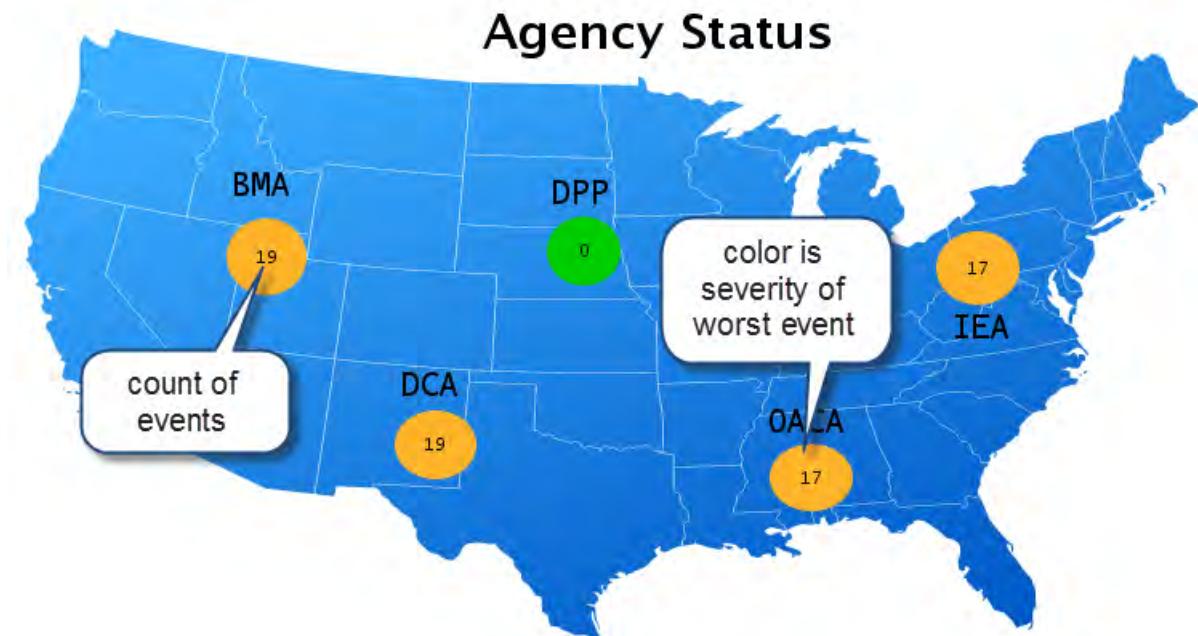
23. Click **File**, and select **Save**. Click **File**, and select **Exit**.



24. Click **NYC**, and then click **Preview**.



The map opens in a new window.



The number inside each circle is the count of events that meet the corresponding filter criteria.
The color of the circle is based on the severity of the worst event.

25. Click the circle for **BMA**.

A screenshot of a detailed event list titled "NYC_BMA". The interface includes a toolbar with icons for refresh, search, and filters, and a status bar showing counts for various event types: 0 red, 8 orange, 3 yellow, 5 blue, 0 purple, and 4 green.

Group	Count	Sev	Agency Id	Site ID	Last Occurrence	Node
All	20	⚠️	bma	bma002	12/29/14 7:55:45 PM	nsm123-b1.bma.gov
bma	2	⚠️	bma	bma002	12/29/14 8:31:08 PM	nsm123-b2.bma.gov
bma001	7	⚠️	bma	bma003	12/29/14 8:34:20 PM	nsm123-c3.bma.gov
bma002	11	⚠️	bma	bma003	12/29/14 8:28:53 PM	nsm123-c2.bma.gov
bma003	0	⚠️	bma	bma001	12/29/14 8:32:32 PM	3620-ce1-s.bma.gov

Annotations highlight the "Agency Id" column header and the "Site ID" column header.

A new Firefox window opens because you configured `_blank` for the *target*. The window contains the **Event Viewer** application because you configured the Event Viewer as the *action*. The Event Viewer uses the **NYC_BMA** filter. The grouping criteria is configured for **AgencyId** and **SiteId**, which is configured in the *view*.

26. Close the Event Viewer window.
27. Click each of the remaining icons, and verify the subsequent actions.
28. Close the map preview window.
29. Close the map creation page.

The last step in the map creation process is to add the map to a Dashboard Application Services page. You add the map to a page in the exercise on dashboards.

Exercise 4 Working with gauges

In the following exercise, you explore the existing gauges and create one of your own.



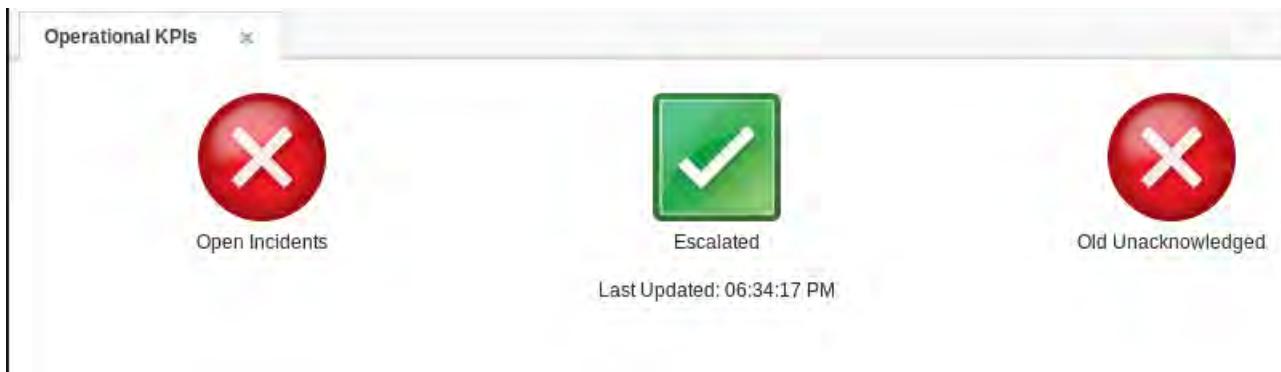
Important: You perform the following exercises on **host2**.

Examining an existing gauge

1. Click the icon, and select **Operational KPIs**.



The page opens.



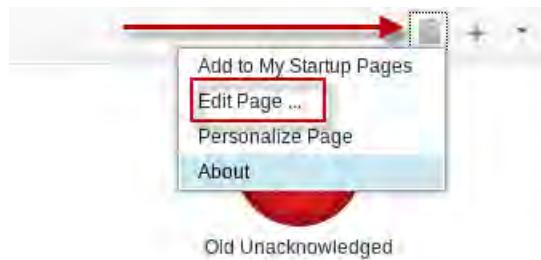
Each gauge on this page represents some characteristic of the events within the ObjectServer.

2. Hover over the first gauge.



A description for the gauge appears. This gauge represents the count of open event records.

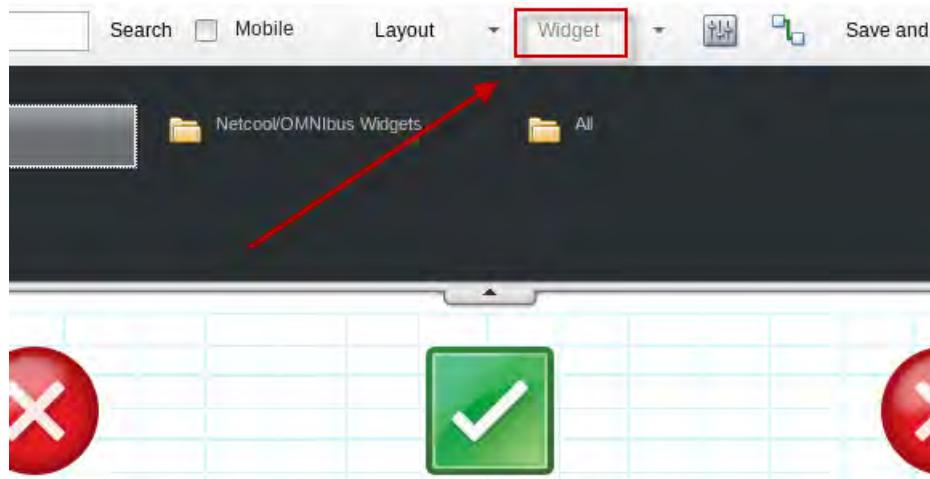
3. Hover over each gauge, and examine their description.
4. Click the icon, and select **Edit Page**.



Hint: Use **Edit Page** to change the page for all users. Use **Personalize Page** to change the page for just the ncoadmin user.

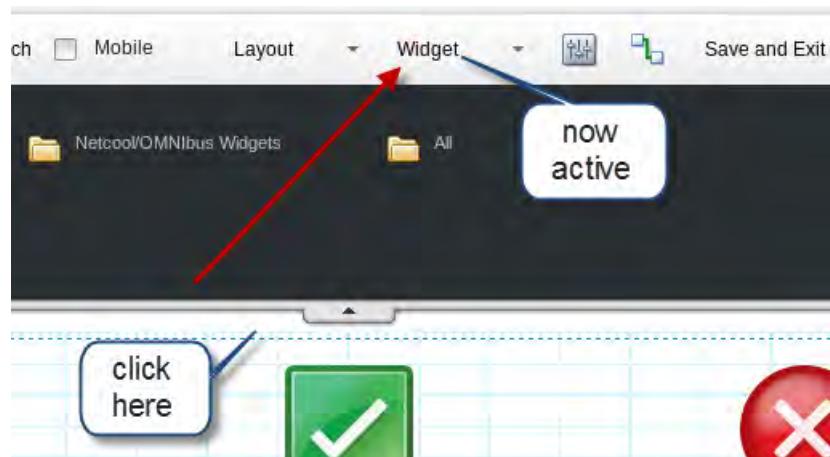
The page opens in the page editor.

5. Observe the **Widget** feature on the menu bar.



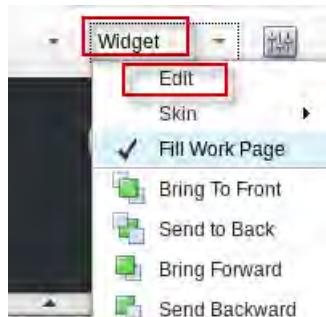
The Widget feature is gray and not accessible.

6. Click the top line of the grid. A *dotted outline* appears around the page, and the **Widget** feature turns active.



Hint: Place the mouse just below the top line in the grid. Slowly move the mouse up until the pointer changes from the *arrow* to a *hand*. The *hand* indicates the correct position.

7. Click **Widget**, and select **Edit**.



The page contains a single *Gauge* widget. The widget contains three gauges.

8. Observe the Refresh Rate.

Gauges

General Settings

Title : Gauges

Gauge size : 80

Refresh rate : **10 seconds**

Show last updated time

SQL query runs based on the refresh rate

Data Sources

Select	Name
<input checked="" type="checkbox"/>	OMNIBUS
<input type="checkbox"/>	OMNIBUS [host=host1.tivoli.edu, port=4100, type=PRIMARY]
<input type="checkbox"/>	OMNIBUS [host=host2.tivoli.edu, port=4100, type=BACKUP]

The Refresh Rate determines the frequency of the SQL query. Every gauge in the widget refreshes at the same rate. If you want to configure a gauge with a different rate, you must create another page.

9. Click the first gauge in the list, and observe the configuration.

Gauges Layout

* Type : Status Button

Preview :

* Metric : Open Incidents

Label : Open Incidents

Unit label : alarms

Description : Total number of open incidents. Current value of metric is {0}.

Click action : Send Event (using wires)

The name of the metric is **Open Incidents**. The label on the page is also **Open Incidents**. The description contains the text that appears when the user hovers over the gauge.

10. Select **HTML for mobile devices**, scroll to the bottom of the page, and click **Save**.

Mobile Devices Access

HTML for mobile devices

11. Repeat the previous steps to edit the widget properties again. Observe the **Mobile Devices Access**.

Mobile Devices Access

HTML for mobile devices

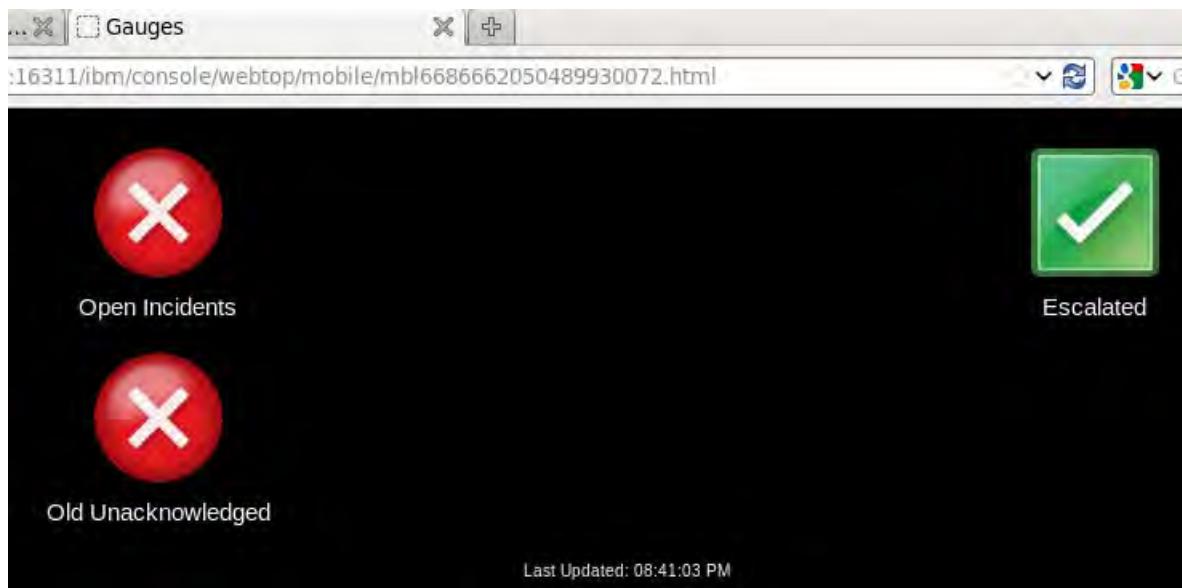
To access this Dashboard on a mobile device, use the following URL, or scan the QR Code using the QR Code reader on:

<https://host2.tivoli.edu:16311/bm/console/webtop/mobile/mbl6686662050489930072.html>



When you select the HTML feature, and save the widget properties, the access information is generated. You can copy the URL and email it to a user.

12. Click the URL.



A new Firefox window opens, and the mobile version of the page appears.

13. Click the **Open Incidents** gauge.

Event Details	Timestamp
host1 ALERT: last 5 mins: high number of events sent by host1: 882	1:43 PM
host1.tivoli.edu ALERT: syslog Probe (Conn ID: 2) sent high number of events: 884	1:43 PM
link4 Link Down on port	12/2/14
Moscow Machine has gone offline	12/2/14
Washington Machine has gone offline	12/2/14
London Machine has gone offline	12/2/14
Sydney Machine has gone offline	12/2/14
127.0.0.1 Cold Start	10/23/14
host1.tivoli.edu ALERT: last 5 mins: high number of events for class: Syslog Probe (200): 769	1:43 PM
host1.tivoli.edu	9:22 PM

A new Firefox window opens with the mobile version of the Event Viewer.



Important: When you clicked the URL, and opened the page with the mobile version of gauges, you did not enter a user name or password. You are not prompted for a user because you are already logged in to Web GUI with the Firefox browser. If you enter the URL in the browser of a mobile device, you need to enter a user name and password.

The mobile event list provides read-only access to event records from a mobile device. A user can click the *right* arrow, and drill down into event details. However, the user cannot run any tools.

14. Close the mobile event list page.
15. Close the mobile gauge page.
16. Scroll down in the property editor, and click **Cancel** to exit the widget property editor.
17. Click **Cancel** to exit the page editor.
18. Close the Operational KPIs page.

Creating metrics

1. Click the icon, and select Metrics.



The list of existing metrics opens.

2. Click **openincidents**, and click the *pencil* icon to open the metric for edit.

Name	Display Name	Description
oldunacknowledged	Old Unacknowledged	Total number of old unacknowledged events
openincidents	Open Incidents	Total number of open incidents
probeconnections	Probe	Number of probe connections

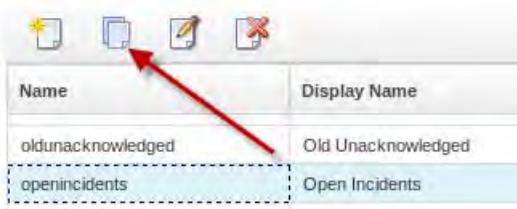
3. Examine the text of the SQL query.

```
select count(RowID) from alerts.status where Manager not like '^.*Watch$' and  
SuppressEscal < 4;
```

The query selects all events records that are not *ConnectWatch* events, and not *escalated* or in *maintenance*.

4. Click **Cancel**.

5. Click the icon to create a copy of the metric definition.



6. Change the name field to **nyc_bma**. Change the display name field to **BMA Incidents**. Change the description field.

The screenshot shows a configuration page for a metric. It has three input fields: 'Name' with value 'nyc_bma', 'Display Name' with value 'BMA Incidents', and 'Description' with value 'Total number of incidents for BMA. Current value of the metric is {0}'. The 'Name' and 'Description' fields are highlighted with red boxes.

7. Change the text for SQL query to the following value:

```
select count(RowID) from alerts.status where AgencyId='bma' and Severity>=3;
```

8. Click **Create Metric**.

9. Copy the **nyc_bma** metric, and create a metric definition for the remaining AgencyId values.

dca

dpp

iea

oaca

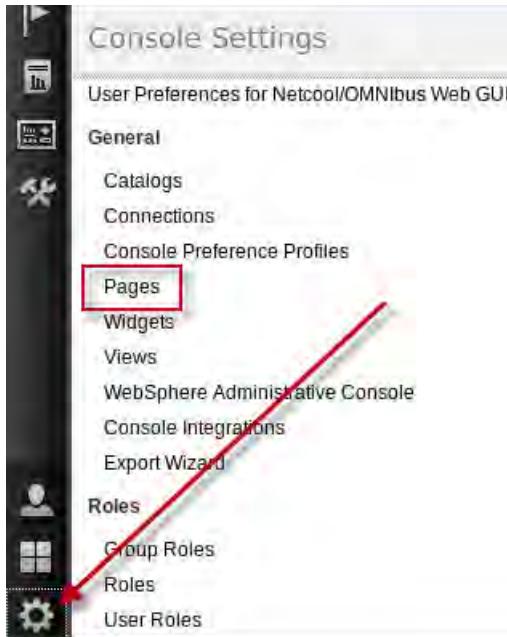
The metric definitions appear as follows:

Name	Display Name	Description
newrow	New Row	Total time taken to process the 'new_row' trigger. Current value of the metric is {0}.
nyc_bma	BMA Incidents	Total number of incidents for BMA. Current value of the metric is {0}.
nyc_dca	DCA Incidents	Total number of incidents for DCA. Current value of the metric is {0}.
nyc_dpp	DPP Incidents	Total number of incidents for DPP. Current value of the metric is {0}.
nyc_iea	IEA Incidents	Total number of incidents for IEA. Current value of the metric is {0}.
nyc_oaca	OACA Incidents	Total number of incidents for OACA. Current value of the metric is {0}.

10. Close the metric definition page.

Creating gauges

1. Click the icon, and select **Pages**.



The list of existing pages opens.

2. Click **New Page**.
3. Enter **NYC Status** for the page name. Select **Proportional** for the layout type. Expand **Optional setting**.

* Page name:
NYC Status

* Page location:
console/Default/

Location...

Page Layout:

Proportional Place and overlay widgets anywhere that will scale on work page.

Freeform - Place and overlay widgets anywhere on work page.

Fluid - Fix the widgets on the page. Great for mobile.

▶ Optional setting

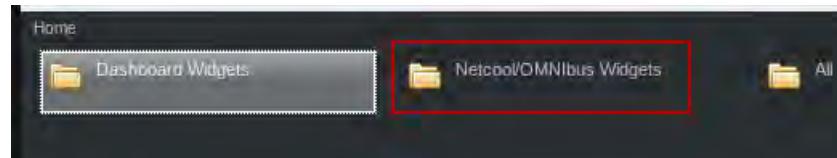
4. Click **Add**.
5. Select **All authenticated portal users**, and click **Add**.

Select	Role Name
<input type="checkbox"/>	administrator
<input checked="" type="checkbox"/>	all authenticated portal users

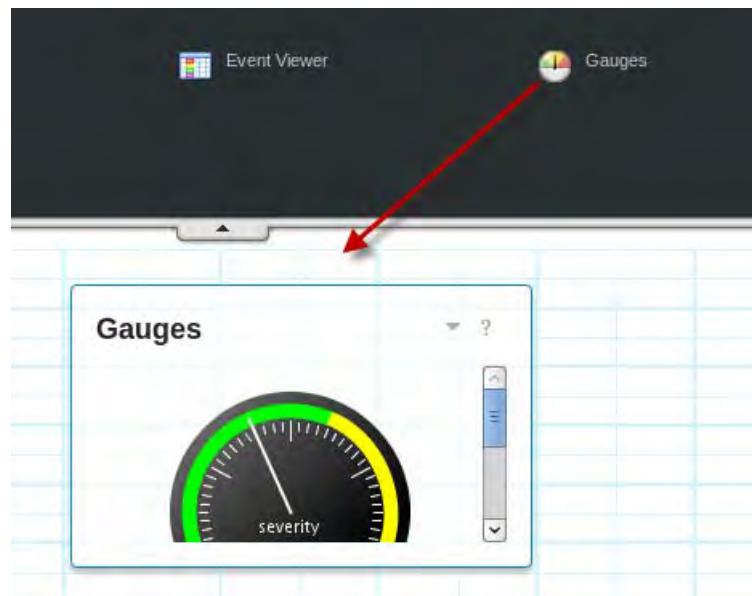
6. Click **OK** to create the page.

The page editor opens.

7. Click **Netcool/OMNIbus Widgets** to expand the folder.



8. Drag the **Gauges** widget to the page layout area.



9. Click the arrow and select **Edit**.



10. Change the title to **Agency Status**. Change gauge size to **50**. Change the refresh rate to **30**.

General Settings

Title:	Agency Status
Gauge size:	50
Refresh rate:	30 seconds
<input checked="" type="checkbox"/> Show last updated time	

11. Click the existing gauge to select it. Select **BMA Incidents** for the metric.

Gauges Layout

* Type : Dial

Preview :

* Metric : **BMA Incidents**

Label : BMA Incidents

Unit label : events

Description : Total number of incidents for BMA. Current value of the metric is {0}.

The values for label, units, and description are populated automatically from the metric definition.

12. Scroll down, and click **Apply Changes**.

Apply Changes



Important: You click Apply Changes to replace the existing gauge definition with your changes.

The BMA gauge replaces the existing gauge.

Gauges Layout

* Type : Dial

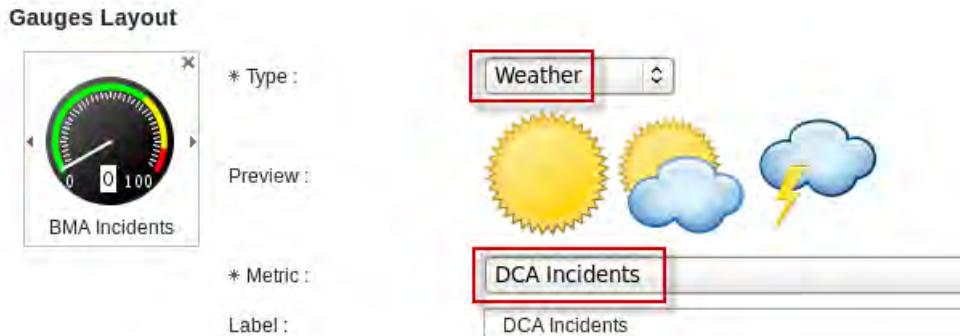
Preview :

* Metric : **BMA Incidents**

Label : BMA Incidents

Unit label : events

13. Select **Weather** as type. Select **DCA Incidents** for the metric.



14. Scroll down, and click **Add Gauge**.

 Apply Changes Add Gauge



Important: You click Add Gauge to add the new definition to the page.

The page now contains two gauges.



15. Repeat the previous step to add a gauge for each of the remaining agencies:

dpp
iea
oaca

Select a different gauge type for each agency.



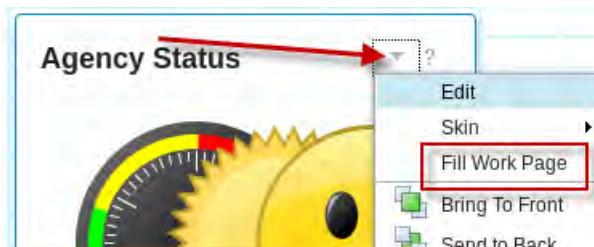
Note: You typically do not use a different type for each gauge. You use different types in this exercise so you can see how each type appears in the subsequent display.

The completed page appears as follows:



16. Scroll to the bottom of the page, and click **Save**.

17. Click the arrow, and select Fill Work Page.



18. Click **Save and Exit** to close the page editor.

The page appears with the agency gauges.

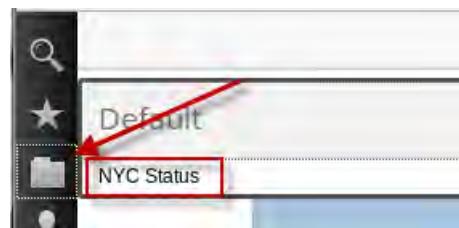


19. Drag the edge of the Firefox browser to change the size of the display area.



The gauges move within the page because you used the *proportional* layout. If you use the freeform layout, the gauges remain in their location when you resize the display area. If the display area is too small, and the gauges do not fit, scroll bars appear.

20. Close the NYC Status Page.
21. Click the icon, and select **NYC Status**.



The Default folder is where pages are stored when you do not change the default location.

22. Close the **NYC Status** page.

Exercise 5 Creating a dashboard

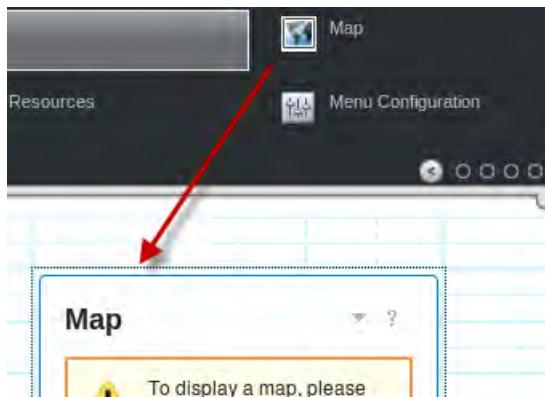
In the following exercise, you create a simple dashboard page.



Important: You perform the following exercises on **host2**.

1. Click the tab to return to the **Pages** page.
2. Click **New Page**.
3. Enter **NYC Dashboard** for the page name. Select **Freeform**. Expand **Optional setting**.
4. Add **All authenticated portal users**, and click **OK**.

5. Click **All** to expand the folder.
6. Drag the **Map** widget to the page layout area.



7. Edit the Map widget properties.
8. Select **NYC**. Enter **60** for the refresh rate. Click **Save**.

Map - Edit Preferences

General Settings

Map name : (The input field is highlighted with a red box.)

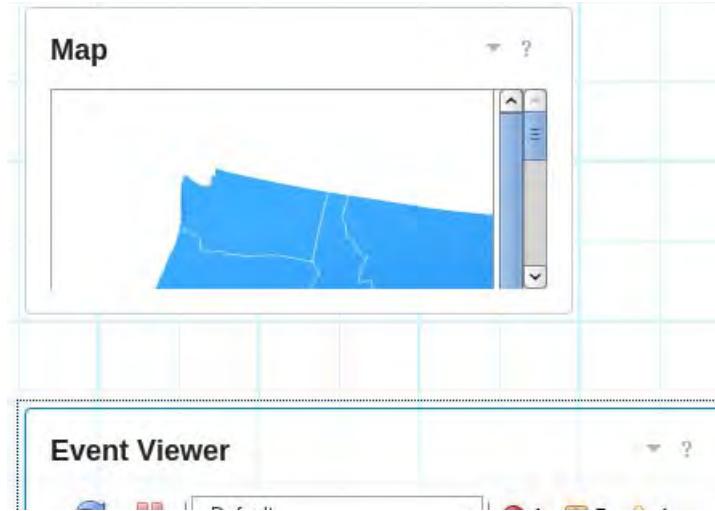
Sound URL :

Refresh rate (in seconds) : (The input field is highlighted with a red box.)

Enable hover help for active objects :

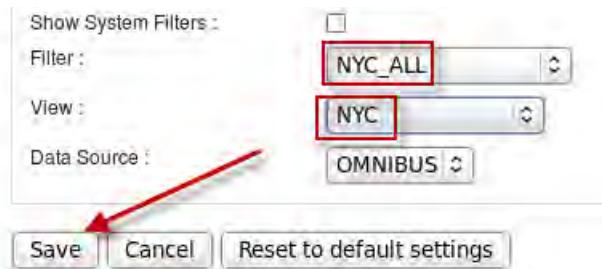
Note: Do not resize the map widget yet.

9. Drag the **Event Viewer** widget to the page layout area below the map.



10. Edit the Event Viewer widget properties.

11. Select **NYC_ALL** for the filter. Select **NYC** for the view. Click **Save**.

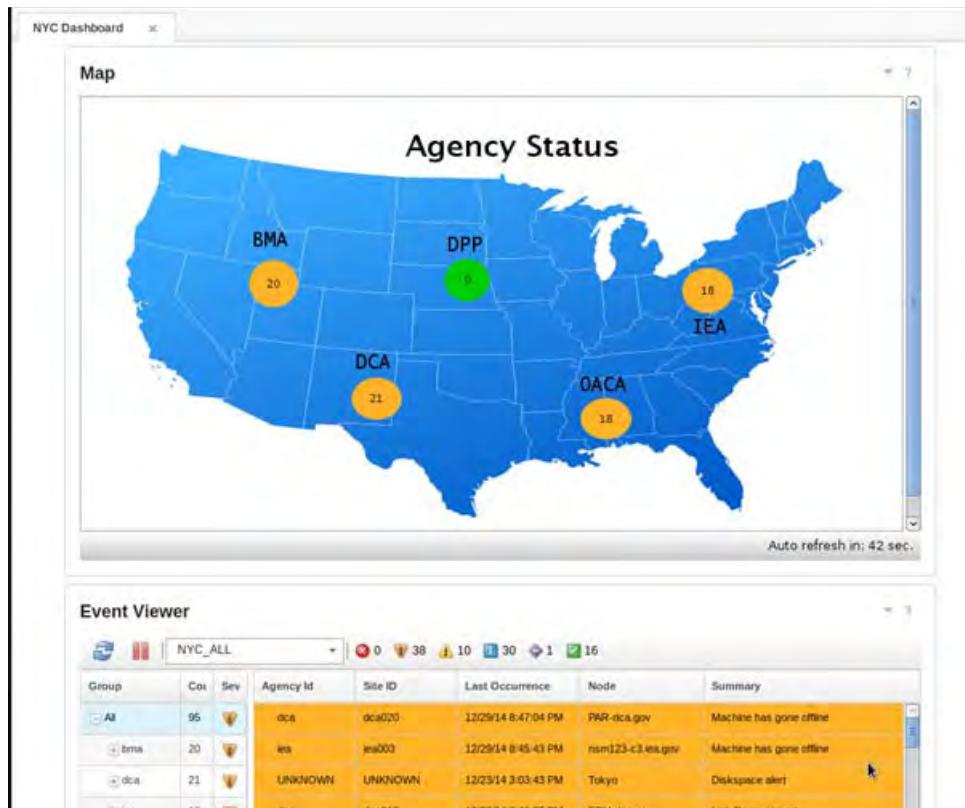


12. Expand the height and width of the Event Viewer.

13. Expand the height and width of the map.

14. Click **Save and Exit**.

The dashboard page opens.



15. Log out of Dashboard Application Services Hub.

16. Close the Firefox browser.

Exercise 6 Working with Web GUI administration API

In the following exercise, you configure the Web GUI administration API client, and explore some of the features of the client.



Important: You perform the following exercises on **host2**.

Configuring the client

You must modify a property file before the client is used for the first time.

1. Open a Terminal window if necessary.
2. Change to the target directory.

```
cd /opt/IBM/netcool/omnibus_webgui/waapi/etc
```

3. Create a copy of the original file before changes.

```
cp waapi.init waapi.init.orig
```
4. Edit the file with the gedit utility.

```
gedit waapi.init
```

5. Locate the following section:

```
waapi.host:localhost
waapi.port:16310
waapi.secureport:16311
waapi.contextpath:/ibm/console/webtop
waapi.user:root
waapi.password:
waapi.password.encryption:none
waapi.file:
waapi.timeoutsecs: 600
```

6. Change the following lines as shown:

```
waapi.user:ncoadmin
waapi.password:object00
```

7. Save the changes, and exit the gedit utility.

Working with the client

The following steps demonstrate a few of the sample XML files that are included with Web GUI.

1. Change the target directory.

```
cd /opt/IBM/netcool/omnibus_webgui/waapi/etc/samples
```

2. Examine the list of sample files.

```
ls
```

The sample files include several files that can be used to list various Web GUI objects, such as filters, views, and others.

3. Change the target directory.

```
cd /opt/IBM/netcool/omnibus_webgui/waapi/bin
```

4. Run the utility to list the configured maps.

```
./runwaapi -file ../etc/samples/list_map.xml
```

```
*****
```

```
WAAPIClient: Request sent to server on
```

```
http://localhost:16310/ibm/console/webtop/...
```

```
Mon Dec 15 18:53:30 UTC 2014
```

```
Maps hosted on the server
```

```
*****
```

```
Example_E-Commerce
```

```
Example_Europe
```

```
Example_Geographic
```

```
NYC
```

```
*****
```

```
WAAPIClient: 1 method was fully executed.
```

The list includes the NYC map.

5. Run the utility to list the configured filters.

```
./runwaapi -file ../etc/samples/list_filter.xml
*****
WAAPIClient: Request sent to server on
http://localhost:16310/ibm/console/webtop/...
Mon Dec 15 19:00:10 UTC 2014
```

Filters hosted on the server

```
*****
```

Global filters:

- AllEvents
- Default
- Escalated
- Information
- InMaintenance
- Last10Mins
- NetcoolStatus
- NYC_ALL
- NYC_BMA
- NYC_DCA
- NYC_DPP
- NYC IEA
- NYC_OACA
- TaskList
- Unacknowledged

The list includes the filters based on AgencyId created previously.

6. Run the utility to list the configured views.

```
./runwaapi -file ../etc/samples/list_view.xml
```

```
*****
```

```
WAAPIClient: Request sent to server on
http://localhost:16310/ibm/console/webtop/...
Mon Dec 15 19:02:14 UTC 2014
```

Views hosted on the server

```
*****
```

Global Views:

- Default
- NYC

The list includes the NYC view that is created previously.

If time allows, feel free to explore other sample XML files.



6 User administration exercises

In this unit, you learn how to configure user access to event information. You configure a user for access to event records, add a restriction to limit the events, and then configure a default startup page for the user.

Exercise 1 Configuring users for event access

In this exercise, you create two groups. You add two existing users to the groups. Then, you add Web GUI roles to the groups, which enable access to event records.



Note: You can use either image for this exercise.

Creating a group

Create a group for the NYC operators and a group for NYC users.

1. Open a Firefox browser if necessary.
2. Log in as the **smadmin** user with password **object00**.
3. Click the icon, and select **WebSphere Administrative Console**.



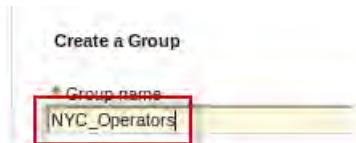
4. Click **Launch WebSphere administrative console**.

5. Expand **Users and Groups**. Click **Manage Groups**.



6. Click **Create**.

7. Enter **NYC_Operators** for the name, and click **Create**.



8. Click **Close**.

The NYC_Operators group is created.

9. Click **Create**.

10. Enter **NYC_End_Users** for the name, and click **Create**.

11. Click **Close**.

The NYC_End_Users group is created.

12. Click **Manage Users**.



13. Click **abraman** to open the user record for edit.

Select	User ID	First name	Last name	E-mail	
<input type="checkbox"/>	abraman	Ariana	Braman	abraman@ibm.com	
<input type="checkbox"/>	adurling	Adeline	Durling	adurling@ibm.com	
<input type="checkbox"/>	bwinebarger	Bart	Winebarger	bwinebarger@ibm.com	
<input type="checkbox"/>	dselan	Dick	Selan	dselan@ibm.com	

14. Select the **Groups** tab, and click **Add**.



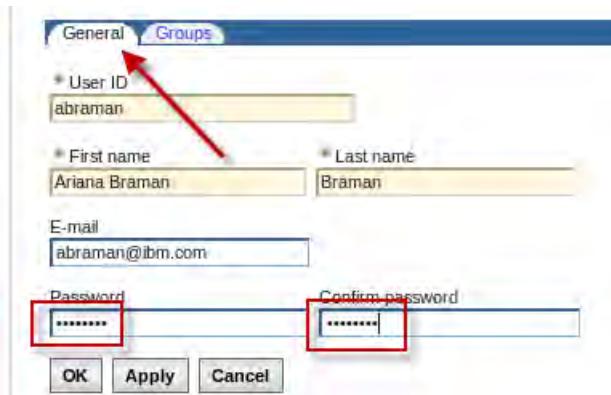
15. Click **Search** to create the list of available groups.

16. Click **NYC_Operators** to select the group, and click **Add** to add the user to the group.

17. Click **Close**, and the **NYC_Operators** group is listed.



18. Select the **General** tab. Enter **object00** for the password, and click **OK** to update the user.



19. Click **dselan** to open the user record for edit.

29 users matched the search criteria.				
	Select	User ID	First name	Last name
<input type="checkbox"/>	abraman	Ariana Braman	Braman	abi
<input type="checkbox"/>	adurling	Adeline Durling	Durling	adi
<input type="checkbox"/>	bwinebarger	Bart Winebarger	Winebarger	bw
<input type="checkbox"/>	dselan	Dick Selan	Selan	dse

20. Select the **Groups** tab, and click **Add**.

21. Click **Search** to create the list of available groups.
22. Click **NYC_End_Users** to select the group, and click **Add** to add the user to the group.
23. Click **Close**, and the **NYC_End_Users** group is listed.
24. Select the **General** tab. Enter **object00** for the password, and click **OK** to update the user.



25. Log out of WebSphere administrative console.



Hint: The Logout button is on the menu bar, all the way to the right.

26. Close the Firefox tab for WebSphere administrative console.
27. Log out of Dashboard Application Services Hub as the smadmin user.

The NYC_End_Users group contains the dselan user, and the NYC_Operators group contains the abraman user.

Adding roles to groups

There are no roles that are assigned to the groups, so the users do not have access to any Dashboard Application Services Hub features. Ariana Braman is an operator in the NYC network operations center. Ariana requires read/write access to event records. Dick Selan is an user of the NYC operation. Dick requires read-only access to event records. In the following steps, you assign roles to each group to grant the required access.

1. Log in to Dashboard Application Services Hub as the **ncoadmin** user with password **object00**.
2. Click the icon, and select **Group Roles**.





Hint: Press the **F11** key to place the browser in full screen mode.

3. Enter **NYC*** for Group ID, and click **Search**.

The screenshot shows a search interface with a 'Group ID:' input field containing 'NYC*' and a red box highlighting it. Below the input field is a 'Number of results to display' dropdown set to '20'. A 'Search' button is located at the bottom right of the search area.

The two groups are listed.

4. Click **NYC_End_Users**.

Group Name	Roles
NYC_End_Users	
NYC_Operators	

5. Select **ncw_user** and **netcool_ro**. Click **Save**.



The roles are added to the **NYC_End_Users** group.

6. Click **NYC_Operators**.

Group Name	Roles
NYC_End_Users	ncw_user, netcool_ro
NYC_Operators	

7. Select **ncw_user**, and **netcool_rw**. Click **Save**.



The roles are added to the group.



8. Close the Group Roles tab.

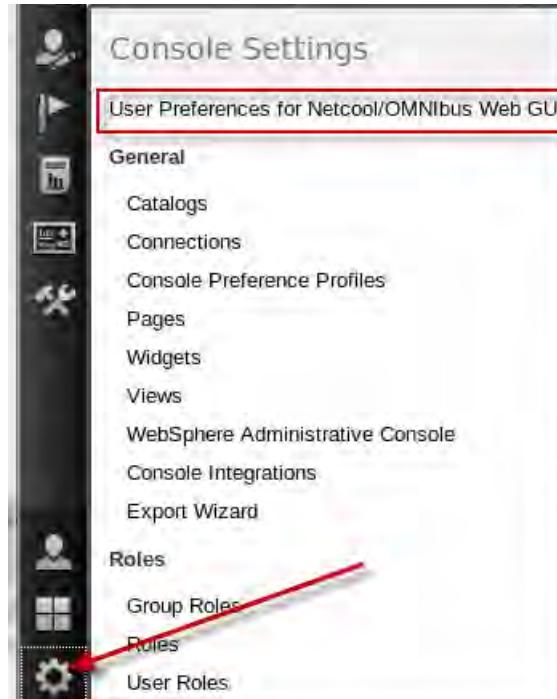
Limiting access to event records

The NYC ObjectServers contain events from the entire infrastructure. The two users are interested in only the events that are related to the managed agencies. In a previous unit, you configured a

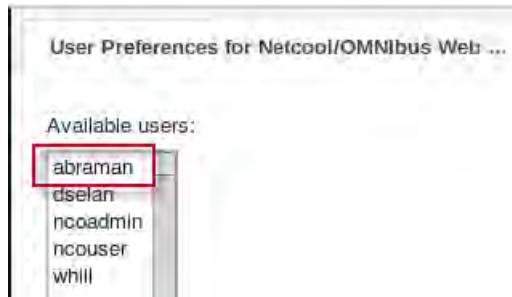
Exercise 1 Configuring users for event access

probe lookup table to populate the AgencyId column. You can use that column to limit access to event records.

1. Click the icon, and select **User Preferences for Netcool/OMNIbus Web GUI**.



2. Click **abraman** to select the entry, and click **Modify**.



3. Enter the following text for the user filter.

`AgencyId <> '' and AgencyId <> 'UNKNOWN'`

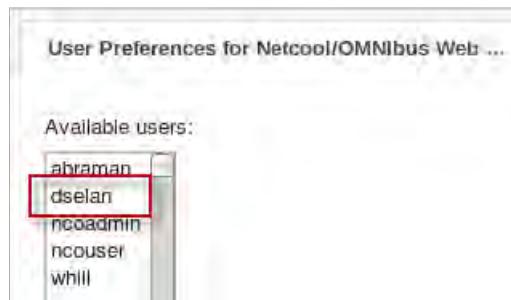
User name:	abraman
User filter:	'Id <>" and AgencyId <> 'UNKNOWN'.'
User's home-page:	/



Important: The complete string does not appear in the screen capture.

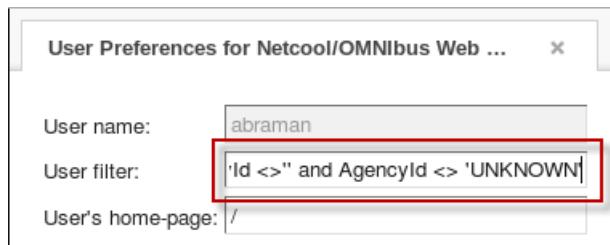
4. Scroll to the bottom of the page, and click **Save**.

5. Click **dselan** to select the entry, and click **Modify**.



6. Enter the following text for the user filter.

AgencyId <> '' and AgencyId <> 'UNKNOWN'



Important: The complete string does not appear in the screen capture.

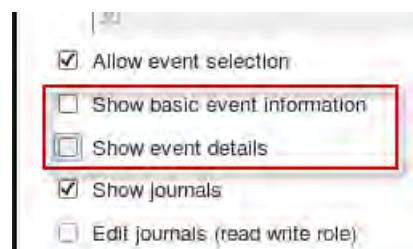
Dick Selan is a normal user, and you want to restrict access to other event features. Dick requires access to journal records, but does not require access to details. Also, you need to ensure that Dick sees only the event columns that you configure.

7. Scroll down, and remove the option for **Allow filter and view selection**.



Removing this feature prevents the user from selecting another filter or view in their desktop.

8. Scroll down, and remove the options for **Show basic event information** and **Show event details**.



9. Scroll to the bottom of the page, and click **Save**.

10. Close the user preferences tab.

Leave the browser session as is. You use it again shortly.

Exercise 2 Configuring default startup pages

In this exercise, you configure default startup pages for the NYC users.



Note: You can use either image for this exercise.

The following list is a summary of the steps that are required to configure default startup pages.

- Create a role
- Create a startup page, and assign access for the role
- Create a view, and assign the startup page to the view
- Create a console preference profile, and assign the view to the profile
- Assign the role to the profile
- Assign the role to the user or group

Creating a role

The role is a key component of default startup pages. It controls access to pages, and associates pages to users. You want to configure the dselan user for access to only the defined startup pages. You want to configure the abraman user for access to the startup pages, and other pages. You must create two roles to implement this behavior.

1. Click the icon, and select **Roles**.



2. Click **New**.

3. Enter **NYC** for the name, and click **Save**.



The role name is just text.

4. Repeat these steps and create the **NYC_Restricted** role.
5. Close the Roles tab.

Assigning the role to a page

You created custom pages in a previous unit. The pages are configured for access by all Dashboard Application Services Hub users. In the following steps, you modify the pages, and change the access to just the users with the NYC or NYC_Restricted role.

1. Click the icon, and select **Pages**.



2. Expand the **Default** folder, and click **NYC Status**.

A screenshot of a user interface showing a list of items under the 'Default' folder. The items are listed in a table with columns 'Select' and 'Name'. The 'Name' column contains 'Web Widget', 'NYC Status' (which is highlighted with a red box), and 'NYC Dashboard'. A red arrow points to the 'NYC Status' item.

Select	Name
	Web Widget
<input type="checkbox"/>	NYC Status
	NYC Dashboard

3. Expand **Roles with Access to This Page**.

A screenshot of a user interface showing the 'Roles with Access to This Page' section. It lists two roles: 'all authenticated portal users' and 'iscadmins'. The 'all authenticated portal users' role is highlighted with a red box. A red arrow points to the 'all authenticated portal users' entry.

Roles with Access to This Page: 2

This section lists the roles that have access to this page.

Add... Remove

Select	Role Name
<input type="checkbox"/>	all authenticated portal users
<input type="checkbox"/>	iscadmins

4. Select **all authenticated portal users**, and click **Remove**.

A screenshot of a user interface showing the 'Roles with Access to This Page' section. The 'all authenticated portal users' role is selected (indicated by a checked checkbox) and highlighted with a red box. A red arrow points to the 'Remove' button.

Add... Remove

Select	Role Name
<input checked="" type="checkbox"/>	all authenticated portal users
<input type="checkbox"/>	iscadmins

5. Click **Add**, select the **NYC** role and the **NYC_Restricted** role. Click **Add**.

A screenshot of a user interface showing the 'Roles with Access to This Page' section. The 'NYC' and 'NYC_Restricted' roles are selected (indicated by checked checkboxes) and highlighted with red boxes. A red arrow points to the 'NYC' role entry.

Add... Remove

Select	Role Name
<input type="checkbox"/>	iscadmins
<input checked="" type="checkbox"/>	NYC
<input checked="" type="checkbox"/>	NYC_Restricted

6. Click **Save** to save the page changes.

7. Repeat these steps for the **NYC Dashboard** page.



8. Click **Save** to save the page changes.

9. Close the Pages tab.

The pages are configured for access by only users with the NYC role or the NYC_Restricted role.

Creating a view



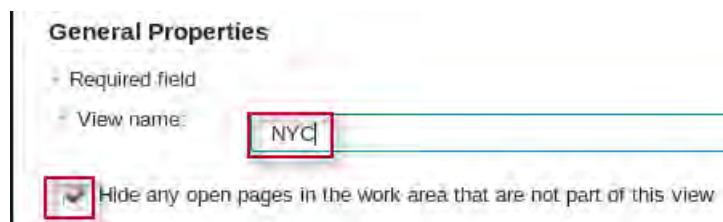
Important: You create a Dashboard Application Services Hub view, which is not the same as a Web GUI event view.

1. Click the icon, and select **Views**.



2. Click **New**.

3. Enter **NYC** for the name, and select **hide any open pages**.



4. Expand Roles with Access to This View, and click Add.

Roles with Access to This View: 1

This section lists the roles with access to this view.

Add... Remove Select Role Name

iscadmins

5. Select the **NYC** role, and the **NYC_Restricted** role. Click **Add**.

Add... Remove

Select Role Name

iscadmins

NYC

NYC_Restricted

6. Expand Pages in This View, and click Add.

Pages in This View: 0

This section lists the pages that are part of this view. To Default, WARNING: Not all content supports rendering

Add... Remove

Select Page Name

None

7. Expand the Default folder. Select **NYC Dashboard** and **NYC Status**. Click **Add**.

Select	Page Name	Type	Unique Name
<input checked="" type="checkbox"/>	Default	System	com.ibm.sysmgmt
<input checked="" type="checkbox"/>	NYC Status	Custom	com.ibm.isclite.adm
<input checked="" type="checkbox"/>	NYC Dashboard	Custom	com.ibm.isclite.adm

8. Select Launch for both pages. Select Default for the NYC Status page.

Select	Page Name	Launch	Default
<input type="checkbox"/>	Default		
<input type="checkbox"/>	NYC Status	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	NYC Dashboard	<input checked="" type="checkbox"/>	<input type="radio"/>

When Launch is selected, the page opens when the user logs in. The page that is selected as Default is *in focus* when the user logs in.

9. Scroll to the bottom of the page, and click **Save**.

10. Close the Views tab.

Creating a console preference profile

The console preference profile is where access to other pages is configured. To implement the correct behavior, you must create two console preference profiles.

1. Click the icon, and select **Console Preference Profiles**.



2. Click **New**.

The first profile is for unrestricted access.

3. Enter **NYC** for the name. Leave the default theme.

The screenshot shows the 'General Properties' dialog. Under 'Preference profile name', the value 'NYC' is entered and highlighted with a red box. Below it, 'Preference profile unique name' is set to '264441785'. The 'Theme' dropdown is set to 'IBM Design Signature'.

4. Leave the default settings for console view options.

The screenshot shows the 'Console view options' dialog. The 'Show view selector' radio button is selected and highlighted with a red box. Below it, the 'Hide view selector' radio button is unselected. A note says: 'Select the view options the user has access to in the View drop down menu in the banner.' Under 'Required view (at least one required view must be selected)', the 'All tasks' checkbox is checked and highlighted with a red box, while 'System and custom views' is also checked.

5. Expand Roles Using This Preference Profile, and click Add.

The screenshot shows the 'Roles Using This Preference Profile' section. A red arrow points to the 'Add...' button, which is highlighted with a red box. Other buttons visible are 'Remove' and 'Filter'.

6. Select the **NYC** role, and click **Add**.

The screenshot shows the 'Add Role' dialog. The 'NYC' role is selected and highlighted with a red box. A red arrow points to the 'Add' button, which is highlighted with a red box. Other buttons are 'Cancel' and 'Remove'.

7. Change the default view to **NYC**.

The screenshot shows the 'Default console view' configuration dialog. In the 'Default view:' dropdown, the value 'NYC' is selected and highlighted with a red box.



Important: By changing the default view to NYC, the default startup pages load when the user logs in.

8. Scroll to the bottom of the page, and click **Save** to create the profile.

9. Click **New**.

The second profile is for restricted access.

10. Enter **NYC_Restricted** for the name. Select **Tivoli Dark** for the theme.

General Properties

Required field

Preference profile name: **NYC_Restricted**

Preference profile unique name: **_96518594**

Theme: **Tivoli Dark**



Important: The theme does not control access to pages. You select the other theme just to demonstrate the appearance.

11. Change the default settings, and select **Hide view selector**. Remove the option for **All tasks**.

Console view options

Show view selector

Hide view selector

Select the view options the user has access to in the View drop down menu in the banner:

Required view (at least one required view must be selected)

All tasks

System and custom views



Important: The console view options control access to other pages. The settings that are shown here restrict this profile to just the pages that are associated with the **NYC_Restricted role**.

12. Expand **Roles Using This Preference Profile**, and click **Add...**.

▼ Roles Using This Preference Profile: 0

This section lists the roles that have access to this preference profile

Add... Remove Filter

13. Select the **NYC_Restricted** role, and click **Add**.

This section lists the roles that have access to this preference profile. If multiple views are selected, then a default view is chosen.

Select	Role Name
<input type="checkbox"/>	NYC_Restricted

Default Console view: **NYC**



Important: The NYC_Restricted profile is limited to a single view.

14. Scroll to the bottom of the page, and click **Save** to create the profile.

Select	Profile Name
<input type="checkbox"/>	NYC
<input type="checkbox"/>	NYC_Restricted

15. Close the console preferences page.

Modifying group roles

In the previous exercise in this unit, you created two groups. When you created the groups, you added roles that granted access to Web GUI features. In this exercise, you created two roles. You

used those roles to configure access to startup pages. You must add those roles to the corresponding groups to complete the user configuration.

1. Click the icon, and select **Group Roles**.



2. Enter NYC*, and click Search.
3. Click **NYC_End_Users**.
4. Select the **NYC_Restricted** role, and click **Save**.

Group Name	Roles	Uni
NYC_End_Users	NYC_Restricted, ncw_user, netcool_ro	cn=
NYC_Operators	ncw_user, netcool_rw	cn=

The NYC_End_Users group contains the NYC_Restricted role.

5. Click **NYC_Operators**, select the **NYC** role, and click **Save**.

Group Name	Roles	Uni
NYC_End_Users	NYC_Restricted, ncw_user, netcool_ro	cn=
NYC_Operators	ncw_user, NYC, netcool_rw	cn=

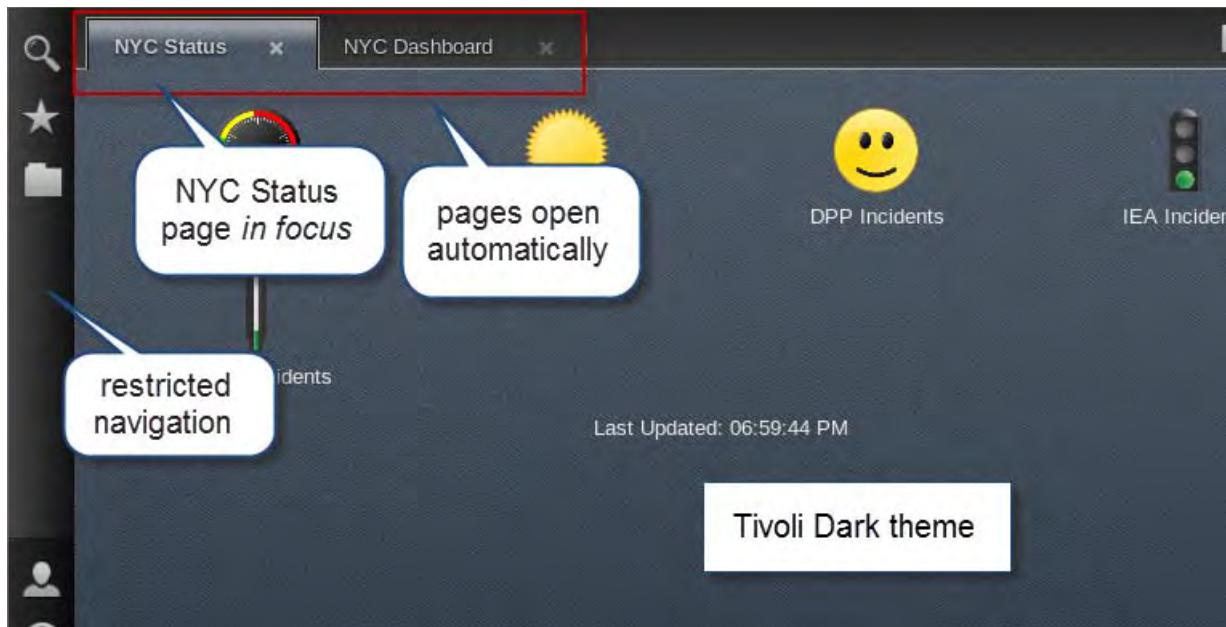
The NYC_Operators group contains the NYC role.

6. Log out of Dashboard Application Services Hub as the **ncoadmin** user.

Validating user access

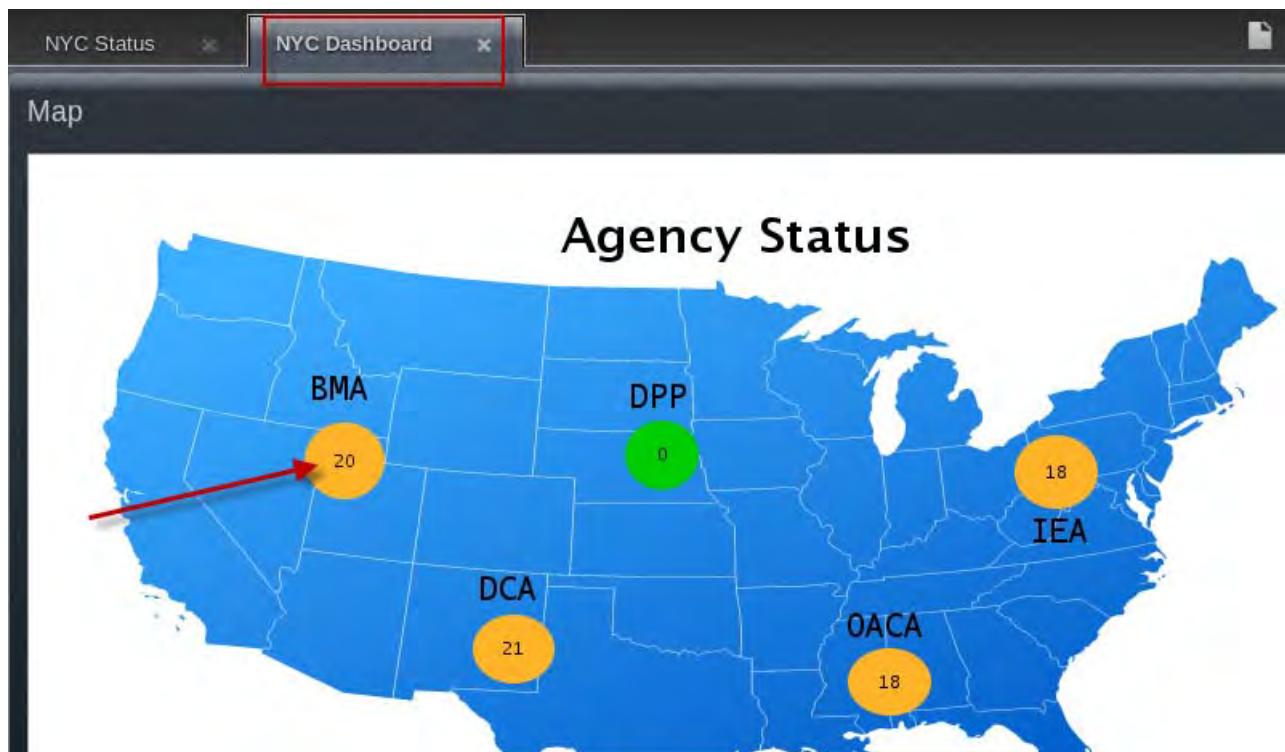
Validate the user environment for the NYC user.

1. Log in as **dselan** with password **object00**.



When the user logs in, the two startup pages open automatically. The NYC Status page is visible. The user cannot access other pages from the navigation bar. The pages appear in the Tivoli Dark theme.

2. Select the **NYC Dashboard** page, and click the status circle for **BMA**.



The Event Viewer opens in a new window.

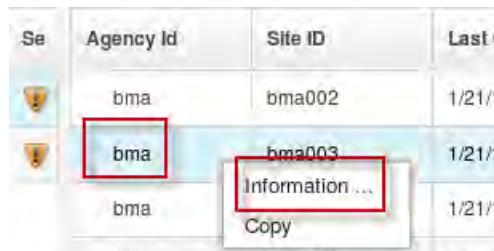
Mozilla Firefox

https://host2.tivoli.edu:16311/ibm/console/webtop/eventViewer/eventViewer.jsp?bidisupport=false&textdirection=l

Group	Co	Se	Agency Id	Site ID	Last Occur	Summary
No items to display			bma	bma002	1/21/15 7:01:00 AM	bma.gov Diskspace alert
			bma	bma002	1/21/15 7:05:34 PM	nsm123-b2.bma.gov Machine has gone o
			bma	bma003	1/21/15 6:59:57 PM	nsm123-c3.bma.gov Diskspace alert
			bma	bma003	1/21/15 7:07:32 PM	nsm123-c2.bma.gov Machine has gone o
			bma	bma003	1/21/15 5:22:17 PM	nsm123-c1.bma.gov Diskspace alert
			bma	bma003	1/21/15 7:07:10 PM	2605-ce2-s.bma.gov Link Down on port

There is no option on the menu bar to select another filter.

- Right-click any event record, and examine the tools.



The user is limited to read-only access to the event.

- Select **Information**.

Journals

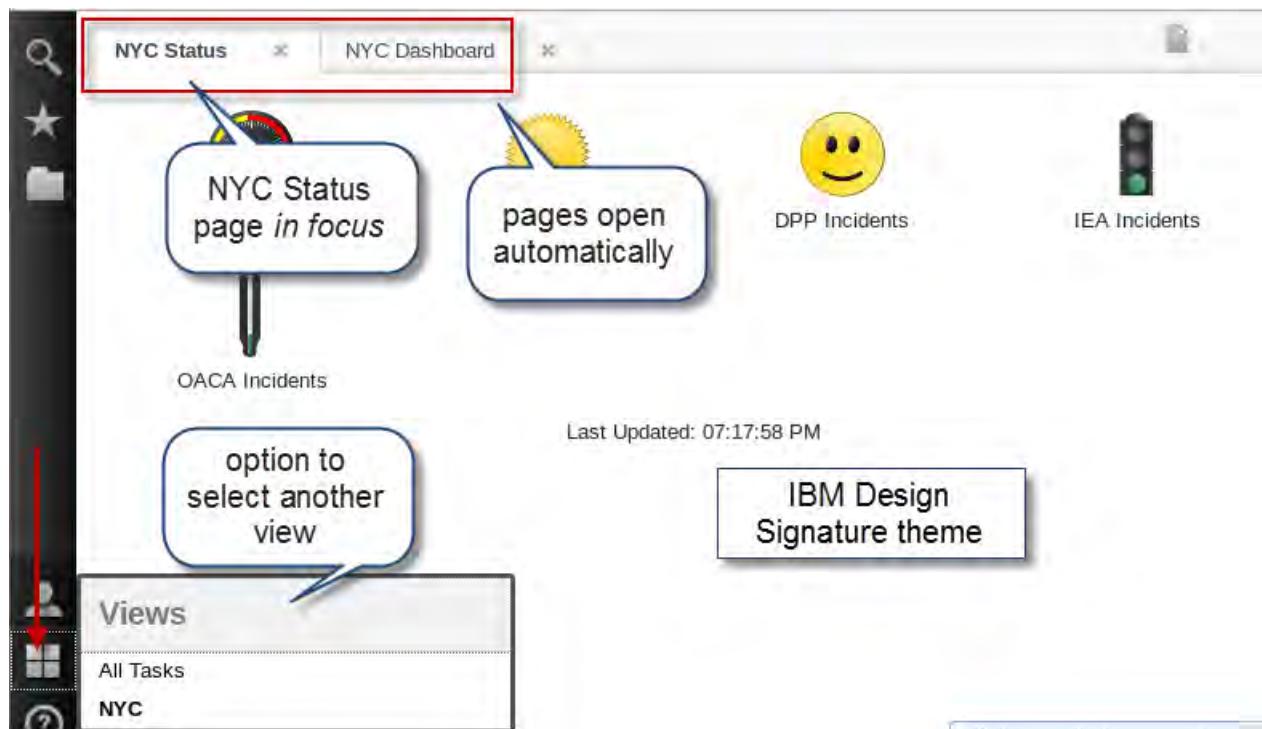
User ID	Date/Tim	Journal Entry
No items to display		

The user can see journal entries, and nothing else.

- Close the Event Viewer window.
- Log out of Dashboard Application Services Hub as the **dselan** user.

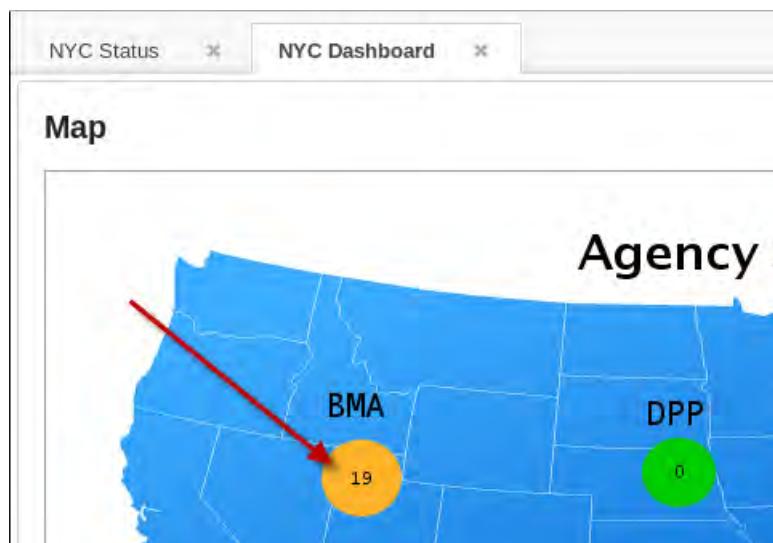
Validate the user environment for the NYC operator user.

1. Log in as **abraman** with password **object00**.



When the user logs in, the two startup pages open automatically. The NYC Status page is visible. The user can click an icon on the navigation bar, and select another view, which provides access to other pages. The pages appear in the IBM Design Signature theme.

2. Select the **NYC Dashboard** page, and click the status circle for **BMA**.



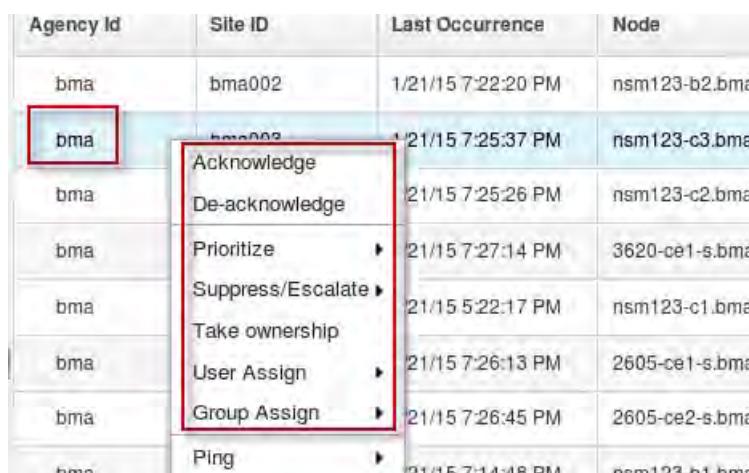
The Event Viewer opens in a new window.

A screenshot of a Mozilla Firefox browser window displaying the IBM Webtop Event Viewer. The URL is https://host2.tivoli.edu:16311/ibm/console/webtop/eventviewer/eventViewer.js. The page shows a list of event records for the group 'NYC_BMA'. The first few rows of the table are:

Agency Id	Site ID	Last Occurrence	Node
bma	bma002	1/21/15 7:22:20 PM	nsm123-b2.bma.gov
bma	bma003	1/21/15 7:25:37 PM	nsm123-c3.bma.gov
bma		1/21/15 7:25:26 PM	nsm123-c2.bma.gov
bma		1/21/15 7:27:14 PM	3620-ce1-s.bma.gov
bma		1/21/15 5:22:17 PM	nsm123-c1.bma.gov
bma		1/21/15 7:26:13 PM	2605-ce1-s.bma.gov
bma		1/21/15 7:26:45 PM	2605-ce2-s.bma.gov
bma		Ping	192.168.1.1

There is an option on the menu bar to select another filter.

3. Right-click any event record, and examine the tools.



The user has access to tools that require read/write access to the event.

4. Select Information.

A screenshot of the Event Viewer showing the 'Fields' tab. It displays the following event details:

Field	Value
Summary	Diskspace alert
Node	nsm123-c3.bma.gov
Severity	Major

The user can see event fields, details, and journal entries.

5. Close the Event Viewer window.

6. Log out of Dashboard Application Services Hub.

With this configuration in place, you can add more users quickly. You can add a user to the NYC_Operator group, and the user inherits the same user environment as the abraman user. You add a user to the NYC_End_User group, and the user inherits the same environment as the dselan user.

Exercise 3 Configuring users for native event list access

In the previous exercise, you created a group for use by NYC operator users. The configuration proves access to event records through the Web GUI desktop. In this exercise, you extend that configuration to enable access to the native desktop.

Configuring the ObjectServer for LDAP password authentication

The users for the class environment are defined in LDAP. WebSphere is configured to use LDAP for the default user repository. When a user is configured with Web GUI roles, the Web GUI synchronization process creates a user in the ObjectServer. The mere existence of the user in the ObjectServer is sufficient for Web GUI. However, if the user attempts to access the ObjectServer directly, the ObjectServer cannot authenticate the password. You must configure each ObjectServer with access to the LDAP server.



Note: You must make configuration changes on both host images for this exercise.

Configuring the host1 image.

1. Switch to the **host1** image.
2. Open a Terminal window if necessary.
3. Configure the LDAP property file as follows.
 - a. Change to the target directory.

```
cd /opt/IBM/tivoli/netcool/omnibus/etc
```
 - b. Save a copy of the file before modifications.

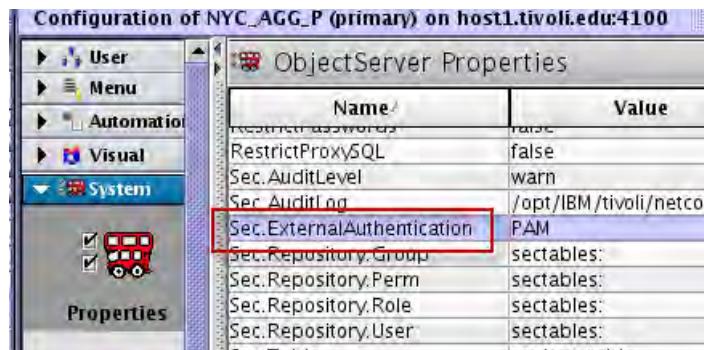
```
cp ldap.props ldap.props.orig
```

- c. Open the file for edit with the gedit utility.

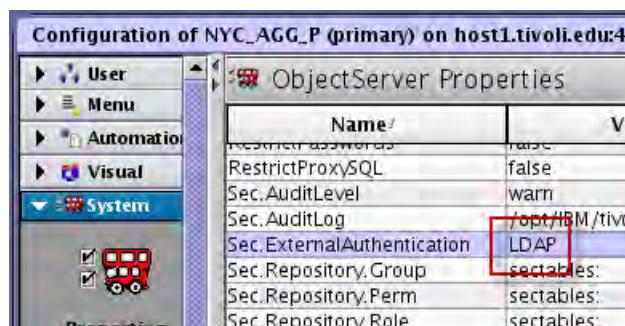
```
gedit ldap.props
```
 - d. Uncomment, and set the following property values:

```
Hostname: 'host1.tivoli.edu'
Port: 389
LDAPSearchFilter: '(uid=%s)'
LDAPSearchBase: 'ou=tipusers,cn=tipRealm,DC=IBM,DC=COM'
```
 - e. Save the changes, and exit the gedit utility.
4. Modify the ObjectServer property to enable the use of LDAP.
 - a. Start the Netcool/OMNIbus Administrator utility.

```
nco_config &
```
 - b. Connect to the **NYC_AGG_P** ObjectServer as the **root** user with password **object00**.
 - c. Locate the **Sec.ExternalAuthentication** property.



- d. Edit the property, and change the value to **LDAP**.



5. Exit the Netcool/OMNIbus Administrator utility.
6. Stop the ObjectServer.

```
nco_pa_stop -server HOST1_PA -password object00 -process MasterObjectServer
```
7. Start the ObjectServer.

```
nco_pa_start -server HOST1_PA -password object00 -process MasterObjectServer
```

8. Verify the status.

```
nco_pa_status -server HOST1_PA -password object00
```

Service Name	Process Name	Hostname	User	Status	PID
Core	MasterObjectServer	host1.tivoli.edunetcool		RUNNING	20618
	SyslogProbe	host1.tivoli.edunetcool		RUNNING	2075
	SnmpProbe	host1.tivoli.edunetcool		RUNNING	2076
	SimnetProbe	host1.tivoli.edunetcool		RUNNING	2449

Configuring the host2 image.

The process is the same on host2 as it was on host1.

1. Switch to the **host2** image.
2. Open a Terminal window if necessary.
3. Configure the LDAP property file as follows.

- a. Change to the target directory.

```
cd /opt/IBM/tivoli/netcool/omnibus/etc
```

- b. Save a copy of the file before modifications.

```
cp ldap.props ldap.props.orig
```

- c. Open the file for edit with the gedit utility.

```
gedit ldap.props
```

- d. Uncomment, and set the following property values:

```
Hostname: 'host1.tivoli.edu'
```

```
Port: 389
```

```
LDAPSearchFilter: '(uid=%s)'
```

```
LDAPSearchBase: 'ou=tipusers,cn=tipRealm,DC=IBM,DC=COM'
```

- e. Save the changes, and exit the gedit utility.

4. Modify the ObjectServer property to enable the use of LDAP.

- a. Start the Netcool/OMNIbus Administrator utility.

```
nco_config &
```

- b. Connect to the **NYC_AGG_B** ObjectServer as the **root** user with password **object00**.

- c. Locate the **Sec.ExternalAuthentication** property.

- d. Edit the property, and change the value to **LDAP**.

5. Exit the Netcool/OMNIbus Administrator utility.

6. Stop the ObjectServer.

```
nco_pa_stop -server HOST2_PA -password object00 -process BackupObjectServer
```

7. Start the ObjectServer.

```
nco_pa_start -server HOST2_PA -password object00 -process BackupObjectServer
```

8. Verify the status.

```
nco_pa_status -server HOST2_PA -password object00
```

Service Name	Process Name	Hostname	User	Status	PID
Core	BackupObjectServer	host2.tivoli.edunetcool		RUNNING	13603
	BackupGateway	host2.tivoli.edunetcool		RUNNING	2602
	ArchiveGateway	host2.tivoli.edunetcool		RUNNING	2178
	LondonObjectServer	host2.tivoli.edunetcool		RUNNING	2179
	SimnetProbe	host2.tivoli.edunetcool		RUNNING	2180

Configuring the ObjectServer user environment

The ObjectServers are configured to use the LDAP server as a source for *external* password authentication. The abraman user exists in the ObjectServer, but is disabled.



Note: You can use either image to complete the remaining steps.

1. Start the Netcool/OMNIbus Administrator utility.

```
nco_config &
```

2. Connect to the **NYC_AGG_P** ObjectServer as the **root** user with password **object00**.

3. Expand the **User** feature, and select **Groups**.

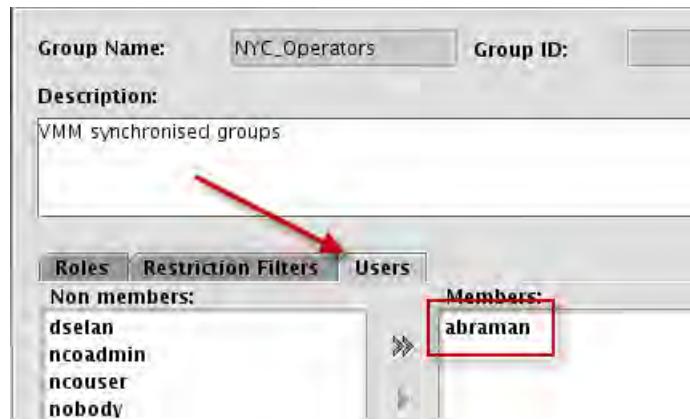
Name	Description
Administrator	Admin Group
Gateway	Permissions required for a gateway user
ISQL	Read only ISQL access
ISQLWrite	Write ISQL access
Netcool_Admin	VMM synchronised groups
Netcool_User	VMM synchronised groups
Normal	Normal Group
NYC_End_Users	VMM synchronised groups
NYC_Operators	VMM synchronised groups
Operations	VMM synchronised groups

The Web GUI synchronization process created the **NYC_End_users**, and **NYC_Operators** groups in the ObjectServer.

6 User administration exercises

Exercise 3 Configuring users for native event list access

4. Right-click **NYC_Operators**, and select **Edit Group**. Select the **Users** tab.



The Web GUI synchronization process added the abraman *ObjectServer* user to **NYC_Operators** *ObjectServer group*. The synchronization process maintains the user-group relationships.

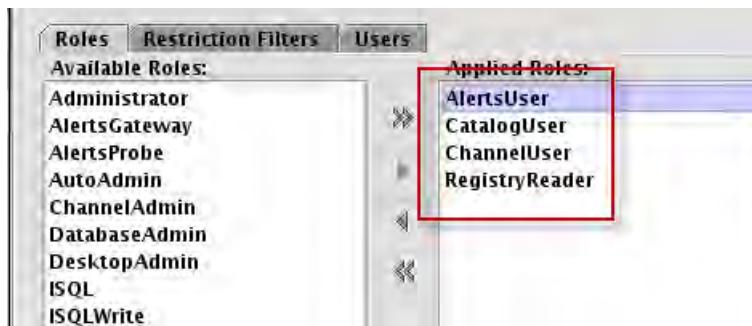
5. Click the **Roles** tab.



The Web GUI synchronization process does not add any *ObjectServer roles* to **NYC_Operators** *ObjectServer group*. You must add the roles manually.

6. Add the following roles to the group:

AlertsUser
CatalogUser
ChannelUser
RegistryReader





Hint: Examine the roles that are assigned to the Normal group to see what a normal user requires.

7. Click **OK** to save the changes.
8. Click **Users**. Right-click **abraman** and select **Edit User**.

The screenshot shows the 'Configuration of NYC_AGG_P (primary) on host1.tivoli.edu:4100' window. On the left, there's a sidebar with a 'User' icon and a 'Users' button. The main area is a table titled 'Users' with columns 'Name/' and 'Full Name'. It lists four entries: 'abraman' (Ariana Braman Braman), 'dselan' (Dick Selan Selan), 'ncoadmin' (Netcool Admin), and 'ncouser' (Netcool User). The 'abraman' row is highlighted with a red box.

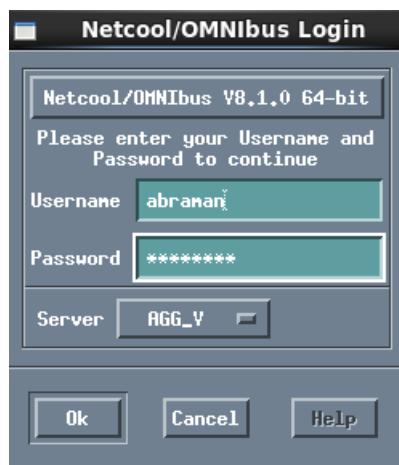
9. Select the **Settings** tab. Enable **External Authentication**. Enable the user. Click **OK** to save the changes.

The screenshot shows the 'Settings' tab of the user configuration dialog. It has tabs for 'Groups', 'Restriction Filters', and 'Settings'. Under 'Settings', there are fields for 'Password' and 'Verify', and a 'User Type' dropdown set to 'Normal User'. Two checkboxes are highlighted with red boxes: 'External Authentication' (which is checked) and 'User Enabled' (which is also checked).

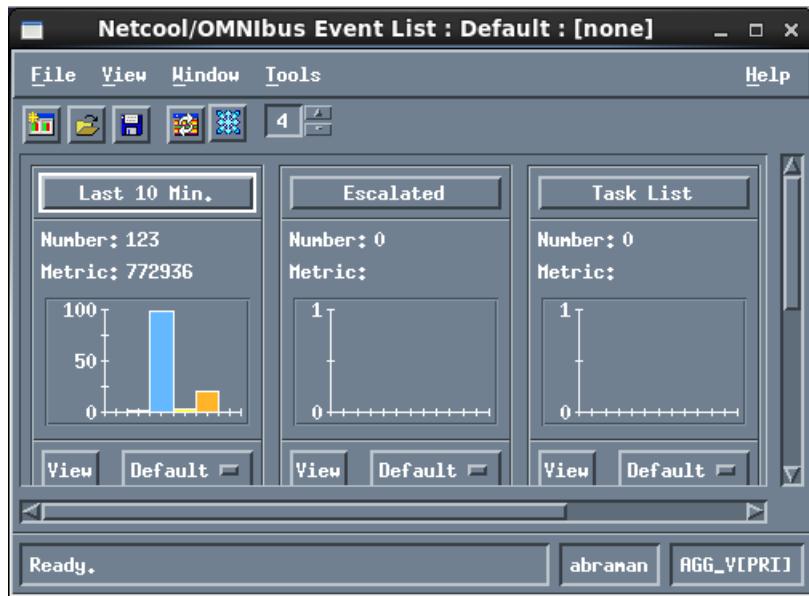
10. Start the native desktop.

```
nco_event &
```

11. Log in as user **abraman** with password **object00**.



The Event List window opens.



The abraman user can access event records with the native desktop.

12. Close the Event List window.

When the abraman user logs in to Dashboard Application Services Hub, the user sees default pages. The desktop pages organize the event records based on filters, and views. The filters, views, and pages are specific to the Web GUI environment. You must create the corresponding filters, and views in the ObjectServer to completely replicate the user environment.

13. Connect to the NYC_AGG_B ObjectServer with the Netcool/OMNIBus Administrator utility.

14. Expand **User**, and select **Groups**. Right-click **NYC_Operators**, and select **Edit Group**.



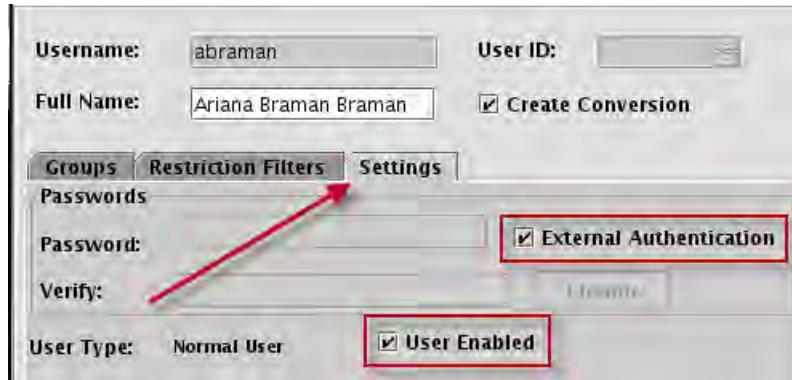
You modified the NYC_Operators group in the NYC_AGG_P ObjectServer. The ObjectServer bidirectional gateway replicated the changes to the NYC_AGG_B ObjectServer.



Important: You did not configure the gateway during this class. The gateway was already configured on the class image.

15. Click **Cancel**.

16. Select **Users**. Right-click **abraman**, and select **Edit User**. Click the **Settings** tab.



You modified the abraman user in the NYC_AGG_P ObjectServer. The ObjectServer bidirectional gateway replicated the changes to the NYC_AGG_B ObjectServer.

17. Close the Netcool/OMNIbus Administrator utility.

With the current configuration in place, if you add a user to the NYC_Operators group with the WebSphere administrative console, the Web GUI synchronization process creates the same user in the ObjectServer. The ObjectServer user is added to the NYC_Operators ObjectServer group. The NYC_Operators ObjectServer group contains the roles that are required for event access. If that user requires access to the native desktop, you must edit the ObjectServer user, set the user for external authentication, and enable the user. The ObjectServer bidirectional gateway replicates the changes to the NYC_AGG_B ObjectServer.



7 Customizing Tivoli Common Reporting reports exercises

The goal of these exercises is to introduce you to using the IBM Cognos® Business Intelligence Modeling tool, Framework Manager. You use Framework Manager to extend an existing Cognos data model for use in Tivoli Common Reporting. In this lab session, you modify the Tivoli Netcool/OMNIbus data model. The model modification supports access to more columns in the Netcool/OMNIbus REPORTER database for use in Tivoli Common Reporting and Cognos reports.

Exercise 1 Modifying the event archive database

In a previous exercise, you added extra columns to the ObjectServer event record. You must modify the archive database to add corresponding columns.



Important: You perform the following exercise on the **host2** image.

1. Switch to the **host2** image.
2. Open a Terminal window if necessary.
3. Change to the db2inst1 user.

```
su - db2inst1  
Password: object00
```



Important: You must modify the archive database as the db2inst1 user.

To facilitate the exercise, the image includes an SQL file. The SQL file contains commands to add columns to the archive database *event* table.

4. Examine the file.

```
cd /workshop/unit07  
more nyc.sql
```

```
ALTER TABLE REPORTER_STATUS ADD COLUMN AgencyId varchar(10) @  
ALTER TABLE REPORTER_STATUS ADD COLUMN SiteId varchar(10) @  
ALTER TABLE REPORTER_STATUS ADD COLUMN ContactName varchar(40) @  
ALTER TABLE REPORTER_STATUS ADD COLUMN ContactEmail varchar(40) @  
ALTER TABLE REPORTER_STATUS ADD COLUMN ContactPhone varchar(30) @  
ALTER TABLE REPORTER_STATUS ADD COLUMN SiteAddr varchar(40) @  
ALTER TABLE REPORTER_STATUS ADD COLUMN SiteCity varchar(40) @  
ALTER TABLE REPORTER_STATUS ADD COLUMN SiteState varchar(10) @  
ALTER TABLE REPORTER_STATUS ADD COLUMN SiteName varchar(40) @  
ALTER TABLE REPORTER_STATUS ADD COLUMN SiteCountry varchar(10) @
```



Note: In this example, the column names for the archive database are the same as the names in the ObjectServer event record, which is not mandatory.

5. Connect to the archive database.

```
db2 connect to REPORTER
```

Database Connection Information

```
Database server      = DB2/LINUXX8664 10.5.3  
SQL authorization ID = DB2INST1  
Local database alias = REPORTER
```

6. Import the SQL file.

```
db2 -td@ -vf nyc.sql
```

- Verify that the column names are added.

```
db2 describe table REPORTER_STATUS
```

.					
.					
.					
AGENCYID	SYSIBM	VARCHAR	10	0	Yes
SITEID	SYSIBM	VARCHAR	10	0	Yes
CONTACTNAME	SYSIBM	VARCHAR	40	0	Yes
CONTACTEMAIL	SYSIBM	VARCHAR	40	0	Yes
CONTACTPHONE	SYSIBM	VARCHAR	30	0	Yes
SITEADDR	SYSIBM	VARCHAR	40	0	Yes
SITECITY	SYSIBM	VARCHAR	40	0	Yes
SITESTATE	SYSIBM	VARCHAR	10	0	Yes
SITENAME	SYSIBM	VARCHAR	40	0	Yes
SITECOUNTRY	SYSIBM	VARCHAR	10	0	Yes

- Exit the **db2inst1** user back to the **netcool** user.

```
exit
```

Exercise 2 Modifying the archive gateway

The Netcool/OMNIbus Gateway for JDBC copies ObjectServer event data to the archive database. You must modify the gateway configuration to accommodate the extra column names.



Important: You perform the following exercise on the **host2** image.

- Stop the archive gateway.

```
nco_pa_stop -server HOST2_PA -process ArchiveGateway -password object00
```

- Verify that the gateway is stopped.

```
nco_pa_status -server HOST2_PA -password object00
```

Service Name	Process Name	Hostname	User	Status	PID
Core	BackupObjectServer	host2.tivoli.edunetcool		RUNNING	8706
	BackupGateway	host2.tivoli.edunetcool		RUNNING	8450
	ArchiveGateway	host2.tivoli.edunetcool		DEAD	0
	LondonObjectServer	host2.tivoli.edunetcool		RUNNING	2231
	SimnetProbe	host2.tivoli.edunetcool		RUNNING	2232

- Modify the gateway configuration as follows.

- Change to the location of the gateway configuration files.

```
cd /opt/IBM/tivoli/netcool/omnibus/etc
```

- b. Save a copy of the file before modifications.

```
cp JDBC_GATE.map JDBC_GATE.map.orig
```

- c. Open the file for edit with the gedit utility.

```
gedit JDBC_GATE.map
```

- d. Locate the following section in the file.

```
'X733PROBABLECAUSE' = '@X733ProbableCause',
'X733SPECIFICPROB' = '@X733SpecificProb'    ON INSERT ONLY,
'X733CORRNOTIF' = '@X733CorrNotif'          ON INSERT ONLY,
'ORIGINALSEVERITY' = '@Severity'              ON INSERT ONLY,
# NB do not concatenate additional values for ServerName and ServerSerial !
'SERVERNAME' = '@ServerName'                ON INSERT ONLY,
'SERVERSERIAL' = '@ServerSerial'             ON INSERT ONLY
);
CREATE MAPPING
(
  'SERIAL'          AS 'serial',
  'DB2 column name'          AS 'ObjectServer column name'
);
```

The gateway map file provides the mapping from ObjectServer column name to DB2 column name, which is why the two names do not have to match.

- e. Add the following lines.

```
'AgencyId'      = '@AgencyId',
'SiteId'        = '@SiteId',
>ContactName'   = '@ContactName',
>ContactEmail'  = '@ContactEmail',
>ContactPhone'  = '@ContactPhone',
'SiteAddr'       = '@SiteAddr',
'SiteCity'       = '@SiteCity',
'SiteState'      = '@SiteState',
'SiteName'       = '@SiteName',
'SiteCountry'    = '@SiteCountry',
```

```
'ORIGINALSEVERITY' = '@Severity'          ON INSERT OF
'AgencyId'      = '@AgencyId',
'SiteId'        = '@SiteId',
>ContactName'   = '@ContactName',
>ContactEmail'  = '@ContactEmail',
>ContactPhone'  = '@ContactPhone',
'SiteAddr'       = '@SiteAddr',
'SiteCity'       = '@SiteCity',
'SiteState'      = '@SiteState',
'SiteName'       = '@SiteName',
'SiteCountry'    = '@SiteCountry',|
```



Important: Spelling, and punctuation is important.

- f. Save the changes, and exit the gedit utility.

4. Remove the old gateway log file.

```
cd /opt/IBM/tivoli/netcool/omnibus/log
rm JDBC_GATE.log
```



Hint: If the gateway does not start, you must look for errors in the log file.

5. Start the gateway.

```
nco_pa_start -server HOST2_PA -process ArchiveGateway -password object00
```

6. Check the status of the gateway.

```
nco_pa_status -server HOST2_PA -password object00
```

Service Name	Process Name	Hostname	User	Status	PID
Core	BackupObjectServer	host2.tivoli.edunetcool		RUNNING	8706
	BackupGateway	host2.tivoli.edunetcool		RUNNING	8450
	ArchiveGateway	host2.tivoli.edunetcool		RUNNING	11712
	LoadOnObjectServer	host2.tivoli.edunetcool		RUNNING	2231
	SimnetProbe	host2.tivoli.edunetcool		RUNNING	2232

If the gateway status is PENDING or DEAD, the gateway configuration file contains a syntax error. The most common issues are spelling and punctuation. Review the log file for issues.

7. Verify whether the gateway is populating the columns in the archive database.

- a. Change to the **db2inst1** user.

```
su - db2inst1
Password: object00
```

- b. Connect to the archive database.

```
db2 connect to REPORTER
```

Database Connection Information

```
Database server      = DB2/LINUXX8664 10.5.3
SQL authorization ID = DB2INST1
Local database alias = REPORTER
```

- c. Remove all records from the **reporter_status** table by using the following command:

```
db2 delete from reporter_status
```

DB20000I The SQL command completed successfully.

Wait a few minutes for the gateway to add new records. By default, the gateway processes records every 60 seconds.

- d. Query the database.

```
db2 select AgencyId,SiteId from REPORTER_STATUS
```

```
.
```

```
.
```

```
.
```

```
dca      dca070
```

```
-        -
```

```
oaca    oaca001
```

```
-        -
```

```
bma      bma001
```

- e. Exit the **db2inst1** user back to the **netcool** user.

The archive database now contains extra column names. The JDBC gateway is populating the columns with data from the ObjectServer event record.



Important: The remaining exercises are performed on the **host3** image.

Exercise 3 Installing DB2

The Framework Manager application uses a local copy of DB2 to access the remote DB2 database. In this exercise, you install DB2 and define a database connection to the REPORTER database, which is running on the **host2** server.

1. Start the **host3** image.

Wait for the image to activate.

2. Log in as **tivuser** with password **object00**.

3. Verify access to the other images.

- a. Open a command window.

- b. Ping host1.

```
ping host1.tivoli.edu
Pinging host1.tivoli.edu [192.168.100.160] with 32 bytes of data:
Reply from 192.168.100.160: bytes=32 time=1ms TTL=64
Reply from 192.168.100.160: bytes=32 time<1ms TTL=64
Reply from 192.168.100.160: bytes=32 time=1ms TTL=64
Reply from 192.168.100.160: bytes=32 time<1ms TTL=64
```

- c. Ping host2.

```
ping host2.tivoli.edu
```

```
Pinging host2.tivoli.edu [192.168.100.161] with 32 bytes of data:
```

```
Reply from 192.168.100.161: bytes=32 time<1ms TTL=64
```

4. Install DB2 as follows:

- a. Open the Windows Explorer utility.

- b. Browse to the location of the installation file.

C:\workshop\db2

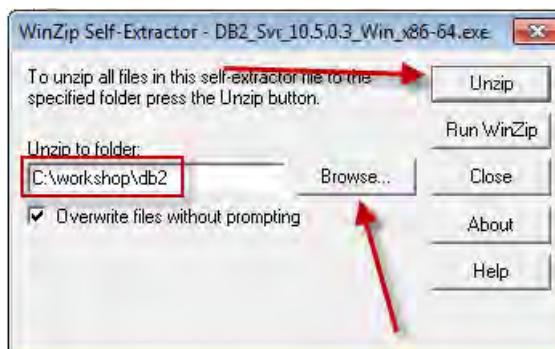
The installation file is packaged as a self-extracting archive.

- c. Double-click the installation file to expand the archive.

DB2_Svr_10.5.0.3_Win_X86-64.exe

- d. Click **Browse**, and select the following folder. Then, click **Unzip**.

C:\workshop\db2



Note: The extractor runs for approximately 10 minutes.

- e. Click **Close** to close the extractor.

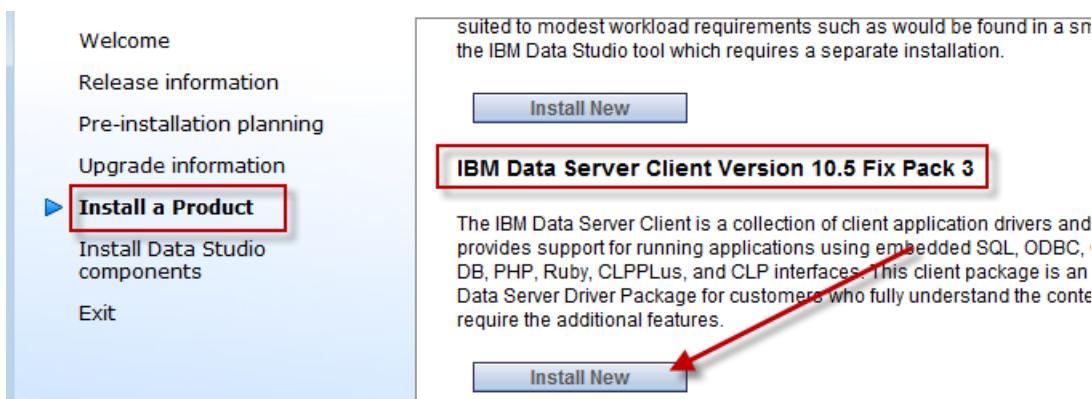
- f. Browse to the location of the installer

C:\workshop\db2\SERVER\image

- g. Double-click the installer

setup.exe

- h. Click **Install a Product**. Scroll down, and locate **IBM Data Server Client Version 10.5 Fix Pack 3**. Click **Install New**.



The DB2 setup wizard opens.

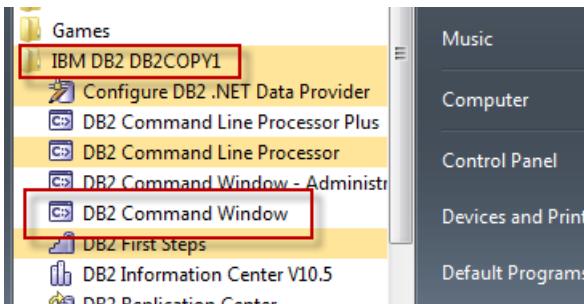
- i. Click **Next**.
- j. Accept the license agreement, and click **Next**.
- k. Leave the Installation Type as **Typical**, and click **Next**.
- l. Leave the default to install the client, and create a response file. Click **Next**.
- m. Leave the default installation folder, and click **Next**.
- n. Click **OK** to close the warning window.
- o. Click **Finish** to start the installation.
- p. Click **Yes** to the security message.
- q. When the installation is complete, click **Next**.
- r. Click **Finish**.
- s. Close the First Steps window.

5. Create a database alias.

The Netcool/OMNIbus Tivoli Common Reporting reports require access to a database called REPORTER. You must create a database alias on **host3** that points to the database on **host2**.

- a. Open a DB2 Command Window.

All Programs > IBM DB2 DB2COPY1 > DB2 Command Window



- b. Catalog the node.

```
db2 catalog tcpip node host2 remote host2.tivoli.edu server 50000
```

DB20000I The CATALOG TCPIP NODE command completed successfully.

DB21056W Directory changes may not be effective until the directory cache is refreshed.

- c. Catalog the database.

```
db2 catalog database REPORTER at node host2
```

DB20000I The CATALOG DATABASE command completed successfully.

DB21056W Directory changes may not be effective until the directory cache is refreshed.

- d. Refresh the directory.

```
db2 terminate
```

DB20000I The TERMINATE command completed successfully.

- e. Verify access to the database.

```
db2 connect to REPORTER user db2inst1 using object00
```

Database Connection Information

Database server = DB2/LINUXX8664 10.5.3

SQL authorization ID = DB2INST1

Local database alias = REPORTER

- f. Close the DB2 Command Window.

Exercise 4 Installing Framework Manager

1. Install Framework Manager as follows:

- a. Open the Windows Explorer utility if necessary.

- b. Browse to the location of the installation file.

C:\workshop\fm

The installation file is packaged as compressed archive.

- c. Double-click the file to expand the archive.

ITCR_3.1.0.1_FOR_CFM_WINS.zip

- d. Browse to the location of the installer
`C:\workshop\fm\CognosModeling\win32`
- e. Double-click the installer
`issetup.exe`
- f. Click **Extract All**.
- g. Leave the default installation folder, and click **Extract**.

Wait the files to extract. When the extract is complete, another copy of the Windows Explorer utility opens. Use the new copy for the remaining steps.

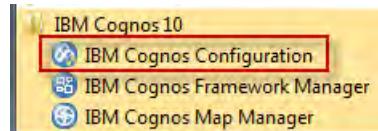
- h. Browse to the location of the installer
`C:\workshop\fm\CognosModeling\win32`
- i. Double-click the installer
`issetup.exe`
- j. Click **Yes** to allow the installation.
- k. Click **Next** at the welcome screen.
- l. Accept the license agreement, and click **Next**.
- m. Leave the default installation folder, and click **Next**.
- n. Click **Yes** to create the folder.
- o. Leave the default selection of components, and click **Next**.
- p. Leave the default shortcut folder, and click **Next**.
- q. Click **Next** at the installation summary.

The installation begins and runs for several minutes.

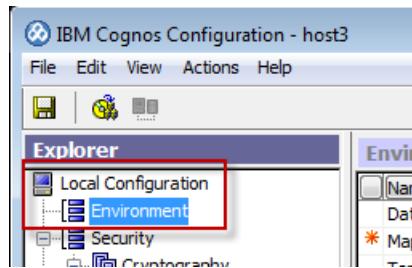
- r. When the installation is complete, click **OK** to close the information window.
- s. Click **Finish** to close the installation utility.
- t. Click **OK** to close the message regarding supplemental language packs.
- u. Click **This program installed correctly**.

2. Start IBM Cognos Configuration.

All Programs > IBM Cognos 10 > IBM Cognos Configuration



- Click **Yes** to allow the changes.
- In the left **Explorer** pane, select **Local Configuration > Environment**.



- In the right **Environment - Group Properties** pane, configure the following two URIs by typing the following text in the applicable **Value** field:

Gateway URI: **http://host2.tivoli.edu:16310/tarf/servlet/dispatch**

Dispatcher URI: **http://host2.tivoli.edu:16310/tarf/servlet/dispatch/ext**

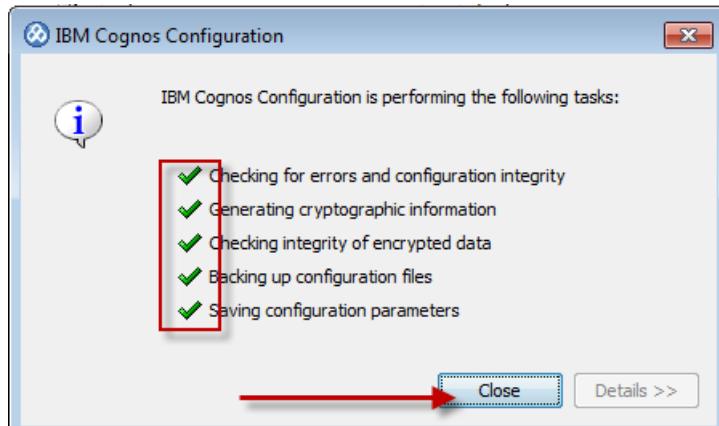
Environment - Group Properties	
Name	Value
Data files location/data
* Map files location/maps
Temporary files location/temp
Encrypt temporary files?	False
Sort buffer size in MB	4
* IP Version for Host Name Resolution	Use IPv4 addresses
Gateway Settings	
* Gateway URI	http://host2.tivoli.edu:16310/tarf/servlet/dispatch
Other URI Settings	
* Dispatcher URI for external applications	http://host2.tivoli.edu:16310/tarf/servlet/dispatch/ext
Font Settings	



Important: The URI values must be entered exactly as shown.

- Click **File > Save**.

- e. Verify that all boxes have a green check mark, and click **Close**.



- f. Click **File > Exit**.

Framework Manager is now installed and configured to connect to Tivoli Common Reporting.

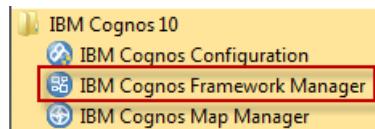
Exercise 5. Modifying and publishing the Cognos data model

To use the newly added database columns in the Tivoli Common Reporting report authoring tools, you must add the fields to the Cognos data model. To preserve the integrity of the original Netcool/OMNIbus model, you make a copy. You make the necessary Cognos data model modifications to the copy. In Framework Manager, the copy action is performed through the action called **branching**. After you perform the branch, you save the model as a new data model name, and then modify it to add the new columns to the necessary views.

The data model is contained within a Framework Manager report package. The report package is included with Netcool/OMNIbus. To facilitate this exercise, the report package file is on the host3 server.

1. Start Framework Manager:

All Programs > IBM Cognos 10 > IBM Cognos Framework Manager



2. Click **Open a project**.
3. Browse to the **C:\workshop\model** directory. Select the **Netcool_OMNIbus_MODEL.cpf** file and click **Open**.



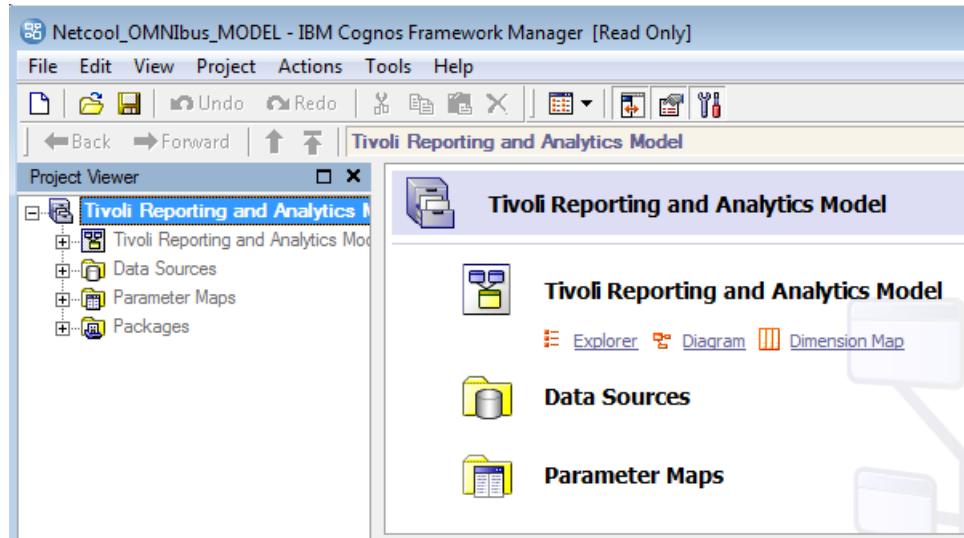
Note: The Cognos data model for the event history reports is bundled with the Netcool/OMNibus software. The model is found here:
\$OMNIHOME/extensions/tcr_event_reports/Model/
You need the complete contents of the **Model** directory.

Framework Manager connects to Tivoli Common Reporting on host2.

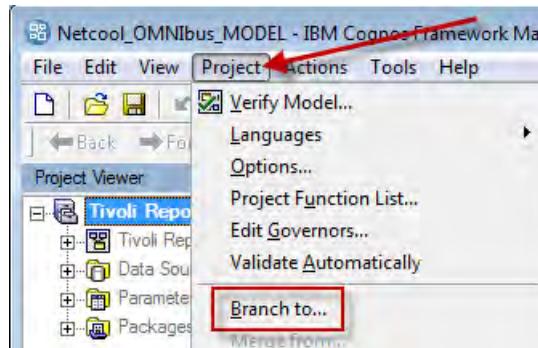
4. Enter the **smadmin**, and **object00** account information as shown. Click **OK**.



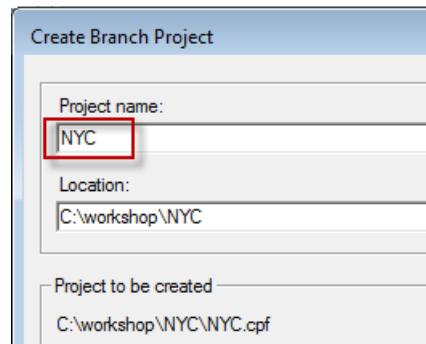
You are now in the original Netcool_OMNIbus_MODEL project as shown:



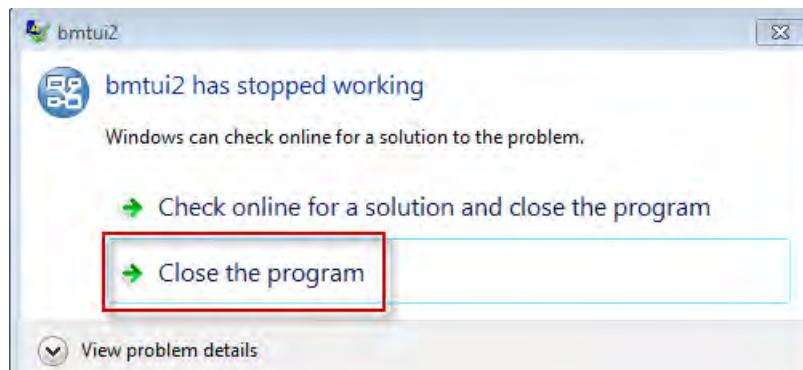
5. In the Framework Manager interface, select **Project > Branch to:**



6. In the Create Branch Project window, enter **NYC** for the project name, and click **OK**.



7. Click **Close the program**.

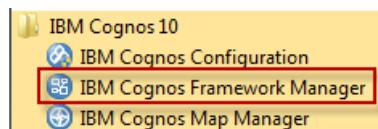


Framework Manager closes.

Note: This behavior is related to the Windows user not having the necessary privileges.

8. Start **Framework Manager**:

All Programs > IBM Cognos 10 > IBM Cognos Framework Manager



9. Click **Open a project**.

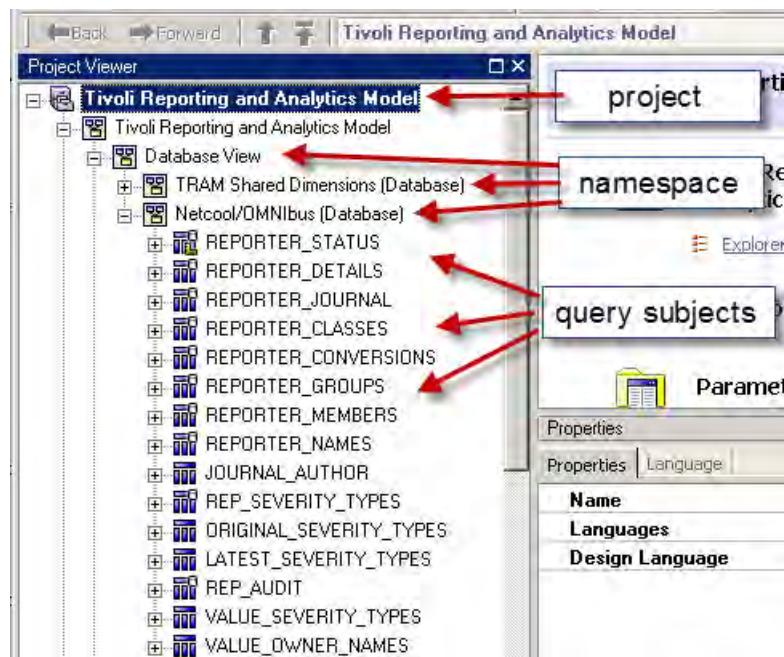
10. Browse to the **C:\workshop\NYC** directory. Select the **NYC.cpf** file, and click **Open**.
11. Enter the **smadmin**, and **object00** account information as shown. Click **OK**.

Database view modifications

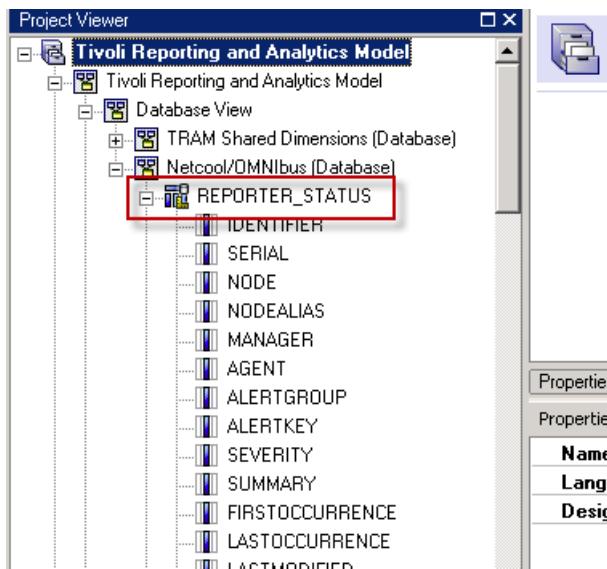
You are now in the NYC copy of the original Netcool/OMNibus data model. The original data model did not have the extra columns in the database. You must add the fields to the various model views. Adding the fields to the model makes them available for use in Tivoli Common Reporting report authoring tools.

In this part of the exercise, you work in the *Database View* to update the current Cognos model to represent the modified physical database model. You add the extra Netcool/OMNibus REPORTER database fields to the REPORTER_STATUS table of the Netcool/OMNibus Database view.

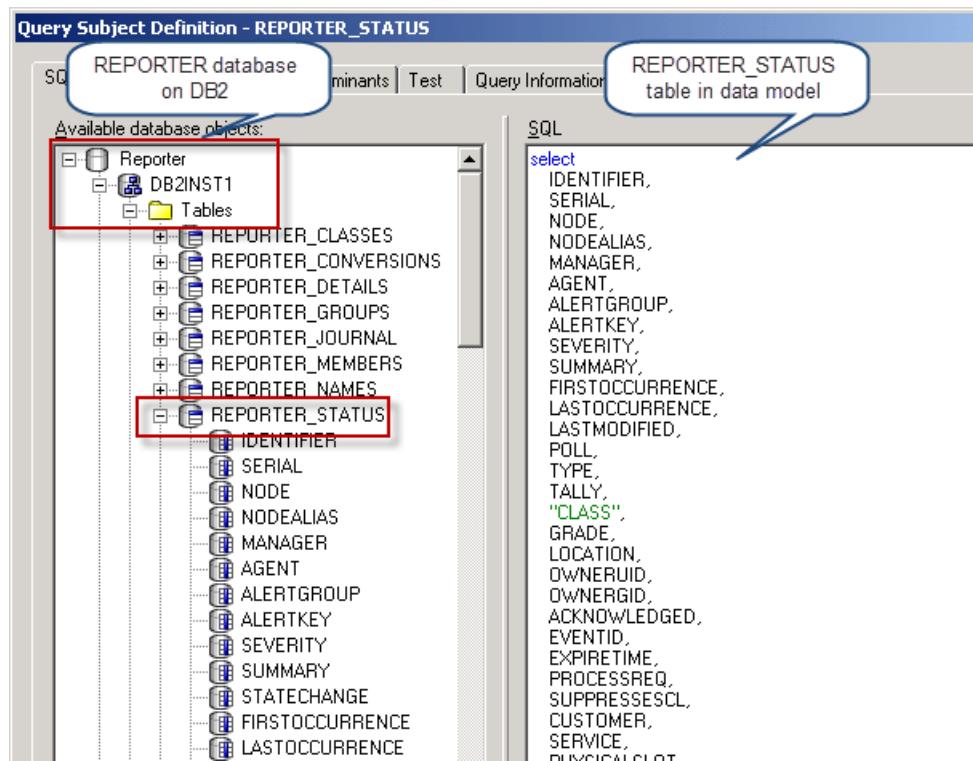
1. Expand the **Tivoli Reporting and Analytics Model > Database View > Netcool/OMNibus (Database)** namespace to see the various object details as shown:



2. Expand the **REPORTER_STATUS** query subject as shown:



3. Double-click the **REPORTER_STATUS** table in the Project Viewer pane and the Query Subject Definition - REPORTER_STATUS window opens as shown:

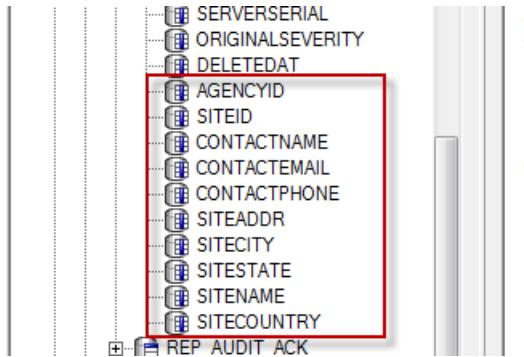


The left side of the view shows the entire contents of the REPORTER database as it is defined in DB2. The right side of the view shows the structure of the REPORTER_STATUS query subject as it is defined in the data model.



Note: A Cognos Query subject is functionally equivalent to a database table.

4. Scroll down in the left pane and locate the new columns.



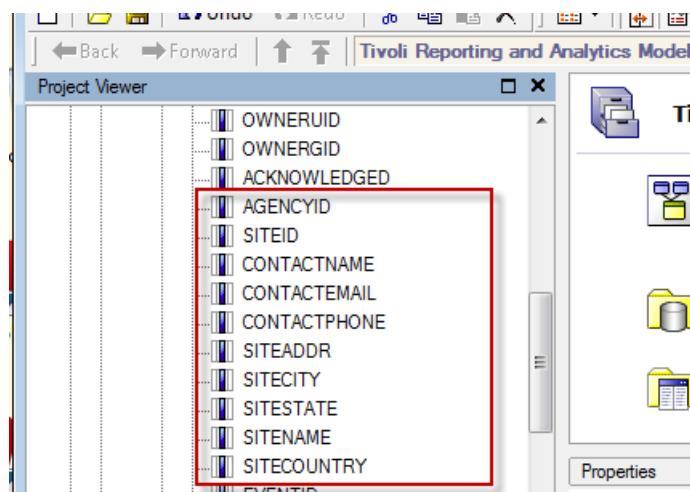
The new database fields must be added to the **Database view > REPORTER_STATUS** query subject (table).

5. Hold the *Ctrl* key down, and click each column to select the columns. Drag the columns into the right window pane in the SQL pane's **select** area as shown. In the example, the columns were placed after the **Acknowledged** column, but any location within the **select** area is acceptable.



6. Click **OK** on the bottom of the page to close the window.

The extra fields are now part of the **Netcool/OMNIbus (Database) > REPORTER_STATUS** table as shown:



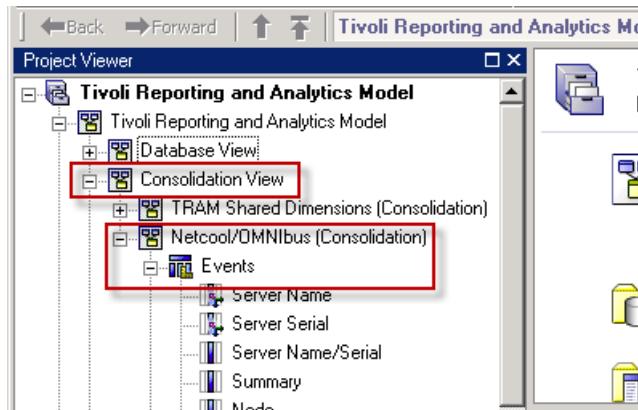
7. Collapse the **Tivoli Reporting and Analytics Model > Database View** namespace.

Consolidation view modifications

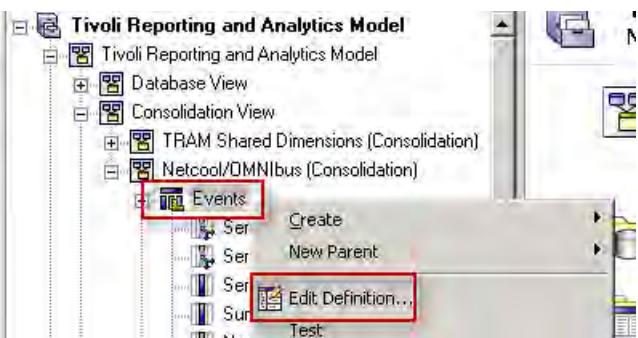
Now that the Database View is updated, you add the new fields to the Consolidation view. Remember that a Consolidation View is the user-oriented, business value, version of the underlying Database View. In the Netcool/OMNIBus Consolidation view, the REPORTER_STATUS table is renamed to the EVENTS table to assist users during the report creation process.

In this part of the lab session, you add the new fields to the **Consolidation View** EVENTS query subject (table). To assist the users, you rename the added columns to more obvious names.

1. Expand the **Tivoli Reporting and Analytics Model > Consolidation View > Netcool/OMNIBus (Consolidation)** namespace to view the object details as shown:



2. Select the **Events** query subject object (table). Right-click, and select **Edit Definition**.



3. In the Query Subject Definition - Events window, expand the **Available Model Objects** tree: **Tivoli Reporting and Analytics Model > Database View > Netcool/OMNIBus (Database) >**

REPORTER_STATUS table to view the query subjects (columns in the table database view) as shown:

The screenshot shows the 'Query Subject Definition - Events' window. The 'Available Model Objects' pane on the left lists 'Tivoli Reporting and Analytics Model', 'Database View', 'TRAM Shared Dimensions (Data)', and 'Netcool/OMNibus (Database)'. Under 'Netcool/OMNibus (Database)', the 'REPORTER_STATUS' table is selected and highlighted with a red box. The 'Query Items and Calculations' pane on the right lists various items with their sources, such as 'Server Name' (source: trim(REPORTER_STATUS.SERVERNAME)), 'Server Serial' (source: REPORTER_STATUS.SERVERSERIAL), and 'REPORTER_STATUS' (source: trim(REPORTER_STATUS.REPORTERSTATUS)). A red box highlights the 'REPORTER_STATUS' item in the list.

All of the table columns in the **Netcool/OMNibus Database View > REPORTER_STATUS** are presented and the new columns are available for use as query items in the **Events (Consolidation View)** object.

4. In the Available Model Objects window, scroll down the **Netcool/OMNibus (Database) > REPORTER_STATUS** table (object) to locate the new columns as shown:

The screenshot shows the 'Available Model Objects' pane with the 'REPORTER_STATUS' table expanded. A red box highlights a group of new columns: 'OWNERID', 'ACKNOWLEDGED', 'AGENCYID', 'SITEID', 'CONTACTNAME', 'CONTACTEMAIL', 'CONTACTPHONE', 'SITEADDR', 'SITECITY', 'SITESTATE', 'SITENAME', 'SITECOUNTRY', 'EVENTID', and 'EXPIRETIME'. These columns represent the new data from the 'REPORTER_STATUS' table.

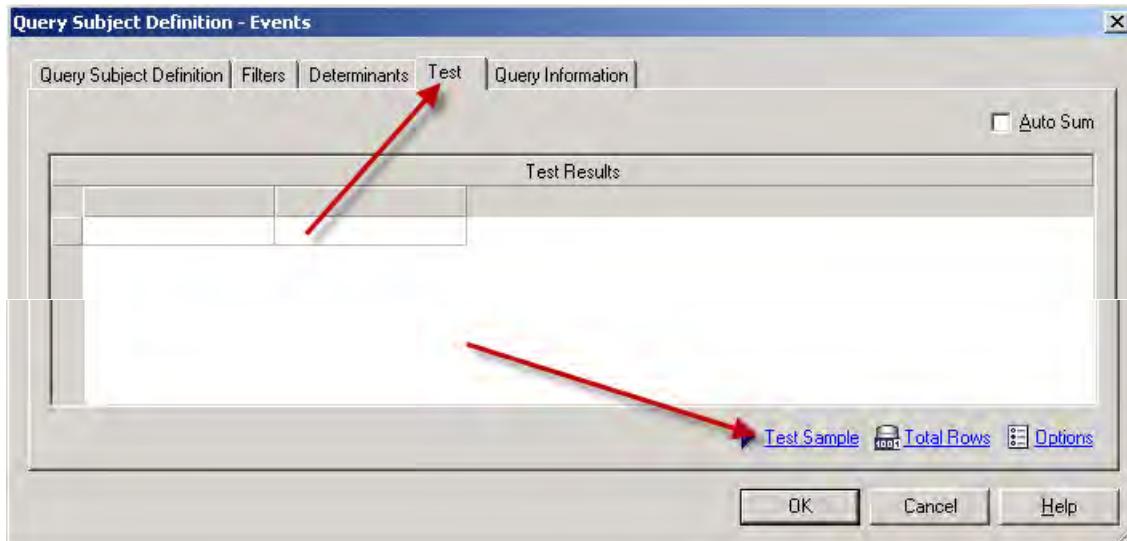
5. Use the **Ctrl** key to select the columns, and drag them to the right pane to add them to the **Query Items and Calculations** list for the **Events** (Consolidation) object as shown:

OWNERUID	Latest Severity Name
OWNERGID	First Occurrence
ACKNOWLEDGED	Last Occurrence
AGENCYID	Deleted At
SITEID	Tally
CONTACTNAME	AGENCYID
CONTACTEMAIL	SITEID
CONTACTPHONE	CONTACTNAME
SITEADDR	CONTACTEMAIL
SITECITY	CONTACTPHONE
SITESTATE	SITEADDR
SITENAME	SITECITY
SITECOUNTRY	SITESTATE
EVENTID	SITENAME
EXPIRETIME	SITECOUNTRY

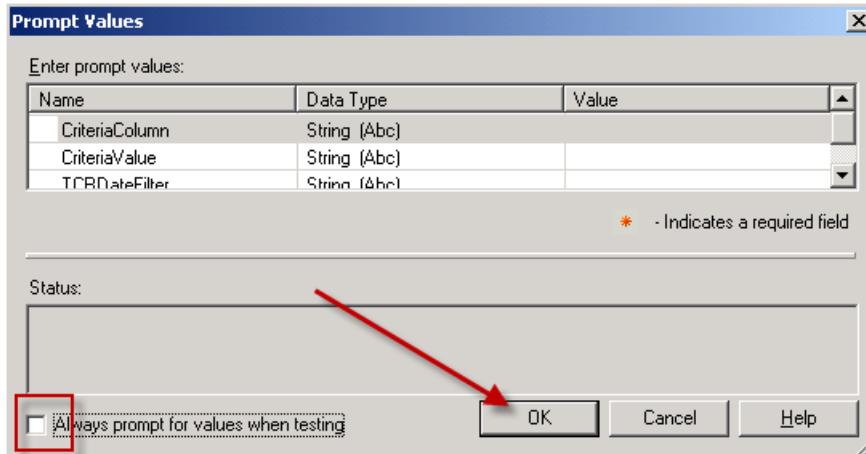
Note: You can drag the columns anywhere to the right. They are added on the end of the list.

Before completing and publishing the model to Tivoli Common Reporting, you perform some basic verification and tests.

6. In the Query Subject Definition - Events window, select the **Test** tab to display the test window, and click **Test Sample**



7. Clear the option **Always prompt for values when testing** and click **OK**.



8. In the Test Results pane, scroll to the right and bottom to locate the newly added fields to verify that these fields are being populated.

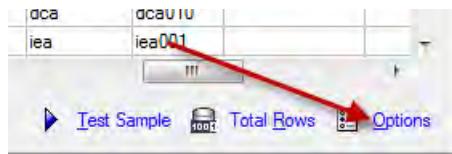
Test results						
	Last Occurrence	Deleted At	Tally	AGENCYID	SITEID	CONTACTNAME
	Jan 5, 2015 7:10:06 PM		3	oaca	oaca003	
	Jan 5, 2015 7:09:34 PM		3	bma	bma003	
	Jan 5, 2015 7:10:34 PM		3	dca	dca070	
	Jan 5, 2015 7:10:39 PM		3	bma	bma002	
	Jan 5, 2015 7:08:31 PM		1	dca	dca040	
	Jan 5, 2015 7:10:19 PM		3	oaca	oaca003	
	Jan 5, 2015 7:08:30 PM		1	UNKNOWN	UNKNOWI	
	Jan 5, 2015 7:10:15 PM		2	bma	bma001	
	Jan 5, 2015 7:09:19 PM		1	dca	dca010	
	Jan 5, 2015 7:09:39 PM		1	iea	iea001	

The Test Sample feature is configured to retrieve 25 rows from the database.



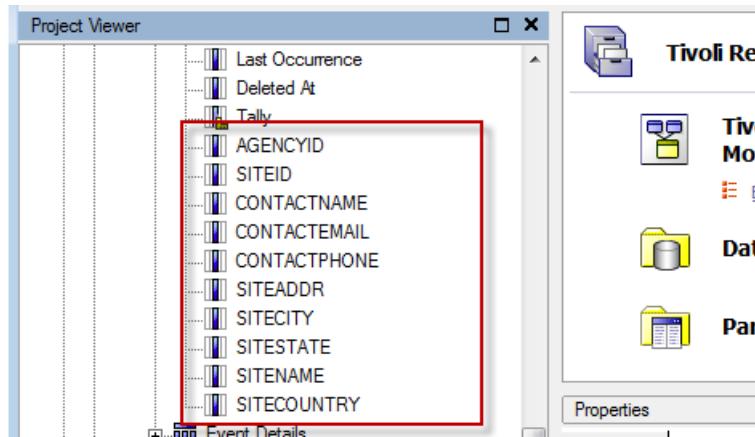
Important: The values for the contact information are not populated.

9. If the columns are not populated, click **Options**, and increase the number of rows to retrieve from the database.



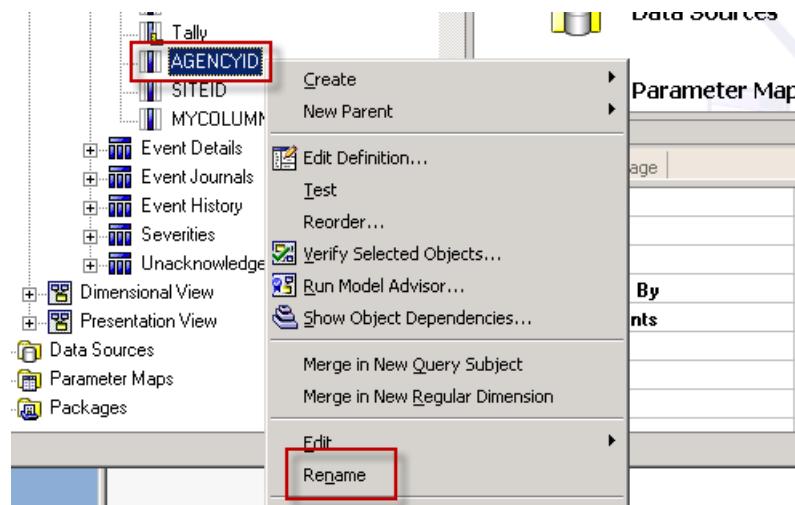
10. After testing is complete, click **OK** to keep the changes.

The Query update window opens and updates the changes in the Consolidation view.



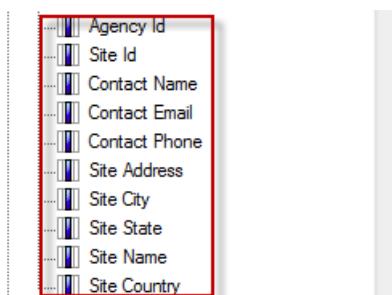
The Consolidation View is designed to provide a user-friendly representation of the data to report designers. One way to make the information more friendly is to change the column names to something that is easy to understand.

- In the **Netcool/OMNIbus (Consolidation) > Events** object, right-click the **AGENCYID** column, and select the **Rename** option from the menu as shown:



- In the column text area, rename the **AGENCYID** column to **Agency ID**.

- Repeat the steps for the other columns.



Presentation view modifications

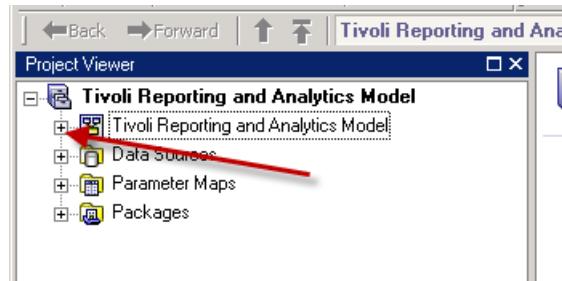
No modifications to the Netcool/OMNibus Presentation view are required for this specific column addition. Because you did not modify the Presentation view query subject (table) shortcuts to the Netcool/OMNibus Consolidation view, the original shortcuts are still valid.

The last step is to publish the extended model.

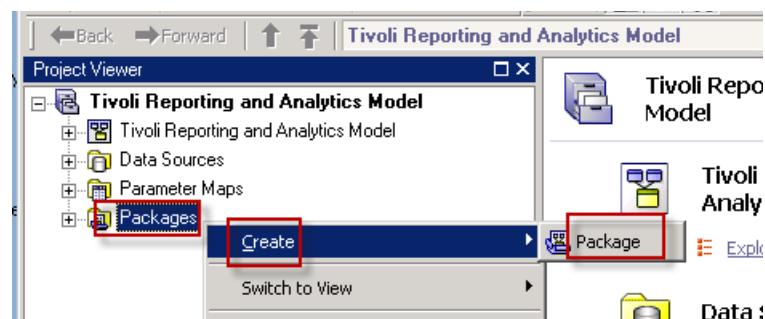
Publishing the updated model

In this exercise, you publish the model to the Cognos Content store. The publish action makes the package available to the user in Tivoli Common Reporting.

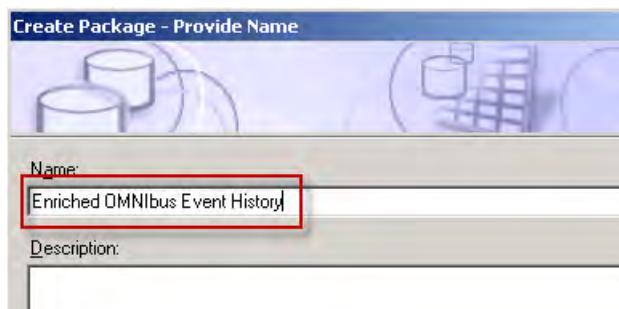
1. In the Project Viewer, click the **plus sign** on the **Tivoli Reporting and Analytics Model > Consolidation View** to collapse the view.



2. Right-click **Packages**, and select **Create > Package**.



3. Enter the package name, **Enriched OMNibus Event History**, and click **Next**.

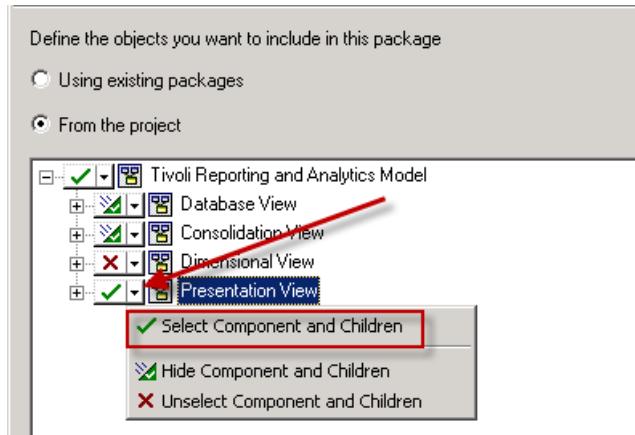




Hint: This name appears inside of the Tivoli Common Reporting interface. Be sure to provide a name that is descriptive and that a user can recognize. Do not use the original name to avoid overwriting any existing package.

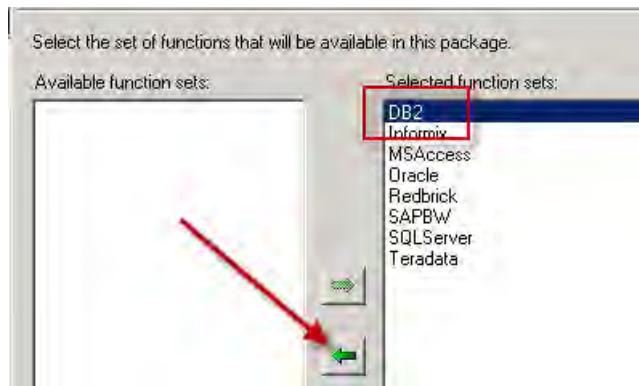
In the next few steps, you configure which views are included in the package. The individual who creates the data model can control which data elements are included in the report package.

4. In the left-menu of each view, use the following options to select the view's inclusion and display mode for the package:
 - a. For the Database view, select the **Hide Component and Children** option. The view is hidden from the user in the report authoring tool but is included in Cognos data model package.
 - b. For the Consolidation view, select the **Hide Component and Children** option. The view is hidden from the user in the report authoring tool but is included in Cognos data model package.
 - c. For the Dimensional view, select the **Unselect the Component and Children**. Because this view was not used in the data model, it should not be in the package or visible to the user.
 - d. For the Presentation view, leave the selection as **Select Component and Children**. The Presentation view is the view that is visible to the user in the report authoring tools.

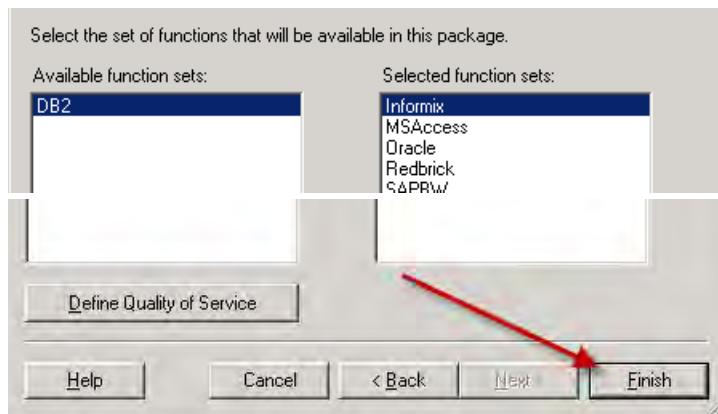


5. Click **Next**.

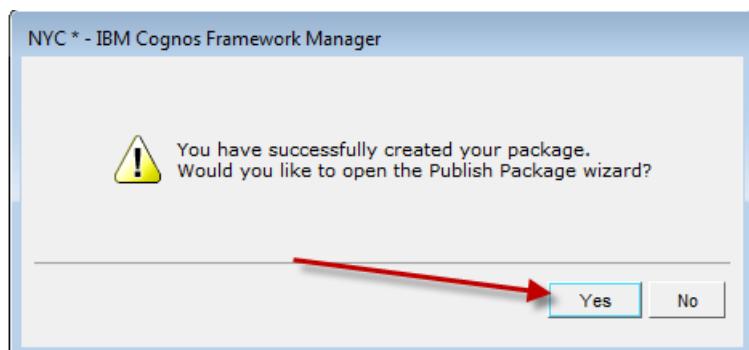
6. Select **DB2** from the list of **Selected functions sets** and click the left arrow to select it and add it to the **Available function sets** list.



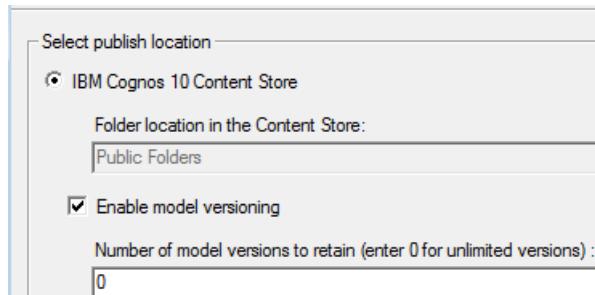
7. Click **Finish**.



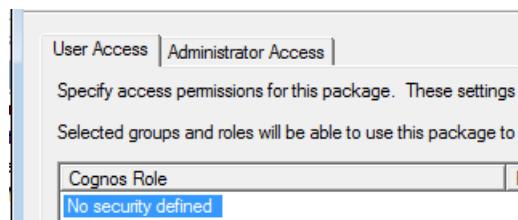
8. Select **Yes** to open the Publish Package wizard.



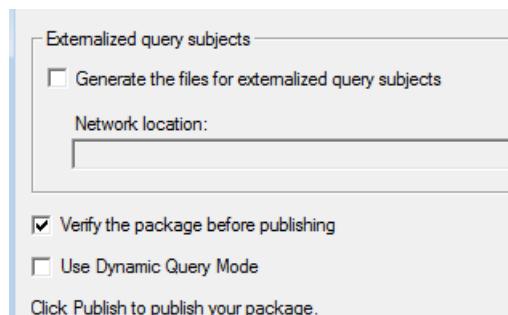
9. Accept all the default settings and click **Next**.



10. Leave the security settings as they are (No security defined), and click **Next**.

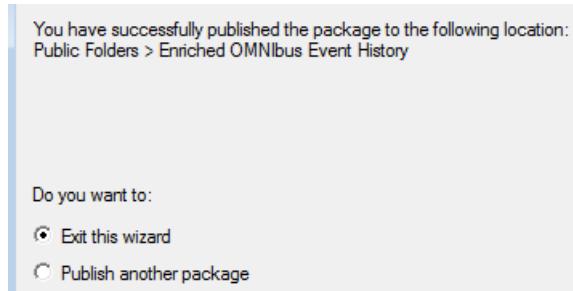


11. Leave the option settings as they are, and click **Publish**.



Hint: Verify the package before publishing so that you can find, and remove any invalid objects that can cause a query to fail.

12. Verify whether the package was published successfully, and click **Finish** to exit the wizard.



13. Select **File > Save**, and **File > Exit**.

The model is created and published to Tivoli Common Reporting. The model is also saved on the Windows image in the NYC folder.

All the steps to modify the Cognos data model for use in Tivoli Common Reporting report authoring tools are complete. Now you can access the published model in Tivoli Common Reporting and create a basic report to test the changes.

Exercise 6. Accessing and testing the new model

Before providing the model to the report writer, you must test the model. In this set of exercises, you access the new model and run a test report.

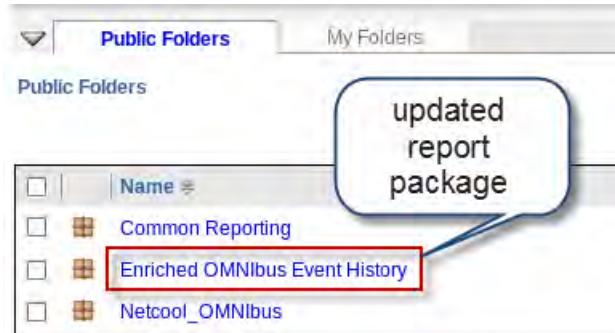
Accessing the new package in Tivoli Common Reporting



Important: You perform the following exercise on the **host2** image.

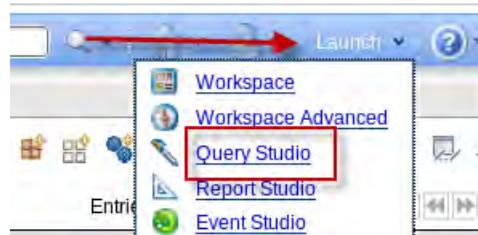
1. Return to the **host2** image.
2. Open a Firefox browser if necessary.
3. Log in as **ncoadmin** with password **object00**.
4. Click the icon, and select **Common Reporting**.



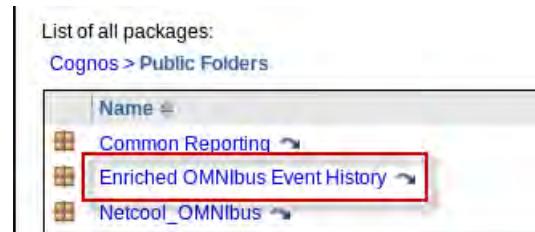


The **Enriched OMNibus Event History** entry is the package that you previously created, modified in Framework Manager, and published to Tivoli Common Reporting. The next series of steps use this package to generate a simple Cognos report with the Query Studio authoring tool.

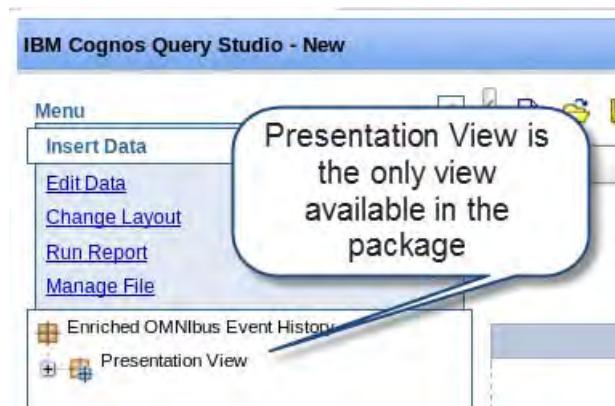
5. Click **Launch > Query Studio**.



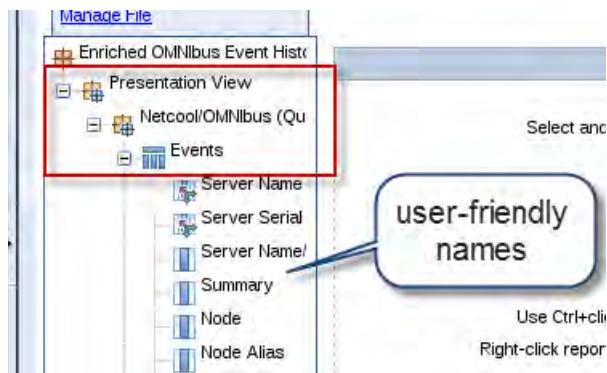
6. In the Select a Package window, select the **Enriched OMNibus Event History** package.



The window refreshes and shows the Cognos Query Studio report authoring tool with the Netcool/OMNibus Enriched OMNibus Event History package.



7. Expand the **Enriched Omnibus Event History > Presentation View > Netcool/OMNIBus Query > Events** object to display the fields for this table.

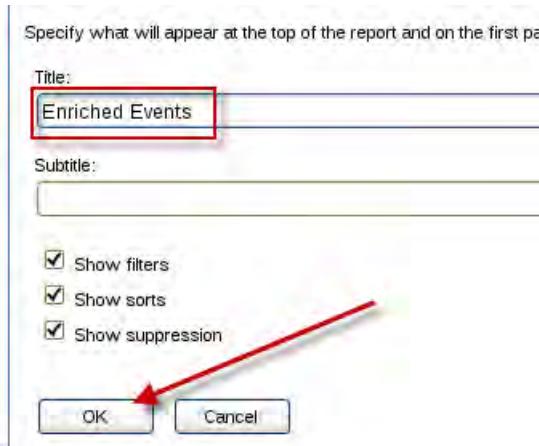


8. Click the **Title** link to edit the report title.



Hint: In Query Studio, report edits are displayed at the bottom of the current report view pane. You can scroll down to the bottom of the report view area to find the edit section.

9. Enter the title, **Enriched Events**, and click **OK** to complete the edit.



10. Double-click the **Events > Node** query item to add it to the report work area as shown:

A screenshot of the Query Studio interface. On the left, there's a tree view of query items under 'Enriched OMNIbus Event History'. A red box highlights the 'Node' item under 'Events'. A red arrow points from this item towards the right pane. The right pane shows a table titled 'Node' with several rows of data.

Node
2605-ce1-s.bma.gov
2605-ce1-s.iea.gov
2605-ce1-s.oaca.gov
2605-ce2-s.bma.gov
2605-ce2-s.oaca.gov
3620-ce1-s.bma.gov
3620-ce2-s.iea.gov
3620-ce2-s.oaca.gov
BRU-CPE-dca.go



Hint: You can also drag a column to add it to the report.

In Query Studio, the database table, and query item are automatically queried, and the data values are returned to populate the work area column.

11. Repeat the previous step to add the following query items (columns). Use the vertical scroll bar to locate the fields.

- **Last Occurrence**
- **Agency Id**
- **Site Id**
- **Site Name**

When complete, the view should look similar to the following screen capture:

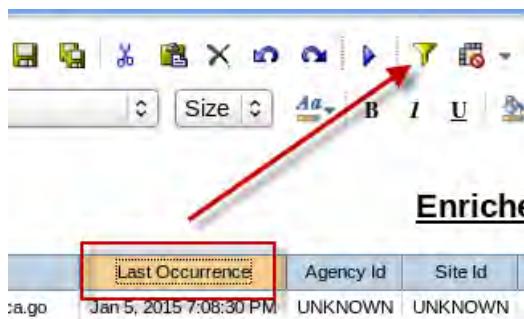
Enriched Events				
Node	Last Occurrence	Agency Id	Site Id	Site Name
BRU-CPE-dca.go	Jan 5, 2015 7:08:30 PM	UNKNOWN	UNKNOWN	UNKNOWN
3620-ce1-s.bma.gov	Jan 5, 2015 7:10:15 PM	bma	bma001	Sydney Building
MOS-dca.gov	Jan 5, 2015 7:10:34 PM	dca	dca070	Moscow
host1	Jan 5, 2015 7:11:04 PM			
host1.tivoli.edu	Jan 5, 2015 7:11:04 PM			
BRU-PE1-dca.gov	Jan 5, 2015 7:11:46 PM	dca	dca010	Brussels
2605-ce1-s.bma.gov	Jan 5, 2015 7:12:44 PM	bma	bma002	Spade Building
3620-ce2-s.oaca.gov	Jan 5, 2015 7:12:55 PM	oaca	oaca001	Bergman Building
NYC-dca.gov	Jan 5, 2015 7:13:34 PM	dca	dca040	New York City
BRU-CPE-dca.go	Jan 5, 2015 7:14:02 PM	UNKNOWN	UNKNOWN	UNKNOWN
BRU-PE1-dca.gov	Jan 5, 2015 7:15:19 PM	dca	dca010	Brussels
3620-ce2-s.iea.gov	Jan 5, 2015 7:15:55 PM	iea	iea001	Robinson Building
host1.tivoli.edu	Jan 5, 2015 7:15:56 PM			
MOS-dca.gov	Jan 5, 2015 7:16:01 PM	dca	dca070	Moscow

The report results might produce a number of blank rows. There are several methods to find the data that is being populated with the enriched event fields.

- Option 1: Use the **Page down** or **Bottom** links at the bottom of the report work area to find rows with values. This is available only if there is sufficient data in the report to fill more than one page.



- Option 2: Create a filter to display only recent results by using the following steps:
 - Select the **Last Occurrence** column header, and click the *funnel icon*.



- Scroll to the bottom of the report work area. Select **Last number of days**. Enter **0** for the value. Click **OK**.

Filter (Type in values)	
Filter on:	Last Occurrence
Operation:	<input type="text" value="Last number of days"/> <input type="button" value="?"/>
Condition:	<input type="button" value="Show only the following"/> <input type="button" value="?"/>
Number of days before today:	<input type="text" value="0"/> <input type="button" value="?"/>

Observe the results. The results look similar to this screen capture.

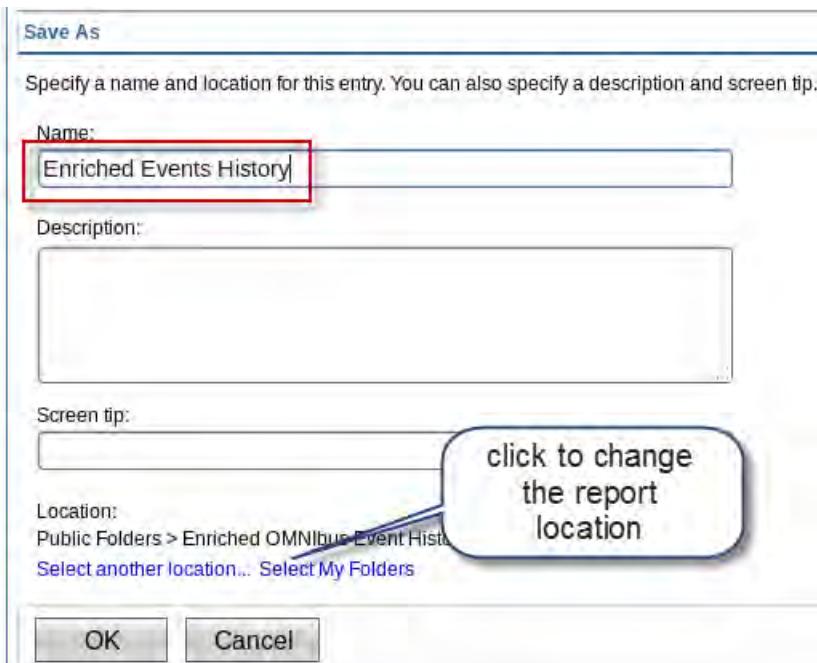
Node	Last Occurrence	Agency Id	Site Id	Site Name
nsm123-c3.iea.gov	Jan 5, 2015 8:01:04 PM			
2605-ce2-s.bma.gov	Jan 5, 2015 7:19:19 PM	bma		
2605-ce2-s.bma.gov	Jan 5, 2015 7:30:17 PM	bma		
2605-ce2-s.bma.gov	Jan 5, 2015 7:37:37 PM	bma		
2605-ce2-s.bma.gov	Jan 5, 2015 7:40:59 PM	bma	bma003	Astor Building
2605-ce2-s.bma.gov	Jan 5, 2015 7:52:34 PM	bma	bma003	Astor Building
2605-ce2-s.bma.gov	Jan 5, 2015 8:03:44 PM	bma	bma003	Astor Building
2605-ce1-s.iea.gov	Jan 5, 2015 7:20:18 PM	iea	iea002	Bacall Building

Any entries for **Last Occurrence**, older than 0 days before today, no longer show. The work area shows the current filter settings. You might still need to scroll or use the **Bottom** link to access the events with Agency Id and Site Id values.

Now that the report is complete, you can save the report.

12. Select the **Manage File > Save** menu option.

13. In the Save As area, type the name, **Enriched Events History**, and click **OK**.



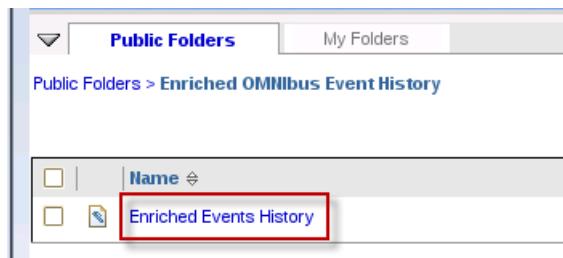
14. Click the *Home* icon to exit Query Studio.



15. Click **Enriched Omnibus Event History** to open the report package.



16. Click **Enriched Event History** to run the report.



17. There are no required prompt values. Scroll to the bottom of the page and click **OK**.

The screenshot shows a 'Prompt' dialog box. At the top, it says 'Provide values for the report you are about to run.' Below this are three sections: 'TCRDateFilter' (with a 'Provide a value:' label and an empty input field), 'CriteriaColumn' (with a 'Provide a value:' label and an empty input field), and 'CriteriaValue' (with a 'Provide a value:' label and an empty input field). Each section has a small descriptive text above its label.

Observe the results. Use the **Page Down** or **Bottom** link to access the most recent events.

<u>Enriched Events</u>				
Node	Last Occurrence	Agency Id	Site Id	Site Name
nsm123-c3.iea.gov	Jan 5, 2015 8:01:04 PM			
2605-ce2-s.bma.gov	Jan 5, 2015 7:19:19 PM	bma	bma003	Astor Building
2605-ce2-s.bma.gov	Jan 5, 2015 7:30:17 PM	bma	bma003	Astor Building
2605-ce2-s.bma.gov	Jan 5, 2015 7:37:37 PM	bma	bma003	Astor Building
2605-ce2-s.bma.gov	Jan 5, 2015 7:40:59 PM	bma	bma003	Astor Building
2605-ce2-s.bma.gov	Jan 5, 2015 7:52:34 PM	bma	bma003	Astor Building
2605-ce2-s.bma.gov	Jan 5, 2015 8:03:44 PM	bma	bma003	Astor Building
2605-ce1-s.iea.gov	Jan 5, 2015 7:20:18 PM	iea	iea002	Bacall Building
2605-ce1-s.iea.gov	Jan 5, 2015 7:25:09 PM	iea	iea002	Bacall Building
2605-ce1-s.iea.gov	Jan 5, 2015 7:34:17 PM	iea	iea002	Bacall Building
2605-ce1-s.iea.gov	Jan 5, 2015 7:44:43 PM	iea	iea002	Bacall Building
2605-ce1-s.iea.gov	Jan 5, 2015 7:57:28 PM	iea	iea002	Bacall Building
2605-ce1-s.iea.gov	Jan 5, 2015 8:02:11 PM	iea	iea002	Bacall Building

You successfully performed these tasks:

- Extended a Cognos data model to support new fields added to a Netcool/OMNIBus historical event database
- Published the new model package for use in Tivoli Common Reporting
- Created a report that uses the new event fields



8 Web GUI high availability exercises

In this exercise, you create a high availability load balanced cluster for Dashboard Application Services Hub. Dashboard Application Services Hub is installed on host2, and configured to use LDAP as the default user repository. Dashboard Application Services Hub is also installed on host1. However, it is not configured to use LDAP. In this exercise, you configure Dashboard Application Services Hub on host1 to use LDAP. You create a DB2 database to store the cluster configuration data. You create the cluster with the instance of Dashboard Application Services Hub that is running on host2. Next, you add the instance of Dashboard Application Services Hub from host1 to the cluster.

The complete details on how to configure Dashboard Application Services Hub for load balancing can be found in *Jazz for Service Management Version 1.1.0.3 Configuration Guide*.

The complete details on how to configure Netcool/OMNibus Web GUI for load balancing can be found in *Tivoli Netcool/OMNibus Version 8 Release 1 Installation and Deployment Guide*.



Important: You complete some of the steps in these exercises on host1, and you complete other steps on host2. Pay close attention to the instructions and make sure you complete the steps on the correct host.

Exercise 1 Configuring Dashboard Application Services Hub on host1

One of the requirements for the load balanced configuration is all Dashboard Application Services Hub servers in the cluster must be configured to use a common user repository. The copy on host2 is configured for LDAP, but the copy on host1 is not.

The user repository is defined in the Virtual Member Manager (VMM) component. The configuration for the Virtual Member Manager component is defined in an XML file. Save a copy of this file before you modify the existing configuration.

1. Switch to **host1**.
2. Open a Terminal window if none is open.

3. Save a copy of the VMM configuration file:

```
cd /opt/IBM/JazzSM/profile/config/cells/JazzSMNode01Cell/wim/config
cp wimconfig.xml /home/netcool
```



Important: If any of the following configuration steps fail, you can recover the original configuration by copying the saved file back to the original location, and restarting Dashboard Application Services Hub.

4. Start Dashboard Application Services Hub on host1.

```
cd /opt/IBM/JazzSM/profile/bin
./startServer.sh server1
```

```
ADMU0116I: Tool information is being logged in file
/opt/IBM/JazzSM/profile/logs/server1/startServer.log
ADMU0128I: Starting tool with the JazzSMProfile profile
ADMU3100I: Reading configuration for server: server1
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server server1 open for e-business; process id is 4595
```

5. Connect to WebSphere administrative console as follows:

- Open a Firefox browser and connect to Dashboard Application Services Hub on host1.
<http://host1.tivoli.edu:16310/ibm/console>

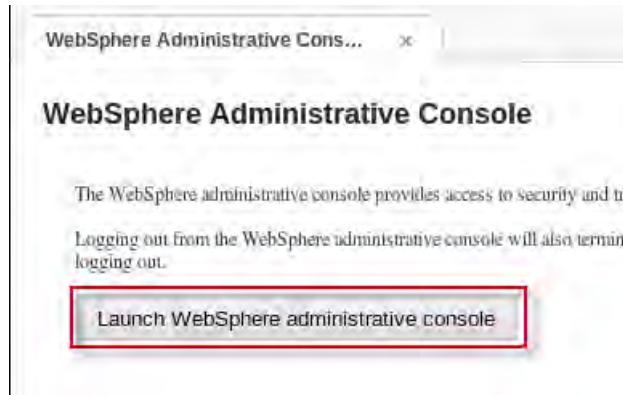


Important: The default Firefox home page is configured for Dashboard Application Services Hub on **host2**. You must change the URL to connect to host1.

- Log in as the **smadmin** user with password **object00**.
- Click the gear icon and select **WebSphere Administrative Console**.



- d. Click **Launch WebSphere administrative console**.



- e. Accept all security messages. The administrative console opens in a new Firefox tab.

6. Adding the LDAP directory as a user repository.

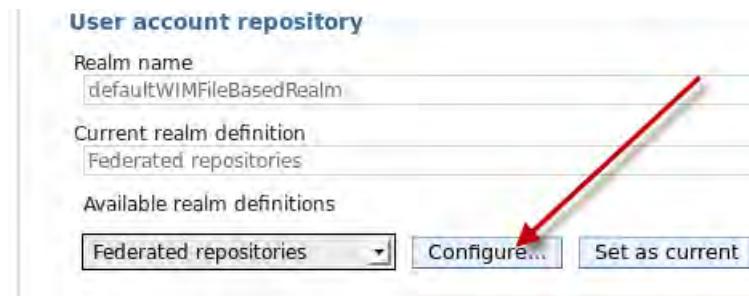


Important: The LDAP directory is on the **host1** image.

- a. Expand **Security** and click **Global Security**.



- b. Scroll down on the page to the *User account repository* section and click **Configure**.



- c. Scroll down on the page to the *Repositories in the realm*, and click **Add repositories**.

Select	Base Entry	Repository Identifier	Repository Type
<input type="checkbox"/>	o=defaultWIMFileBasedRealm	InternalFileRepository	File

Total 1

- d. Click **New Repository** and select **LDAP repository**.

General Properties

* Repository: none defined

* Unique distinguished name:

- New Repository... → (highlighted)
- LDAP repository (highlighted)
- Custom repository
- File repository

Distinguished name in the repository is different

- e. Change the repository name to **TIVIDS**.



Important: The repository name is relevant when you configure two copies of WebSphere to share a common LDAP for single sign-on. The repository name must be defined with the same text in each copy of WebSphere.

- f. Set the primary host name to **host1.tivoli.edu**.
- g. Verify that the port is set to **389**.
- h. Set the **Bind distinguished name** field to **cn=root**.
- i. Set the **Bind password** field to **object00**.
- j. Set the **Federated repository properties for login** field to **uid;cn**

k. Scroll to the bottom of the page and click **OK**.

General Properties

* Repository identifier
TIVIDS

Repository adapter class name
com.ibm.ws.wim.adapter.ldap.LdapAdapter

LDAP server

* Directory type
IBM Tivoli Directory Server

* Primary host name
host1.tivoli.edu

Port
389

Security

Bind distinguished name
cn=root

Bind password

Federated repository properties for login
uid;cn

l. Enter **dc=ibm,dc=com** for the Unique distinguished name field, and click **OK**

General Properties

* Repository
TIVIDS

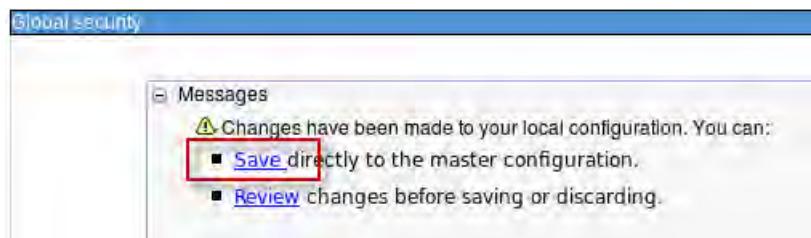
New Repository...

* Unique distinguished name of the base (or parent) entry in federated
dc=ibm,dc=com

Distinguished name in the repository is different

Apply OK Reset Cancel

m. Click **Save**.



Important: The base entry is mapped to the root of the LDAP directory. All operations are performed at root, which causes errors on most LDAP servers. More configuration is required.

The next step is to configure how the LDAP object names are mapped to each corresponding Virtual Member Manager resource type. You configure each resource type to specify search criteria. The search criteria is used to locate values for each of the object classes. These definitions essentially define the LDAP subtree where the Netcool user information is located.

7. Defining LDAP object class mappings.

- a. Scroll down on the page and click **TIVIDS**.

Repositories in the realm:

		Add repositories (LDAP, custom, etc)...	Use built-in repository	Remove
Select	Base Entry	Repository Identifier	Repository Type	
<input type="checkbox"/>	do=ibm,dc=com	TIVIDS	LDAP:IDS	
<input type="checkbox"/>	o=defaultWimFileBasedRealm	InternalFileRepository	File	

- b. Scroll down and click **Federated repositories entity types to LDAP object classes mapping**.

Additional Properties

- [Performance](#)
- [Federated repositories entity types to LDAP object classes mapping](#)
- [Federated repositories property names to LDAP attributes mapping](#)
- [Group attribute definition](#)



Important: The following steps are unique to the configuration of the LDAP server. The steps that are shown here are relevant to the LDAP configuration used for the class. The process is the same regardless of the LDAP configuration. It is the values that are used in these steps that must change for some other LDAP server.

- c. Click **Group**.

Select	Entity Type ▾	Object Classes ▾
You can administer the following resources:		
<input type="checkbox"/>	Group	groupOfNames
<input type="checkbox"/>	OrgContainer	organization;organizationalUnit;domain;container
<input type="checkbox"/>	PersonAccount	inetOrgPerson

- d. Enter **ou=tipgroups,cn=tipRealm,DC=IBM,DC=COM** for Search bases and click **OK**.

* Entity type

* Object classes

Search bases

Search filter

Apply **OK** **Reset** **Cancel**

e. Click OrgContainer.

Select	Entity Type	Object Classes
You can administer the following resources:		
<input type="checkbox"/>	Group	groupOfNames
<input type="checkbox"/>	OrgContainer	organization;organizationalUnit;domain;container
<input type="checkbox"/>	PersonAccount	inetOrgPerson

f. Verify that Search bases is empty and click OK.

* Entity type

* Object classes

Search bases

Search filter

g. Click PersonAccount.

Select	Entity Type	Object Classes
You can administer the following resources:		
<input type="checkbox"/>	Group	groupOfNames
<input type="checkbox"/>	OrgContainer	organization;organizationalUnit;domain;container
<input type="checkbox"/>	PersonAccount	inetOrgPerson

h. Enter **ou=tipusers,cn=tipRealm,DC=IBM,DC=COM** for the **Search bases** field and click OK.

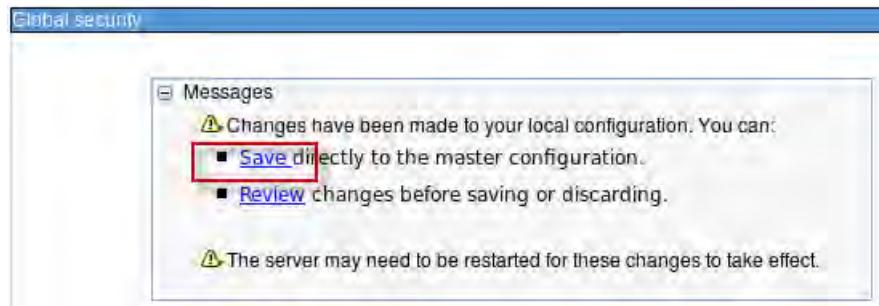
* Entity type

* Object classes

Search bases

Search filter

- Click **Save**.

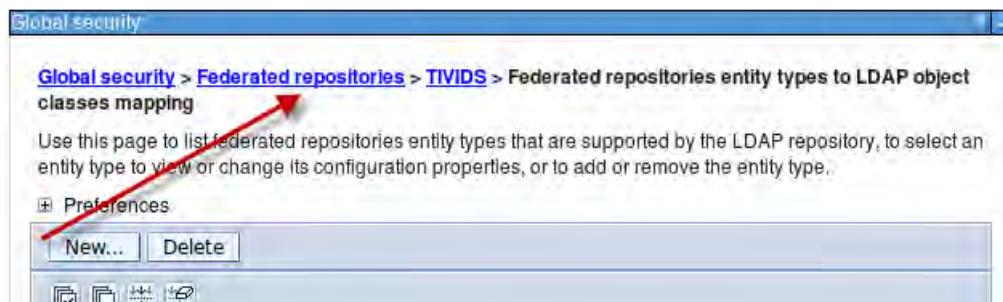


Now the Virtual Member Manager is configured to retrieve user information from a specific subtree within LDAP.

The last step is to configure Dashboard Application Services Hub to write new users and groups to the LDAP directory. This configuration process is similar to the previous steps. You define which LDAP object classes are modified when a new user or group is created.

- Configure IBM Dashboard Application Services Hub to write to LDAP as follows:

- Click **Federated repositories**.



- Scroll to the bottom of the page and click **Supported entity types**.

The screenshot shows the 'Supported entity types' section. It includes a table of repositories and a list of additional properties. The 'Additional Properties' list has a red box around the 'Supported entity types' link.

Repositories in the realm:			
Add repositories (LDAP, custom, etc...) Use built-in repository Remove		Select	Base Entry
		Repository Identifier	Repository Type
<input type="checkbox"/>	dc=ibm,dc=com	TIVIDS	LDAP:IDS
<input type="checkbox"/>	o=defaultWIMFileBasedRealm	InternalFileRepository	File
Total 2			

Additional Properties

- [Property extension repository](#)
- [Entry mapping repository](#)
- Supported entity types**
- [User repository attribute mapping](#)

Related Items

- [Manage repositories](#)
- [Trusted authentication realms - inbound](#)

c. Click Group.

Entity Type	Base Entry for the Default Parent	Relative Distinguished Name Properties
You can administer the following resources:		
Group	o=defaultWIMFileBasedRealm	cn
OrgContainer	o=defaultWIMFileBasedRealm	o;ou;dc;cn
PersonAccount	o=defaultWIMFileBasedRealm	uid



Important: Observe the values in the table that say `o=defaultWIMFileBasedRealm`. In the present state, if a new user is added to Dashboard Application Services Hub, the entry is written to the internal file-based repository.

d. Enter `ou=tipgroups,cn=tipRealm,DC=IBM,DC=COM` for Base entry for the default parent and click OK.

* Entity type
Group

* Base entry for the default parent
`ou=tipgroups,cn=tipRealm,DC=IBM,DC=COM`

* Relative Distinguished Name properties
cn

Apply OK Reset Cancel

e. Click OrgContainer.

Entity Type	Base Entry for the Default Parent	Relative Distinguished Name Properties
You can administer the following resources:		
Group	ou=tipgroups,cn=tipRealm,DC=IBM,DC=COM	cn
OrgContainer	o=netcoolObjectServerRepository	o;ou;dc;cn
PersonAccount	o=netcoolObjectServerRepository	uid

- f. Enter **dc=ibm,dc=com** for Base entry for the default parent and click **OK**.

* Entity type
OrgContainer

* Base entry for the default parent
dc=ibm,dc=com

* Relative Distinguished Name properties
o;ou;dc;cn

Apply OK Reset Cancel

- g. Click **PersonAccount**.

Entity Type	Base Entry for the Default Parent	Relative Distinguished Name properties
You can administer the following resources:		
Group	ou=tipgroups,cn=tipRealm,DC=IBM,DC=COM	cn
OrgContainer	dc=ibm,dc=com	o;ou;dc;cn
PersonAccount	o=netcool\ObjectServerRepository	uid

- h. Enter **ou=tipusers,cn=tipRealm,DC=IBM,DC=COM** for Base entry for the default parent and click **OK**.

* Entity type
PersonAccount

* Base entry for the default parent
ou=tipusers,cn=tipRealm,DC=IBM,DC=COM

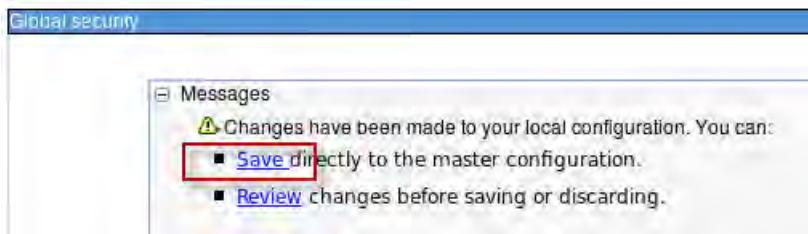
* Relative Distinguished Name properties
uid

Apply OK Reset Cancel

The revised entries are listed as shown.

Entity Type	Base Entry for the Default Parent	Relative Distinguished Name properties
You can administer the following resources:		
Group	ou=tipgroups,cn=tipRealm,DC=IBM,DC=COM	cn
OrgContainer	dc=ibm,dc=com	o;ou;dc;cn
PersonAccount	ou=tipusers,cn=tipRealm,DC=IBM,DC=COM	uid

- i. Click Save.



9. Log out of WebSphere administrative console.



10. Close the Firefox tab for the WebSphere administrative console.

11. Log out of IBM Dashboard Application Services Hub.

12. Restart Dashboard Application Services Hub.

- a. Stop Dashboard Application Services Hub.

```
cd /opt/IBM/JazzSM/profile/bin  
./stopServer.sh server1 -user smadmin -password object00
```

```
ADMU0116I: Tool information is being logged in file  
        /opt/IBM/JazzSM/profile/logs/server1/stopServer.log  
ADMU0128I: Starting tool with the JazzSMPProfile profile  
ADMU3100I: Reading configuration for server: server1  
ADMU3201I: Server stop request issued. Waiting for stop status.  
ADMU4000I: Server server1 stop completed.
```

- b. Start Dashboard Application Services Hub.

```
./startServer.sh server1
```

```
ADMU0116I: Tool information is being logged in file  
        /opt/IBM/JazzSM/profile/logs/server1/startServer.log  
ADMU0128I: Starting tool with the JazzSMPProfile profile  
ADMU3100I: Reading configuration for server: server1  
ADMU3201I: Server launched. Waiting for initialization status.  
ADMU3000I: Server server1 open for e-business; process id is 6653
```

Dashboard Application Services Hub is now configured with two user repositories: internal file-based, and LDAP. The LDAP users, and groups that are located within the defined subtree are available within Dashboard Application Services Hub.

13. Verify that the LDAP users are available within Dashboard Application Services Hub.
- Log in as the **smadmin** user with password **object00**.
 - Click the gear icon and select **WebSphere Administrative Console**.



- Click **Launch WebSphere administrative console**.



- Expand **Users and Groups** and click **Manage Users**.



- e. Observe the list of users.

Select	User ID	First name	Last name	E-mail	Unique Name
<input type="checkbox"/>	abraman	Ariana	Braman	abraman@ibm.com	cn=Ariana Braman,ou=tipusers,cn=tipRealm,DC=IBM,DC=COM
<input type="checkbox"/>	adurling	Adeline	Durling	adurling@ibm.com	cn=Adeline Durling,ou=tipusers,cn=tipRealm,DC=IBM,DC=COM
<input type="checkbox"/>	bwinebarger	Bart	Winebarger	bwinebarger@ibm.com	cn=Bart Winebarger,ou=tipusers,cn=tipRealm,DC=IBM,DC=COM
<input type="checkbox"/>	dselan	Dick	Selan	dselan@ibm.com	cn=Dick Selan,ou=tipusers,cn=tipRealm,DC=IBM,DC=COM
<input type="checkbox"/>	eange	Earline	Ange	eange@ibm.com	cn=Earline Ange,ou=tipusers,cn=tipRealm,DC=IBM,DC=COM
<input type="checkbox"/>	elotempio	Emelda	Lotempio	elotempio@ibm.com	cn=Emelda Lotempio,ou=tipusers,cn=tipRealm,DC=IBM,DC=COM
<input type="checkbox"/>	ezegarelli	Else	Zegarelli	ezegarelli@ibm.com	cn=Else Zegarelli,ou=tipusers,cn=tipRealm,DC=IBM,DC=COM
<input type="checkbox"/>	gbaillo	Gerald	Baillo	gbaillo@ibm.com	cn=Gerald Baillo,ou=tipusers,cn=tipRealm,DC=IBM,DC=COM
<input type="checkbox"/>	hdold	Houston	Dold	hdold@ibm.com	cn=Houston Dold,ou=tipusers,cn=tipRealm,DC=IBM,DC=COM
<input type="checkbox"/>	jguglielmo	Jasper	Guglielmo	jguglielmo@ibm.com	cn=Jasper Guglielmo,ou=tipusers,cn=tipRealm,DC=IBM,DC=COM
<input type="checkbox"/>	jmulberry	Jorge	Mulberry	jmulberry@ibm.com	cn=Jorge Mulberry,ou=tipusers,cn=tipRealm,DC=IBM,DC=COM
<input type="checkbox"/>	jsmith	Jane	Smith		uid=jsmith,ou=tipusers,cn=tipRealm,DC=IBM,DC=COM
<input type="checkbox"/>	jvasso	Jaime	Vasso	jvasso@ibm.com	cn=Jaime Vasso,ou=tipusers,cn=tipRealm,DC=IBM,DC=COM
<input type="checkbox"/>		Keesha			cn=Keesha

Dashboard Application Services Hub is now aware of 29 users. Note the values in the Unique Name column of the table. These values indicate that the user is defined in the LDAP directory. When one of these users logs in to Dashboard Application Services Hub, the Virtual Member Manager component uses the password that is defined in LDAP to authenticate the login.

14. Log out of WebSphere administrative console.

15. Log out of Dashboard Application Services Hub.

Configuring Dashboard Application Services Hub to allow logins if LDAP is down

Dashboard Application Services Hub is configured to use two user repositories:

internal file-based
LDAP

Dashboard Application Services Hub is based on WebSphere. There is a WebSphere property called `allowOperationIfReposDown`. The default setting for this property is false. When set to false, if one of the repositories is unavailable, users cannot log in to Dashboard Application Services Hub. If the property is true, and the LDAP server goes down, you can log in to Dashboard Application Services Hub as the `smadmin` user because that user is defined in the file-based repository.

To facilitate this exercise, a script is provided which runs a utility to change the value of the property to true.

1. Change directory to the location of the supplied script:

```
cd /workshop/unit08/dash
```

2. Examine the script as follows:

```
more wsadmin.sh
```

```
#!/bin/sh
#
# This script configures DASH to allow logins if not all
# repositories are available.
# This is required in order to use LDAP
#
# This script runs the wsadmin utility and passes it a jython command file
cd /opt/IBM/JazzSM/profile/bin
./wsadmin.sh -lang jython -user smadmin -password object00 -f
/workshop/unit08/dash/ldap.py
```

```
echo "Restart DASH to activate the changes"
```

This script calls the utility wsadmin.sh and passes the name of a Jython file.

3. Examine the Jython file as follows:

```
more ldap.py
```

```
AdminTask.updateIdMgrRealm ('[-name defaultWIMFileBasedRealm
-allowOperationIfReposDown true]')
AdminConfig.save()
```

The first line in the file contains the command sequence to change the property to true. The second line contains the command to save the revised property setting.

4. Run the script as follows:

```
./wsadmin.sh
```

```
WASX7209I: Connected to process "server1" on node JazzSMNode01 using SOAP
connector; The type of process is: UnManagedProcess
Restart DASH to activate the changes
```



Important: Dashboard Application Services Hub must be running to run this utility.

5. Restart Dashboard Application Services Hub.

a. Stop Dashboard Application Services Hub.

```
cd /opt/IBM/JazzSM/profile/bin  
./stopServer.sh server1 -user smadmin -password object00
```

```
ADMU0116I: Tool information is being logged in file  
          /opt/IBM/JazzSM/profile/logs/server1/stopServer.log  
ADMU0128I: Starting tool with the JazzSMPProfile profile  
ADMU3100I: Reading configuration for server: server1  
ADMU3201I: Server stop request issued. Waiting for stop status.  
ADMU4000I: Server server1 stop completed.
```

b. Start Dashboard Application Services Hub.

```
./startServer.sh server1
```

```
ADMU0116I: Tool information is being logged in file  
          /opt/IBM/JazzSM/profile/logs/server1/startServer.log  
ADMU0128I: Starting tool with the JazzSMPProfile profile  
ADMU3100I: Reading configuration for server: server1  
ADMU3200I: Server launched. Waiting for initialization status.  
ADMU3000I: Server server1 open for e-business; process id is 6653
```

To verify that the change works, you must temporarily stop the LDAP server on host1.

6. Stop the LDAP server as follows:

a. Change to the **root** user.

```
su -  
Password: object00
```

b. Stop LDAP.

```
/etc/init.d/ibmslapd stop
```

```
Stopping SDS instance dsrdbm01 Stopping SDS Admin Server instance dsrdbm01  
[root]
```



Important: Leave the terminal window as is. You return shortly and use it to restart the LDAP server.

7. Verify that you can log in to Dashboard Application Services Hub.

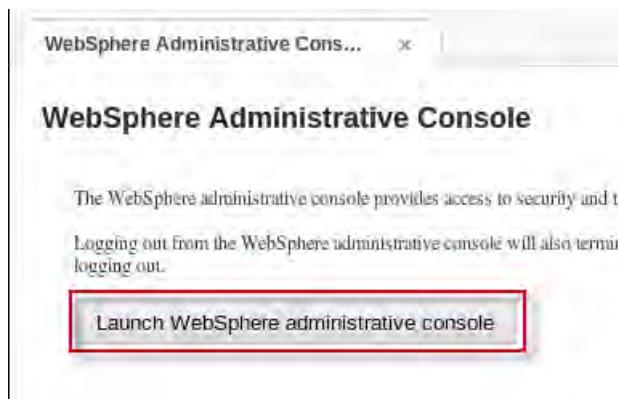
a. Log in as the **smadmin** user with password **object00**.

The successful login verifies that the property change is correct.

- b. Click the gear icon and select **WebSphere Administrative Console**.



- c. Click **Launch WebSphere administrative console**.



- d. Expand **Users and Groups** and click **Manage Users**.



- Observe the list of users.

1 users matched the search criteria.

Select	User ID	First name	Last name	E-mail	Unique Name
<input type="checkbox"/>	smadmin	smadmin	smadmin		uid=smadmin,o=defaultWIMFileBasedRealm

Page 1 of 1 Total: 1

Dashboard Application Services Hub is aware of only one user: **smadmin**.



Important: Leave the browser session as is. You return to it shortly.

- Restart the LDAP server as follows:

```
/etc/init.d/ibmslapd start
```

```
Starting SDS instance dsrdbm01 Starting SDS Admin Server instance dsrdbm01
[root]
```

- Exit the **root** user and return to the **netcool** user.

```
exit
```

- Return to the administrative console session and click **Search**.

29 users matched the search criteria.

Select	User ID	First name	Last name	E-mail	Unique Name
<input type="checkbox"/>	abraman	Ariana	Braman	abraman@ibm.com	cn=Ariana Braman,ou=tipusers,ou=tipRealm,DC=IBM,DC=COM
<input type="checkbox"/>	adurling	Adeline	Durling	adurling@ibm.com	cn=Adeline Durling,ou=tipusers,ou=tipRealm,DC=IBM,DC=COM
<input type="checkbox"/>	bwinebarger	Bart	Winebarger	bwinebarger@ibm.com	cn=Bart Winebarger,ou=tipusers,ou=tipRealm,DC=IBM,DC=COM
<input type="checkbox"/>	dselan	Dick	Selan	dselan@ibm.com	cn=Dick Selan,ou=tipusers,ou=tipRealm,DC=IBM,DC=COM
<input type="checkbox"/>	eange	Earline	Ange	eange@ibm.com	cn=Earline Ange,ou=tipusers,ou=tipRealm,DC=IBM,DC=COM
<input type="checkbox"/>	elotempio	Emelda	Lotempio	elotempio@ibm.com	cn=Emelda Lotempio,ou=tipusers,ou=tipRealm,DC=IBM,DC=COM
<input type="checkbox"/>	ezegarelli	Else	Zegarelli	ezegarelli@ibm.com	cn=Else Zegarelli,ou=tipusers,ou=tipRealm,DC=IBM,DC=COM
<input type="checkbox"/>	gbaillo	Gerald	Baillo	gbaillo@ibm.com	cn=Gerald Baillo,ou=tipusers,ou=tipRealm,DC=IBM,DC=COM
<input type="checkbox"/>	hdold	Houston	Dold	hdold@ibm.com	cn=Houston Dold,ou=tipusers,ou=tipRealm,DC=IBM,DC=COM
<input type="checkbox"/>	jguglielmo	Jasper	Guglielmo	jguglielmo@ibm.com	cn=Jasper Guglielmo,ou=tipusers,ou=tipRealm,DC=IBM,DC=COM
<input type="checkbox"/>	jmulberry	Jorge	Mulberry	jmulberry@ibm.com	cn=Jorge Mulberry,ou=tipusers,ou=tipRealm,DC=IBM,DC=COM
<input type="checkbox"/>	jsmith	Jane			uid=jsmith,ou=tipusers,ou=tipRealm,DC=IBM,DC=COM

All 29 users are again available.

- Log out of WebSphere administrative console.

- Log out of Dashboard Application Services Hub.

Both copies of Dashboard Application Services Hub are now configured to use a common LDAP user repository.

Exercise 2 Creating the configuration database on host2

Configuration data for all nodes in a cluster is stored in a DB2 database. You must create the database manually. For the class configuration, you use the copy of DB2 installed on host2.

1. Log in to host2 as user **netcool** with password **object00**.

2. Open a Terminal window.

3. Change to the **db2inst1** user with password **object00**.

```
su - db2inst1  
Password: object00
```

4. Create a database that is called LBCONFIG with the supplied script.



Important: The database name that is used does not matter. It can be any valid DB2 database name.

```
cd /workshop/unit08/lb  
../create_db.sh
```

```
This script must be run as the db2inst1 user  
Create LBCONFIG database  
DROP DATABASE LBCONFIG  
SQL1013N  The database alias name or database name "LBCONFIG" could not be  
found.  SQLSTATE=42705  
  
CREATE DATABASE LBCONFIG COLLATE USING SYSTEM PAGESIZE 8192 USER TABLESPACE  
MANAGED BY SYSTEM USING ('LBCONFIG') EXTENTSIZE 64 PREFETCHSIZE 32 TEMPORARY  
TABLESPACE MANAGED BY SYSTEM USING ('LBCONFIGTemp') EXTENTSIZE 32 PREFETCHSIZE  
16  
DB20000I  The CREATE DATABASE command completed successfully.
```



Note: The SQL1013N message appears the first time that this script is run. The sql file contains a command to drop an existing database.

5. Verify access to the database by the **db2inst1** user.

```
db2 connect to LBCONFIG
```

Database Connection Information

```
Database server      = DB2/LINUXX8664 10.5.3
SQL authorization ID = DB2INST1
Local database alias = LBCONFIG
```

6. Exit the **db2inst1** user back to the **netcool** user.

```
exit
```

Exercise 3 Creating the cluster with host2

The cluster is created by adding the first node. When the first node is added, the database tables are created in the configuration store database for the first node. The tables are populated with the configuration information from the first node.

The copy of Dashboard Application Services Hub on host2 already contains customizations. The copy on host1 does not contain any customizations.

When you are creating a new load balanced cluster, you must first export all data from the stand-alone instance. Then, import the previously exported data after the cluster is set up. The first step in this exercise is to export the data from the copy of Dashboard Application Services Hub on **host2**.

1. Export the configuration from host2 as follows:

- a. Change to the target directory:

```
cd /opt/IBM/JazzSM/ui/bin
```

- b. Enter the following command to export the configuration:

```
./restcli.sh export -username smadmin -password object00 -destination
/tmp/data.zip
```

ATKRST200I The command completed successfully.



Important: This step exports the configuration of Dashboard Application Services Hub. It does not contain the configuration of Web GUI.

The next step is to modify a property file and configure parameters that are related to the cluster.

2. Modify the property file as shown here:

- a. Change to the target directory.

```
cd /opt/IBM/JazzSM/ui/bin/ha
```

- b. Save a copy of the file before modifications.

```
cp tipha.properties tipha.properties.orig
```

- c. Modify the file with gedit.

```
gedit tipha.properties
```

- d. Configure the DB2 parameters as follows:

```
DBHost=host2.tivoli.edu
```

```
DBPort=50000
```

```
DBName=LBCONFIG
```

```
#####
# Host name of DB2 server
# Example: DBHost=tipdb.ibm.com
#
#DBHost=host2.tivoli.edu

#####
# Port opened for DB connections
# Default: 50000
#
#DBPort=50000

#####
# Database name for TIP HA
# Example: DBName=tipdb
#
#DBName=LBCONFIG
```

- e. Scroll down in the file and modify the DB2 driver path as follows:

```
DBJDBCDriverPath=/opt/IBM/JazzSM/lib/db2
```

```
#####
# Directory of DB2 JDBC driver libraries
# Example: DBJDBCDriverPath=C:/IBM/tivoli/tip/universalDriver/lib
#
#DBJDBCDriverPath=/opt/IBM/JazzSM/lib/db2
```

- f. Scroll down in the file and change the local host parameters as follows:

```
LocalHost=host2.tivoli.edu
```

```
LocalPort=16311
```

```
#####
# Hostname of local TIP node
# Example: LocalHost=tip01.ibm.com
#
#LocalHost=host2.tivoli.edu

#####
# TIP SSL port
# Default value: 16311
#
#LocalPort=16311
```

- g. Scroll down in the file and change the WAS parameters as follows:

```
WasRoot=/opt/IBM/WebSphere/AppServer
TipHome=/opt/IBM/JazzSM/ui
ProfilePath=/opt/IBM/JazzSM/profile
ProfileName=JazzSMPProfile
CellName=JazzSMNode01Cell
NodeName=JazzSMNode01
```

```
#####
# WAS home path
# Example: WasRoot=C:/IBM/tivoli/was
#
WasRoot=/opt/IBM/WebSphere/AppServer

#####
# TIP home path
# Example: TipHome=C:/IBM/tivoli/tipHome
#
TipHome=/opt/IBM/JazzSM/ui

#####
# WAS profile path
# Example: ProfilePath=C:/IBM/tivoli/profile
#
ProfilePath=/opt/IBM/JazzSM/profile

#####
# WAS profile name
# Example: ProfileName=TIPPProfile
#
ProfileName=JazzSMPProfile

#####
# WAS cell name
# Example: CellName=TIPCell
#
CellName=JazzSMNode01Cell

#####
# WAS node name
# Example: NodeName=TIPNode
#
NodeName=JazzSMNode01
```

- h. Scroll down in the file and change the logging level as follows:

```
LoggerLevel=FINER
```

```
#####
#High Availability Logging
#Change the next line if you want to enable logging into file and console,
#
#LoggerLevel=OFF
LoggerLevel=FINER
```



Hint: The change to the logging level is not a requirement for load balancing. You change the value to provide detailed logging in case there are issues during the initial configuration process. After the cluster is functioning correctly, you can change the value back to OFF.

- i. Observe the value for *High Availability Status* on the end of the file:

```
#####
#High Availability Status
#Do not edit the value manually.
#It is used to decide whether HA is enabled or not
#
HAEenabled=false
```

The value of this parameter changes dynamically from false to true when the node is added to the cluster.

- j. Save the modified file and exit gedit.

A number of the values for the parameters in the property file can be found in a log file. The log file is created when Dashboard Application Services Hub is installed. The log file can be found here:

```
/opt/IBM/WebSphere/AppServer/logs/manageprofiles/JazzSMProfile_create.log
```

Scroll down in the log file and locate the section that is shown in the following screen capture:

```
<thread>0</thread>
<message>Current command line is: { "-profilePath", "/opt/IBM/JazzSM/profile",
"-enableService", "false", "-nodeName", "JazzSMNode01", "-applyPerfTuningSetting",
"standard", "-winServiceCheck", "false", "-portFile", "/opt/IBM/JazzSM/va
r/JazzSMProfile_portDef.properties", "-serverName", "server1", "-profileName",
"JazzSMProfile", "-create", "-cellName", "JazzSMNode01Cell", "-isDeveloperServer",
"-enableAdminSecurity", "true", "-adminUserName", "smadmin", "-hostName", "hos
t2.tivoli.edu", "-isDefault", "-adminPassword", "*****", "-templatePath", "/o
pt/IBM/WebSphere/AppServer/profileTemplates/default", "-omitAction", "samplesInst
allAndConfig", "defaultAppDeployAndConfig" }</message>
```

3. Stop Dashboard Application Services Hub on host2.

```
cd /opt/IBM/JazzSM/profile/bin
./stopServer.sh server1 -user smadmin -password object00
```

Wait for the server to stop.

4. Add Dashboard Application Services Hub on host2 to the cluster.

The following command adds the configuration information for Dashboard Application Services Hub on host2 to the DB2 database and creates the initial cluster.

```
cd /opt/IBM/JazzSM/ui/bin/ha
```

```
/opt/IBM/WebSphere/AppServer/bin/ws_ant.sh -f install.ant configHA
-Dusername=db2inst1 -Dpassword=object00 -DWAS_username=smadmin
-DWAS_password=object00
```



Note: The command runs for several minutes.

```
.
```

```
.
```

```
.
```

```
verifyStopServerTimeout:
```

```
join:
```

```
configHA:
```

```
BUILD SUCCESSFUL  
Total time: 7 minutes 7 seconds
```

5. Verify the current state of high availability.

```
cd /opt/IBM/JazzSM/ui/bin/ha  
tail tipha.properties
```

```
,LoggerLevel=FINER  
,#High Availability Status  
,#Do not edit the value manually.  
,#It is used to decide whether HA is enabled or not  
,#  
HAEnabled=true
```

6. Start Dashboard Application Services Hub.

```
cd /opt/IBM/JazzSM/profile/bin  
. ./startServer.sh server1
```

Wait for the server to start.

7. Verify the state of the cluster.

```
cd /opt/IBM/JazzSM/ui/bin/ha
./HATool.sh modules db2inst1 object00 -byNodes -showAll

com.ibm.net.SocketKeepAliveParameters
Node: host2.tivoli.edu:16311
    com.ibm.twl.ssd.Dashlets - 1.1.0.0
    development.export.sample.1346858222067.unique - 1.0.0
    com.ibm.tivoli.tip.ThirdPartyIntegrationWidget - 1.0.2
    com.ibm.tivoli.tip.utilportlets - 1.0.2
    export_import_portlet - 1.0.0
    com.ibm.isclite.ISCWire - 1.0.0
    com.ibm.tivoli.tip.RaveWidget - 1.0.2
    com.ibm.tivoli.tip.chart.TIPChartPortlet.ad48252c01 - 6.1
    com.ibm.tivoli.reporting.advanced.cognos.portlet - 3.1.0
    com.ibm.isclite.ISCAAdminPortlet - 3.1.0.3
    com.ibm.tivoli.tip.TIPWebWidgetPortlet.e5bd988b51 - 1.0.2
    com.ibm.tivoli.tip.changepasswd.TIPChangePasswd - 6.1.0
    com.ibm.tivoli.em.iscmodule - 8.1.0
    com.ibm.isclite.ISCAAdminPortlets - 3.1.0.3
```



Important: The HATool utility queries the cluster database. The user name and password are for DB2.

The cluster contains a single node, host2.tivoli.edu.

Exercise 4 Adding host1 to the cluster

The process to add host1 to the cluster is essentially the same as the process for host2, with a few exceptions. The copy of Dashboard Application services Hub on host1 does not contain any modifications so you do not have to export the configuration. Most of the changes to the ha property file are the same as for host2. You can retrieve a copy of the file from host2, and make minor changes to save time.

1. Switch to the **host1** image.
2. Use SFTP to retrieve a copy of the property file from host2 as follows:
 - a. Change to the target directory.

```
cd /opt/IBM/JazzSM/ui/bin/ha
```

- b. Save a copy of the file before modifications.

```
cp tipha.properties tipha.properties.orig
```

- c. Connect to host2 with SFTP.

```
sftp host2
```

Connecting to host2...

The authenticity of host 'host2 (192.168.100.161)' can't be established.

RSA key fingerprint is 1c:2c:83:be:ca:fd:a4:86:14:29:16:2f:76:65:af:55.

Are you sure you want to continue connecting (yes/no)?

- d. Enter yes.

Warning: Permanently added 'host2,192.168.100.161' (RSA) to the list of known hosts.

netcool@host2's password:

- e. Enter **object00** for the password.

```
sftp>
```

- f. Change to the target directory.

```
sftp> cd /opt/IBM/JazzSM/ui/bin/ha
```

- g. Retrieve the file.

```
sftp> get tipha.properties
```

Fetching /opt/IBM/JazzSM/ui/bin/ha/tipha.properties to tipha.properties

/opt/IBM/JazzSM/ui/bin/ha/tipha.properties 100% 5831 5.7KB/s 00:00

- h. Exit SFTP.

```
sftp> quit
```

One modification must be made to the copy of the host2 property file.

3. Modify the property file as shown here:

- a. Modify the file with gedit.

```
gedit tipha.properties
```

- b. Scroll down in the file and change the local host parameter as follows:

```
LocalHost=host1.tivoli.edu
```



- c. Save the modified file and exit gedit.

4. Stop Dashboard Application Services Hub on host1.

```
cd /opt/IBM/JazzSM/profile/bin  
./stopServer.sh server1 -user smadmin -password object00
```

Wait for the server to stop.

5. Add Dashboard Application Services Hub on host1 to the cluster.

The following command adds the configuration information for Dashboard Application Services Hub on host1 to the DB2 database and adds the node to the cluster.

```
cd /opt/IBM/JazzSM/ui/bin/ha
```

```
/opt/IBM/WebSphere/AppServer/bin/ws_ant.sh -f install.ant configHA  
-Dusername=db2inst1 -Dpassword=object00 -DWAS_username=smadmin  
-DWAS_password=object00
```



Note: The command runs for several minutes.

```
.  
. .  
install:
```

```
join:
```

```
[java] Join to HA system procedure commencing...  
[java] Join to HA system procedure finished successfully.  
[copy] Copying 1 file to  
/opt/IBM/JazzSM/profile/config/cells/JazzSMNode01Cell/applications/isc.ear/deployments/isc/isclite.war/WEB-INF
```

```
configHA:
```

```
BUILD SUCCESSFUL  
Total time: 4 minutes 0 seconds
```

6. Verify the current state of high availability.

```
cd /opt/IBM/JazzSM/ui/bin/ha  
tail tipha.properties
```

```
,LoggerLevel=FINER  
,#High Availability Status  
,#Do not edit the value manually.  
,#It is used to decide whether HA is enabled or not  
#  
HAEEnabled=true
```

7. Start Dashboard Application Services Hub.

```
cd /opt/IBM/JazzSM/profile/bin  
./startServer.sh server1
```

Wait for the server to start.

8. Verify the state of the cluster.

```
cd /opt/IBM/JazzSM/ui/bin/ha
./HATool.sh modules db2inst1 object00 -byNodes -showAll

com.ibm.net.SocketKeepAliveParameters
Node: host2.tivoli.edu:16311
    com.ibm.twl.ssd.Dashlets - 1.1.0.0
    development.export.sample.1346858222067.unique - 1.0.0
    com.ibm.tivoli.tip.ThirdPartyIntegrationWidget - 1.0.2
    com.ibm.tivoli.tip.utilportlets - 1.0.2
    export_import_portlet - 1.0.0
    com.ibm.isclite.ISCWire - 1.0.0
    com.ibm.tivoli.tip.RaveWidget - 1.0.2
    com.ibm.tivoli.tip.chart.TIPChartPortlet.ad48252c01 - 6.1
    com.ibm.tivoli.reporting.advanced.cognos.portlet - 3.1.0
    com.ibm.isclite.ISCAdminPortlet - 3.1.0.3
    com.ibm.tivoli.tip.TIPWebWidgetPortlet.e5bd988b51 - 1.0.2
    com.ibm.tivoli.tip.changepasswd.TIPChangePasswd - 6.1.0
    com.ibm.tivoli.em.iscmodule - 8.1.0
    com.ibm.isclite.ISCAdminPortlets - 3.1.0.3

Node: host1.tivoli.edu:16311
    com.ibm.twl.ssd.Dashlets - 1.1.0.0
    development.export.sample.1346858222067.unique - 1.0.0
    com.ibm.tivoli.tip.ThirdPartyIntegrationWidget - 1.0.2
    com.ibm.tivoli.tip.utilportlets - 1.0.2
    export_import_portlet - 1.0.0
    com.ibm.isclite.ISCWire - 1.0.0
    com.ibm.tivoli.tip.RaveWidget - 1.0.2
    com.ibm.tivoli.tip.chart.TIPChartPortlet.ad48252c01 - 6.1
    com.ibm.tivoli.reporting.advanced.cognos.portlet - 3.1.0
    com.ibm.isclite.ISCAdminPortlet - 3.1.0.3
    com.ibm.tivoli.tip.TIPWebWidgetPortlet.e5bd988b51 - 1.0.2
    com.ibm.tivoli.tip.changepasswd.TIPChangePasswd - 6.1.0
    com.ibm.tivoli.em.iscmodule - 8.1.0
    com.ibm.isclite.ISCAdminPortlets - 3.1.0.3
```

The cluster now contains two nodes: host2.tivoli.edu and host1.tivoli.edu.

The cluster configuration ensures that any change made on one node is replicated to the other nodes in the cluster. When a node enters an existing cluster, the node is not synchronized to the contents of the other nodes. The host1 node does not contain the same configuration as the host2 node, even though it belongs to the cluster. You must import the host2 configuration that was created previously. Retrieve the file that contains the export of the host2 configuration, and import the configuration into host1.

9. Retrieve the export file from host2 with SFTP.

- a. Change to the tmp directory.

```
cd /tmp
```

- b. Connect to host2 with SFTP.

```
sftp host2
Connecting to host2...
netcool@host2's password: object00
sftp>
```

- c. Change to the tmp directory on host2

```
sftp> cd /tmp
```

- d. Retrieve the file.

```
sftp> get data.zip
Fetching /tmp/data.zip to data.zip
/tmp/data.zip
```

100% 67KB 66.7KB/s 00:00

- e. Exit SFTP.

```
sftp> quit
```

10. Import the file.

```
cd /opt/IBM/JazzSM/ui/bin
./restcli.sh import -username smadmin -password object00 -source /tmp/data.zip
```

ATKRST200I The command completed successfully.

Exercise 5 Configuring server to server trust

The following steps configure server-to-server trust between the two servers. Server-to-server trust is a requirement of load balancing that enables the copies of Dashboard Application Services Hub to connect to each other and send notifications. These configuration steps must be performed on both VMware images.



Note: The changes to the ssl.client.props file described here are the same for both copies of Dashboard Application Services Hub. In the interest of time, it is more efficient to change the file on host1 and then send a copy by FTP to host2.

Configuring trust on host1

1. Modify the ssl.client.props file as shown here:

- a. Change to the target directory.

```
cd /opt/IBM/JazzSM/profile/properties
```

- b. Save a copy of the file before modifications.

```
cp ssl.client.props ssl.client.props.orig
```

- c. Modify the file with gedit.

```
gedit ssl.client.props
```

 Note: There are several places in the file that require modifications.

- d. Locate the section that is shown below and remove the # character from the first seven lines so that it looks like the following screen capture:

```
-----  
# Another SSL configuration (this is a template, uncomment and modify)  
# You can configure the dynamicSelectionInfo OR reference this alias  
# from another protocol (e.g., soap.client.props or sas.client.props)  
#-----  
com.ibm.ssl.alias=AnotherSSLSettings  
com.ibm.ssl.protocol=SSL_TLS  
com.ibm.ssl.securityLevel=HIGH  
com.ibm.ssl.trustManager=IbmX509  
com.ibm.ssl.keyManager=IbmX509  
com.ibm.ssl.contextProvider=IBMJSSE2  
com.ibm.ssl.enableSignerExchangePrompt=true  
#com.ibm.ssl.keyStoreClientAlias=default  
#com.ibm.ssl.customTrustManagers=  
#com.ibm.ssl.customKeyManager=  
#com.ibm.ssl.dynamicSelectionInfo=  
#com.ibm.ssl.enabledCipherSuites=
```

uncomment these lines

- e. Scroll down to the end of the file, locate the section that is shown below and remove the # character from all of the lines so that it looks like the following screen capture:

```
# TrustStore information  
com.ibm.ssl.trustStoreName=AnotherTrustStore  
com.ibm.ssl.trustStore=${user.root}/etc/trust.p12  
com.ibm.ssl.trustStorePassword={xor}CD09Hgw=  
com.ibm.ssl.trustStoreType=PKCS12  
com.ibm.ssl.trustStoreProvider=IBMJCE  
com.ibm.ssl.trustStoreFileBased=true  
com.ibm.ssl.trustStoreReadOnly=false
```

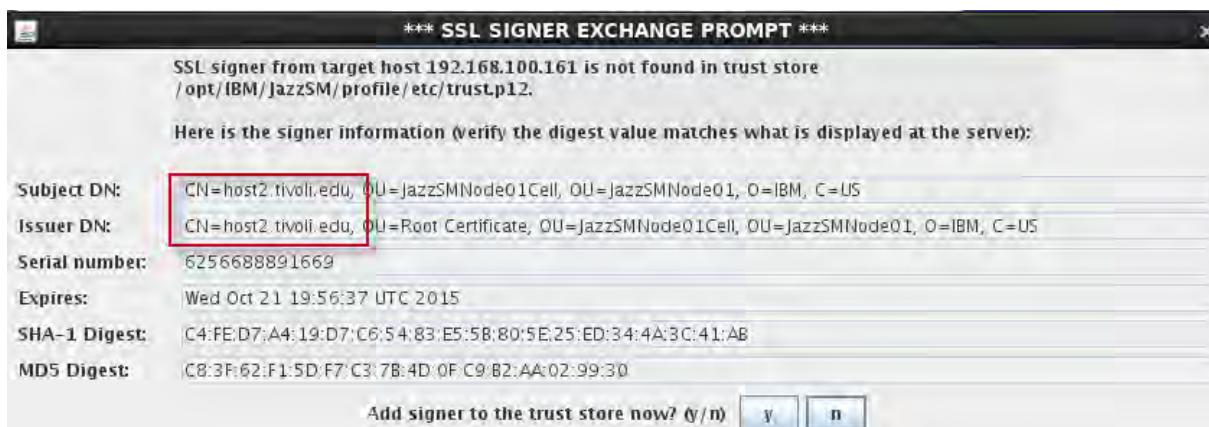
uncomment these lines

- f. Locate the second line in the *TrustStore* section and modify the line as follows:

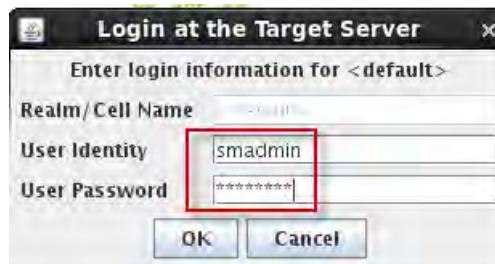
```
com.ibm.ssl.trustStore=${user.root}/config/cells/JazzSMNode01Cell/nodes/JazzSMNode01/trust.p12
```

```
# TrustStore information  
com.ibm.ssl.trustStoreName=AnotherTrustStore  
#com.ibm.ssl.trustStore=${user.root}/etc/trust.p12  
com.ibm.ssl.trustStore=${user.root}/config/cells/JazzSMNode01Cell/nodes/JazzSMNode01/trust.p12|  
com.ibm.ssl.trustStorePassword={xor}CD09Hgw=  
com.ibm.ssl.trustStoreType=PKCS12  
com.ibm.ssl.trustStoreProvider=IBMJCE  
com.ibm.ssl.trustStoreFileBased=true  
com.ibm.ssl.trustStoreReadOnly=false
```

- g. Save the modified file and close gedit.
2. Retrieve Signer Certificate from host2.
- ```
cd /opt/IBM/JazzSM/profile/bin
./retrieveSigners.sh NodeDefaultTrustStore AnotherTrustStore
-host host2.tivoli.edu -port 16313
```
- a. Verify that the information is correct and click **Y**.



- b. Enter the Dashboard Application Services Hub administrator credentials and click **OK**.



```
CWPKI0308I: Adding signer alias "root_1" to local keystore
"AnotherTrustStore"
with the following SHA digest:
C4:FE:D7:A4:19:D7:C6:54:83:E5:5B:80:5E:25:ED:34:4A:3C:41:AB
```

3. Stop Dashboard Application Services Hub.

```
cd /opt/IBM/JazzSM/profile/bin
./stopServer.sh server1 -user smadmin -password object00
```

Wait for the server to stop.

4. Start Dashboard Application Services Hub.

```
cd /opt/IBM/JazzSM/profile/bin
./startServer.sh server1
```

Wait for the server to start.

## Configuring trust on host2

You repeat the previous steps on host2 to complete the server to server trust configuration.



**Note:** You can physically switch to host2, or open a Terminal window on host1 and ssh to host2. Either way is acceptable.

1. Switch to the **host2** image.

2. Copy the revised ssl.client.props file from host1 with FTP as follows:

- a. Change to the target directory.

```
cd /opt/IBM/JazzSM/profile/properties
```

- b. Save a copy of the original file.

```
cp ssl.client.props ssl.client.props.orig
```

- c. Connect to host1 with SFTP.

```
sftp host1
```

The authenticity of host 'host1 (192.168.100.160)' can't be established.

RSA key fingerprint is 1c:2c:83:be:ca:fd:a4:86:14:29:16:2f:76:65:af:55.

Are you sure you want to continue connecting (yes/no)?

- d. Enter **yes**.

Warning: Permanently added 'host1,192.168.100.160' (RSA) to the list of known hosts.

```
netcool@host1's password:
```

- e. Enter **object00** for the password.

```
sftp>
```

- f. Change to the target directory on host1.

```
sftp> cd /opt/IBM/JazzSM/profile/properties
```

- g. Retrieve the file.

```
sftp> get ssl.client.props
```

Fetching /opt/IBM/JazzSM/profile/properties/ssl.client.props to  
ssl.client.props

```
/opt/IBM/JazzSM/profile/properties/ssl.client 100% 5109 5.0KB/s 00:00
```

- h. Exit SFTP.

```
sftp> quit
```

3. Retrieve Signer Certificate from host1.

```
cd /opt/IBM/JazzSM/profile/bin
.retrieveSigners.sh NodeDefaultTrustStore AnotherTrustStore
-host host1.tivoli.edu -port 16313
```

- a. Verify that the information is correct and click Y.



- b. Enter the Dashboard Application Services Hub administrator credentials and click OK.



```
CWPKI0308I: Adding signer alias "root_1" to local keystore
"AnotherTrustStore"
with the following SHA digest:
C4:FE:D7:A4:19:D7:C6:54:83:E5:5B:80:5E:25:ED:34:4A:3C:41:AB
```

4. Stop Dashboard Application Services Hub.

```
cd /opt/IBM/JazzSM/profile/bin
.stopServer.sh server1 -user smadmin -password object00
```

Wait for the server to stop.

5. Start Dashboard Application Services Hub.

```
cd /opt/IBM/JazzSM/profile/bin
.startServer.sh server1
```

Wait for the server to start.

# Exercise 6 Verifying cluster configuration

The cluster is configured to synchronize Dashboard Application Services Hub changes to all nodes in the cluster. In the following exercise, you create a Dashboard Application Services Hub page on host1 and verify that the page exists in Dashboard Application Services Hub on host2.

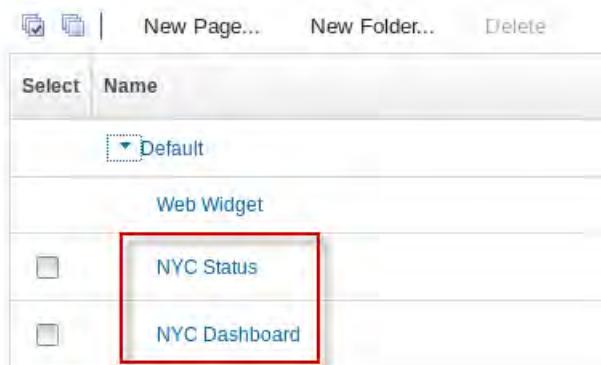


**Note:** The node that is selected for the change does not matter. You can create the page on host2. The cluster synchronizes the change to host1.

1. Open a Firefox browser on host1.
2. Change the URL to the following location:  
`http://host1.tivoli.edu:16310/ibm/console`
3. Log in as **smadmin** with password **object00**.
4. Click the gear icon and select **Pages**.



5. Expand the **Default** folder.



The folder contains NYC Status and NYC Dashboard. These pages were created on host2 in a previous exercise.

6. Click **New Page**.



7. Enter **Check cluster** for the name, scroll to the bottom of the page, and click **OK**.

### Page Settings

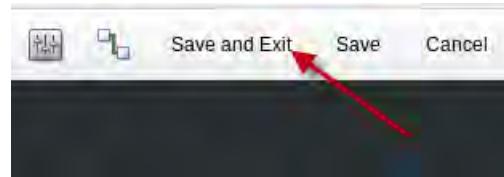
Provide a name for your new workpage and pick the default layout of The navigation location is the area where you want the new workpage

\* Required field

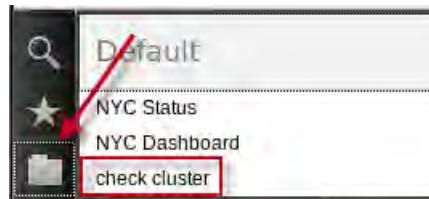
\* Page name:   

\* Page location:

8. Click **Save and Exit**.



9. Click the *folder* icon and verify that the page is listed.



10. Log out of Dashboard Application Services Hub on host1.

11. Close the Firefox browser.

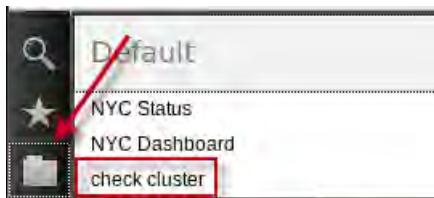
12. Open a Firefox browser.



**Important:** Make sure that the browser connects to host2.tivoli.edu.

13. Log in as **smadmin** with password **object00**.

14. Click the *folder* icon and verify that the page is listed.



When the page is added on host1, the modification is written to the cluster database. A notification is sent to host2, and the change is retrieved from the database. The change is then applied to Dashboard Application Services Hub on host2 and both nodes have the same page.

15. Log out of Dashboard Application Services Hub.

## Exercise 7 Configuring Web GUI load balancing

The server configuration is configured to support Dashboard Application Services Hub high availability. This configuration ensures that the configurations of the two Dashboard Application Services Hub instances are synchronized by storing this information in a common DB2 database. The last step is to configure the two instances of Web GUI to synchronize their configurations.

### Configuring host2

1. Switch to the **host2** image.
2. Modify Web GUI server.init file as follows:
  - a. Change to the target directory.  
`cd /opt/IBM/netcool/omnibus_webgui/etc`
  - b. Save a copy of the file before modification.  
`cp server.init server.init.orig`
  - c. Modify the file.  
`gedit server.init`
  - d. Locate the following parameter and modify as follows:  
`timedtasks.enabled:true`

- e. Locate the following parameters and modify as follows:

```
cluster.mode:on
cluster.hostname:host2.tivoli.edu
cluster.port:16311
```

```
#####
cluster.mode:on
cluster.hostname:host2.tivoli.edu
cluster.port:16311
cluster.replication.file.mode:0
cluster.waapi.notification.delay:2000
```

- f. Save the changes and exit gedit.

3. Restart Dashboard Application Services Hub.

- a. Stop the server.

```
cd /opt/IBM/JazzSM/profile/bin
.stopServer.sh server1 -user smadmin -password object00
```

Wait for the server to stop.

- b. Remove all Web GUI log files.

```
cd /opt/IBM/JazzSM/profile/logs/ncw
.bin/rm ncw.0.*
```



**Note:** When Web GUI starts, new files are created. Removing the files now ensures that the contents are smaller. The smaller log files makes it easier to review the files for any errors or issues.

- c. Start the server.

```
cd /opt/IBM/JazzSM/profile/bin
.startServer.sh server1
```

Wait for the server to start.

## Configuring host1

1. Switch to the **host1** image.
2. Modify Web GUI server.init file as follows:
  - a. Change to the target directory.  

```
cd /opt/IBM/netcool/omnibus_webgui/etc
```
  - b. Save a copy of the file before modification.  

```
cp server.init server.init.orig
```
  - c. Modify the file.  

```
gedit server.init
```

- d. Locate the following parameter and modify as follows:

timedtasks.enabled:true

- e. Locate the following parameters and modify as follows:

cluster.mode:on

cluster.hostname:host1.tivoli.edu

cluster.port:16311

```
#####
#cluster.mode:on
#cluster.hostname:host1.tivoli.edu
#cluster.port:16311
#cluster.replication.file.mode:0
#cluster.waapi.notification.delay:2000
```



**Important:** The server.init file on host2 also contains two other changes. The changes are not related to load balancing. You made changes in a previous unit to the server.init file on host2. You must also apply the same changes to the file on host1.

- f. Locate and modify the following lines as shown here:

```
columngrouping.allowedcolumns=Acknowledged,AlertGroup,Class,Customer,Location,Node,AgencyId,SiteId,NodeAlias,NmosCauseType,NmosManagedStatus,Severity,Service
```

eventviewer.tools.command:true

- g. Save the changes and exit gedit.

3. Restart Dashboard Application Services Hub.

- a. Stop the server.

```
cd /opt/IBM/JazzSM/profile/bin
```

```
./stopServer.sh server1 -user smadmin -password object00
```

Wait for the server to stop.

- b. Remove all Web GUI log files.

```
cd /opt/IBM/JazzSM/profile/logs/ncw
```

```
/bin/rm ncw.0.*
```



**Note:** When Web GUI starts, new files are created. Removing the files now ensures that the contents are smaller. The smaller log files makes it easier to review the files for any errors or issues.

- c. Start the server.

```
cd /opt/IBM/JazzSM/profile/bin
```

```
./startServer.sh server1
```

Wait for the server to start.

## Adding a data source

The copy of Web GUI on host1 does not have a definition for an ObjectServer. This typically happens when the post-installation utility is run. The utility was not run, so there is no definition. You must create a definition manually.

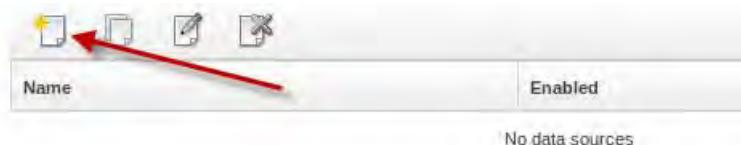


**Note:** You can perform the following steps from either host1 or host2.

1. Start a Firefox browser.
2. Enter the following URL:  
`http://host1.tivoli.edu:16310/ibm/console`
3. Log in as **ncoadmin** with password **object00**.
4. Click the icon, and select **Data Sources**.



5. Click the icon to create a data source.



6. Enter **OMNIBUS** for the name. Enter **host1.tivoli.edu** for the host. Enter **object00** for the password.

Create New Data Source

General   Failover   Display Servers   Self Monitoring   Caching   Connection Pools

Name: **OMNIBUS**  Enabled  In Default Group

Primary ObjectServer

Host: **host1.tivoli.edu**  
Port: **4100**  Use SSL ?

Authentication

User ID: **root**  
Password: **\*\*\*\*\***  Encrypted

7. Click **Failover**. Enter **host2.tivoli.edu** for the host. Enter **4100** for the port. Scroll to the bottom of the page, and click **Save Datasource**.

General   **Failover**   Display Servers   Self Monitoring   Caching   Connection Pools

Backup ObjectServer

Host: **host2.tivoli.edu**  
Port: **4100**  Use SSL ?

8. Close the data source page.  
9. Log out of Dashboard Application Services Hub.  
10. Close the Firefox browser.

## Validating Web GUI load balancing

With Web GUI load balancing, changes that are made to the Web GUI instance that is running on host1 are automatically propagated to host2. You can create a custom event filter, and verify whether it propagates correctly.

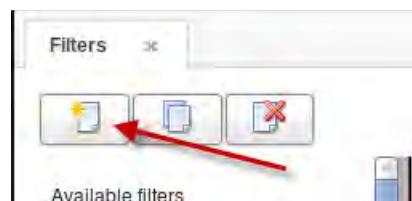


**Note:** You can perform the following steps from either host1 or host2.

1. Start a Firefox browser.
2. Enter the following URL:  
`http://host1.tivoli.edu:16310/ibm/console`
3. Log in as **ncoadmin** with password **object00**.
4. Click the person icon and select **Filters**.



5. Click the icon to create a new filter.



6. Select **global**, scroll to the bottom of the page, and click **OK**.
7. Enter **Cluster\_check** for the filter name and click **Save and Close**.

Edit Filter: New Filter

| Filter Attributes |               |
|-------------------|---------------|
| Name:             | Cluster_check |
| Default view:     | Default       |



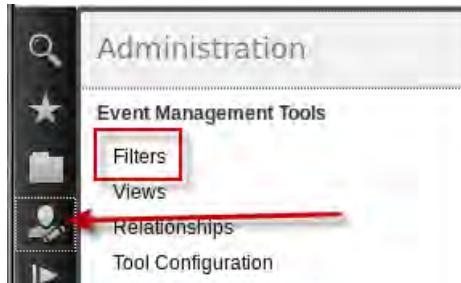
**Note:** It is necessary only to provide a name. The sample is not used as a real filter. It is intended to test load balancing.

8. Log out of Dashboard Application Services Hub.
9. Close the Firefox browser
10. Open a Firefox browser.

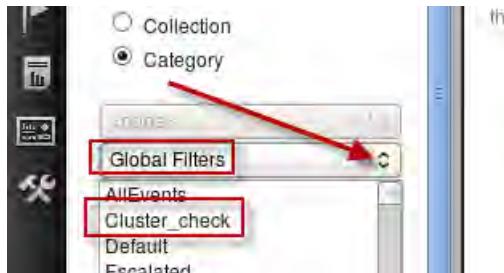


**Important:** Make sure that the browser connects to host2.tivoli.edu.

11. Log in as **ncoadmin** with password **object00**.
12. Click the person icon and select **Filters**.



13. Select **Global\_Filters** and verify that **Cluster\_check** is listed.



14. Log out of Dashboard Application Services Hub.

## Troubleshooting Web GUI load balancing

The Web GUI trace log file contains messages that are related to load balancing.



**Note:** You perform the following steps on host1.

1. Examine the Web GUI trace log.

```
cd /opt/IBM/JazzSM/profile/logs/ncw
more ncw.0.trace
```

2. Scroll down in the file and locate the following line:

```
[2014-11-26T21:19:14.977] [server.startup : 0]
com.micromuse.ncw.configstore.HaStoreAccessGatewayJpaImpl.finishInit FINE:
Joined HA cluster, initiating download from HA database.
```

This message is generated when Web GUI starts. The message indicates that the Web GUI is retrieving its configuration from the database.

If there are issues with access to the DB2 database, those issues would appear here.

3. Scroll down in the file and locate the following lines:

```
[2014-11-26T21:22:25.509] [WebContainer : 2]
com.micromuse.wave.fvbuilder.dao.XMLFilterBox.auditModification AUDIT:
HEMFV0192A=Filter 'Cluster_check' of category global has been modified by user
'ncoadmin'.
[2014-11-26T21:22:25.623] [pool-4-thread-1]
com.ibm.tivoli.ncw.ha.notification.NotificationTask.sendGetRequest FINE:
HEMHA0002I=Successfully notified the node
https://host2.tivoli.edu:16311/ibm/console/webtop/public/HANotification of a
change in the cluster configuration.
```

The first message is generated when the Cluster\_check filter is saved.

The second message is generated when the notification of the change is sent to host2.



**Note:** You perform the following steps on host2.

1. Examine the Web GUI trace log.

```
cd /opt/IBM/JazzSM/profile/logs/ncw
more ncw.0.trace
```

2. Scroll down in the file and locate the following line:

```
[2014-11-26T21:47:07.932] [server.startup : 1]
com.micromuse.ncw.configstore.HaStoreAccessGatewayJpaImpl.finishInit FINE:
Joined HA cluster, initiating download from HA database.
```

Web GUI on host2 is downloading its configuration from the database.

3. Scroll down in the file and locate the following line:

```
[2014-11-26T21:51:00.269] [pool-5-thread-1]
com.ibm.tivoli.ncw.ha.config.ConfigurationItemsServiceJpaImpl.syncExistingItemU
pdatedOnAnotherNode AUDIT: HEMHA0030A=Synchronised/data/global/filter.xml
with changes made on another node.
```

This message is generated when Web GUI on host2 retrieves the new filter that is created on host1.

In a production environment, you typically add a load balancer to the configuration. The current configuration supports only synchronization of changes. It does not contain load balancing.



## 9 Security exercises

In these exercises, you change the ObjectServer encryption method to use AES instead of DES. You configure the ObjectServer to run in FIPS 140-2 mode. You implement Secure Socket Layer communications.

### Overview

The classroom architecture consists of the following components:

- High availability pair of ObjectServers
  - NYC\_AGG\_P running on the host1 image
  - NYC\_AGG\_B running on the host2 image
  - NYC\_AGG ObjectServer gateway is running on the host2 image
- Process Activity agents
  - HOST1\_PA running on the host1 image
  - HOST2\_PA running on the host2 image
- Probes
  - Syslog probe is running on the host1 image
  - SNMP probe is running on the host1 image
  - Simnet probe is running on the host1 image
- Web GUI configured for high availability, with one instance on host1, and another instance on host2

In the following exercises, you configure the ObjectServers to support client connections over SSL. You configure the gateway, process agents, and probes to connect to the ObjectServers over SSL. You change the configuration of the ObjectServers to store user passwords in AES FIPS 140-2 encrypted format. You also change the ObjectServers to run in *secure mode*, which is not a requirement for SSL or encryption. You run the ObjectServers in *secure mode* to demonstrate the requirement for a user name, and password by probes, and gateways.



**Important:** Some of the following steps are completed on the host1 image, and some steps are completed on the host2 image. Make certain that you perform the steps on the correct image.

## Exercise 1 Configuring FIPS 140-2 encryption

### Configuring the host1 image



**Important:** The components in the class image are configured for ObjectServer high availability. As you complete the configuration steps on host1, the probes and gateways automatically switch to the backup ObjectServer.

#### Modifying the user environment

1. Switch to the **host1** image.
2. Modify the **netcool** user environment variables.
  - a. Change to the home directory.  
`cd /home/netcool`
  - b. Save a copy of the file before modifications.  
`cp .bashrc .bashrc.orig`
  - c. Open the file for edit with the gedit utility.  
`gedit .bashrc`
  - d. Add the following lines to the end of the file:  
`export KEYDB=$NCHOME/etc/security/keys/omni.kdb  
export PATH=$KEYDB:$PATH`
  - e. Save the changes, and exit the gedit utility.
  - f. Source the updated file.  
`source .bashrc`

- g. Verify the defined variable KEYDB.

```
env | grep KEYDB
```

```
KEYDB=/opt/IBM/tivoli/netcool/etc/security/keys/omni.kdb
```

## **Changing the ObjectServer to use AES encryption**

1. Change the **NYC\_AGG\_P** ObjectServer to use AES encryption.
  - a. Start the Netcool/OMNIbus Administrator utility.  
`nco_config &`
  - b. Connect to the **NYC\_AGG\_P** ObjectServer as **root** with password **object00**.
  - c. Select **System > Properties**. Scroll down and locate **PasswordEncryption**.



- d. Edit the property, and change the value to **AES**.

| Name                      | Value            |
|---------------------------|------------------|
| PA.Password               | EDEAAPAIANFMCHCB |
| PA.Username               | netcool          |
| <b>PasswordEncryption</b> | <b>AES</b>       |
| PasswordFormat            | 8:1:1:0          |
| Profile                   | true             |

- e. Exit the Netcool/OMNIbus Administrator utility.

A change to the PasswordEncryption property requires an ObjectServer restart.

2. Stop the **NYC\_AGG\_P** ObjectServer.

```
nco_pa_stop -server HOST1_PA -process MasterObjectServer -password object00
```

3. Start the **NYC\_AGG\_P** ObjectServer.

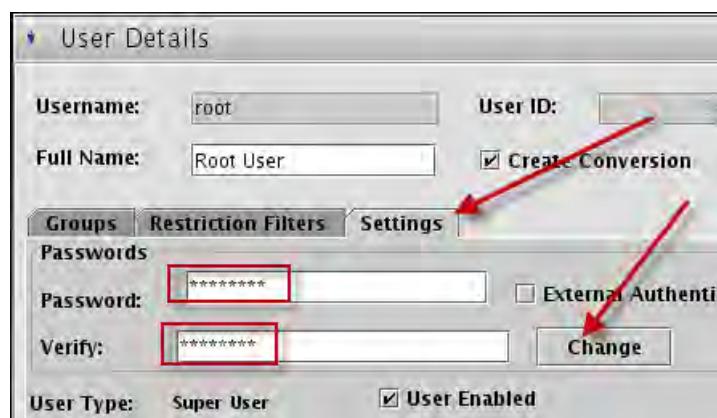
```
nco_pa_start -server HOST1_PA -process MasterObjectServer -password object00
```

4. Verify the status of the **NYC\_AGG\_P** ObjectServer.

```
nco_pa_status -server HOST1_PA -password object00
```

The PasswordEncryption property controls how the password is encrypted for users that are authenticated against the ObjectServer. The **root** user is authenticated against the ObjectServer. The password for the root user is currently stored in the ObjectServer in DES format. You must modify the root user password to force the ObjectServer to encrypt the password with AES format.

5. Change the password for the **root** user in the **NYC\_AGG\_P** ObjectServer.
  - a. Start the Netcool/OMNIbus Administrator utility.  
nco\_config &
  - b. Connect to the **NYC\_AGG\_P** ObjectServer as **root** with password **object00**.
  - c. Select **User > Users**.
  - d. Right-click **root**, and select **Edit User**.
  - e. Click the **Settings** tab, and click **Change**. Enter **object00** for the password, and click **OK** to save the user.



- f. Click **Yes** to confirm the change.



**Note:** This process changes the encryption of the old saved user password to the new AES encryption. This change can be seen in the **security.users** database table data tab as shown in the following screen capture.

| erName | SystemUser | FullName      | Password                 |
|--------|------------|---------------|--------------------------|
| 1      |            | Root User     | U9Y9yBIQk4EvD61FaDapMw== |
| 3      |            | Netcool Admin |                          |
| 3      |            | Netcool User  |                          |
| 3      |            | William Hill  |                          |
| 4      |            | Nobody        |                          |

6. Close the Netcool/OMNIbus Administrator utility.

## Configuring the JRE for FIPS 140–2 mode

To configure the Netcool/OMNIbus JRE for FIPS 140–2 operation, you change the configuration of the security properties file.

Modify the security properties file as follows.

1. Edit the file.

```
cd $NCHOME/platform/linux2x86/jre64_1.7.0/jre/lib/security/
cp java.security java.security.orig
chmod +w java.security
gedit java.security
```

2. Modify the file as follows:

- a. In the list of providers and their preference orders section, add the following lines:

```
security.provider.1=com.ibm.fips.jsse.IBMJSSEFIPSProvider
security.provider.2=com.ibm.crypto.fips.provider.IBMJCEFIPS
```

- b. For all other providers, increment the number by two, as shown in bold in the following list:

```
security.provider.1=com.ibm.fips.jsse.IBMJSSEFIPSProvider
security.provider.2=com.ibm.crypto.fips.provider.IBMJCEFIPS
security.provider.3=com.ibm.jsse2.IBMJSSEProvider2
security.provider.4=com.ibm.crypto.provider.IBMJCE
security.provider.5=com.ibm.security.jgss.IBMJGSSProvider
security.provider.6=com.ibm.security.cert.IBMCertPath
security.provider.7=com.ibm.security.sasl.IBMSASL
security.provider.8=com.ibm.xml.crypto.IBMXMLCryptoProvider
security.provider.9=com.ibm.xml.enc.IBMXMLEncProvider
security.provider.10=com.ibm.security.jgss.mech.spnego.IBMSPNEGO
security.provider.11=sun.security.provider.Sun
security.provider.12=com.ibm.security.cmskeystore.CMSProvider
```

- c. Set the default key and trust manager factory algorithms for the javax.net.ssl package:

```
ssl.KeyManagerFactory.algorithm=IbmX509
ssl.TrustManagerFactory.algorithm=IbmX509
```

```

Determines the default key and trust manager factory algorithms for
the javax.net.ssl package.

ssl.KeyManagerFactory.algorithm=IbmX509
ssl.TrustManagerFactory.algorithm=IbmX509
```

- d. Set the default SSLSocketFactory, and SSLSocketFactory provider implementations for the javax.net.ssl package by adding the following two lines:

```
ssl.SocketFactory.provider=com.ibm.jsse2.SSLSocketFactoryImpl
ssl.ServerSocketFactory.provider=com.ibm.jsse2.SSLSocketFactoryImpl

#
Determines the default key and trust manager factory algorithms for
the javax.net.ssl package.
#
ssl.KeyManagerFactory.algorithm=IbmX509
ssl.TrustManagerFactory.algorithm=IbmX509
ssl.SocketFactory.provider=com.ibm.jsse2.SSLSocketFactoryImpl
ssl.ServerSocketFactory.provider=com.ibm.jsse2.SSLSocketFactoryImpl
```

3. Save the changes, and exit the gedit utility.

## **Creating FIPS configuration file**

A FIPS configuration file is required for FIPS initialization. This file is called **fips.conf**, and is required on each computer where an ObjectServer component is installed.

1. Create the FIPS configuration file as follows:

```
cd $NCHOME/etc/security
touch fips.conf
chmod 777 fips.conf
```

2. Stop the **NYC\_AGG\_P** ObjectServer.

```
nco_pa_stop -server HOST1_PA -process MasterObjectServer -password object00
```

3. Start the **NYC\_AGG\_P** ObjectServer.

```
nco_pa_start -server HOST1_PA -process MasterObjectServer -password object00
```

4. Verify the status of the **NYC\_AGG\_P** ObjectServer.

```
nco_pa_status -server HOST1_PA -password object00
```

The **NYC\_AGG\_P** ObjectServer is now configured to support AES encryption. The **NYC\_AGG\_P** ObjectServer is running in FIPS 140-2 mode.

You must repeat these steps on the host2 image to configure the **NYC\_AGG\_B** ObjectServer.

## **Configuring the host2 image**

The modifications for host2 are the same as for host1. The only difference is the name of the ObjectServer.

## Modifying the user environment

1. Switch to the **host2** image.
2. Modify the **netcool** user environment variables.

- a. Change to the home directory.

```
cd /home/netcool
```

- b. Save a copy of the file before modifications.

```
cp .bashrc .bashrc.orig
```

- c. Open the file for edit with the gedit utility.

```
gedit .bashrc
```

- d. Add the following lines to the end of the file:

```
export KEYDB=$NCHOME/etc/security/keys/omni.kdb
export PATH=$KEYDB:$PATH
```

- e. Save the changes, and exit the gedit utility.

- f. Source the updated file.

```
source .bashrc
```

- g. Verify the defined variable KEYDB.

```
env | grep KEYDB
```

```
KEYDB=/opt/IBM/tivoli/netcool/etc/security/keys/omni.kdb
```

## Changing the ObjectServer to use AES encryption

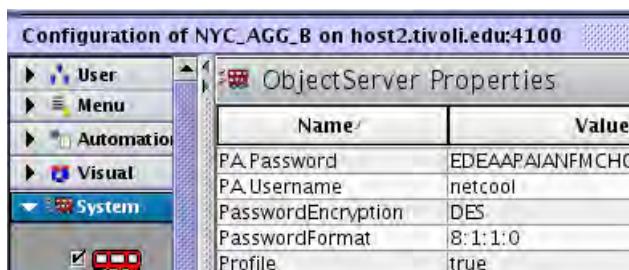
1. Change the **NYC\_AGG\_B** ObjectServer to use AES encryption.

- a. Start the Netcool/OMNIbus Administrator utility.

```
nco_config &
```

- b. Connect to the **NYC\_AGG\_B** ObjectServer as **root** with password **object00**.

- c. Select **System > Properties**. Scroll down and locate **PasswordEncryption**.



- d. Edit the property, and change the value to **AES**.

| Parameter                 | Value             |
|---------------------------|-------------------|
| PA_Password               | EDEAAAPAIANFMCHCB |
| PA_Username               | netcool           |
| <b>PasswordEncryption</b> | <b>AES</b>        |
| PasswordFormat            | 8:1:1:0           |
| Profile                   | true              |

- e. Exit the Netcool/OMNIbus Administrator utility.

2. Stop the **NYC\_AGG\_B** ObjectServer.

```
nco_pa_stop -server HOST2_PA -process BackupObjectServer -password object00
```

3. Start the **NYC\_AGG\_B** ObjectServer.

```
nco_pa_start -server HOST2_PA -process BackupObjectServer -password object00
```

4. Verify the status of the **NYC\_AGG\_B** ObjectServer.

```
nco_pa_status -server HOST2_PA -password object00
```

5. Change the password for the **root** user in the **NYC\_AGG\_B** ObjectServer.

- a. Start the Netcool/OMNIbus Administrator utility.

```
nco_config &
```

- b. Connect to the **NYC\_AGG\_B** ObjectServer as **root** with password **object00**.

- c. Select **User > Users**.

- d. Right-click **root**, and select **Edit User**.

- e. Click the **Settings** tab, and click **Change**. Enter **object00** for the password, and click **OK** to save the user.

- f. Click **Yes** to confirm the change.

6. Close the Netcool/OMNIbus Administrator utility.

## Configuring the JRE for FIPS 140–2 mode

Modify the security properties file as follows.

1. Edit the file.

```
cd $NCHOME/platform/linux2x86/jre64_1.7.0/jre/lib/security/
cp java.security java.security.orig
chmod +w java.security
gedit java.security
```

2. Modify the file as follows:



**Note:** The file changes are the same as host1.

- a. In the list of providers and their preference orders section, add the following lines:

```
security.provider.1=com.ibm.fips.jsse.IBMJSSEFIPSPProvider
security.provider.2=com.ibm.crypto.fips.provider.IBMJCEFIPS
```

- b. For all other providers, increment the number by two, as shown in bold in the following list:

```
security.provider.1=com.ibm.fips.jsse.IBMJSSEFIPSPProvider
security.provider.2=com.ibm.crypto.fips.provider.IBMJCEFIPS
security.provider.3=com.ibm.jsse2.IBMJSSEProvider2
security.provider.4=com.ibm.crypto.provider.IBMJCE
security.provider.5=com.ibm.security.jgss.IBMJGSSProvider
security.provider.6=com.ibm.security.cert.IBMCertPath
security.provider.7=com.ibm.security.sasl.IBMSASL
security.provider.8=com.ibm.xml.crypto.IBMXMLCryptoProvider
security.provider.9=com.ibm.xml.enc.IBMXMLEncProvider
security.provider.10=com.ibm.security.jgss.mech.spnego.IBMSPNEGO
security.provider.11=sun.security.provider.Sun
security.provider.12=com.ibm.security.cmskeystore.CMSProvider
```

- c. Set the default key and trust manager factory algorithms for the javax.net.ssl package:

```
ssl.KeyManagerFactory.algorithm=IbmX509
ssl.TrustManagerFactory.algorithm=IbmX509
```

```

Determines the default key and trust manager factory algorithms for
the javax.net.ssl package.

ssl.KeyManagerFactory.algorithm=IbmX509
ssl.TrustManagerFactory.algorithm=IbmX509
```

- d. Set the default SSLSocketFactory, and SSLSocketFactory provider implementations for the javax.net.ssl package by adding the following two lines:

```
ssl.SocketFactory.provider=com.ibm.jsse2.SSLSocketFactoryImpl
ssl.ServerSocketFactory.provider=com.ibm.jsse2.SSLSocketFactoryImpl
```

```

Determines the default key and trust manager factory algorithms for
the javax.net.ssl package.

ssl.KeyManagerFactory.algorithm=IbmX509
ssl.TrustManagerFactory.algorithm=IbmX509
ssl.SocketFactory.provider=com.ibm.jsse2.SSLSocketFactoryImpl
ssl.ServerSocketFactory.provider=com.ibm.jsse2.SSLSocketFactoryImpl
```

3. Save the changes, and exit the gedit utility.

## **Creating FIPS configuration file**

1. Create the FIPS configuration file as follows:

```
cd $NCHOME/etc/security
touch fips.conf
chmod 777 fips.conf
```

2. Stop the **NYC\_AGG\_B** ObjectServer.

```
nco_pa_stop -server HOST2_PA -process BackupObjectServer -password object00
```

3. Start the **NYC\_AGG\_B** ObjectServer.

```
nco_pa_start -server HOST2_PA -process BackupObjectServer -password object00
```

4. Verify the status of the **NYC\_AGG\_B** ObjectServer.

```
nco_pa_status -server HOST2_PA -password object00
```

The **NYC\_AGG\_B** ObjectServer is now configured to support AES encryption. The **NYC\_AGG\_B** ObjectServer is running in FIPS 140-2 mode.

# **Exercise 2. Configuring the ObjectServers for SSL access**

In this exercise, you configure the ObjectServers to support access over SSL. You start by adding SSL access to the ObjectServers while retaining the non-SSL access. After the SSL configuration is completed, and verified, you remove the non-SSL access.

Again, changes are required on both host images.

## **Configuring the host1 image**

### ***Adding SSL access to the ObjectServers***

1. Switch to the **host1** image.
2. Start the Server Editor utility.

```
nco_xigen &
```

3. Add SSL to all ObjectServers as follows:
  - a. Click **NYC\_AGG\_P** to select the entry.
  - b. Enter **4110** for the SSL port number, and click **Update**.

The screenshot shows the 'Server' list and the 'Edit Server' dialog box. In the 'Server' list, several entries are listed under 'AGG\_V' (Backup1, HOST1\_PA, HOST2\_PA, NYC\_AGG\_B, NYC\_AGG\_GATE) and one entry 'NYC\_AGG\_P' is selected and highlighted with a red box. In the 'Edit Server' dialog box, the 'Name' field contains 'NYC\_AGG\_P', the 'Host' field contains 'host1.tivoli.edu', the 'Port' field contains '4100', and the 'SSL' field contains '4110'. A red arrow points from the 'SSL' field in the dialog box to the 'SSL' field in the 'Server' list.

| Server           | Hostname         | Port |
|------------------|------------------|------|
| AGG_V            | host1.tivoli.edu | 4100 |
| └ Backup1:       | host2.tivoli.edu | 4100 |
| HOST1_PA         | host1.tivoli.edu | 4200 |
| HOST2_PA         | host2.tivoli.edu | 4200 |
| NYC_AGG_B        | host2.tivoli.edu | 4100 |
| NYC_AGG_GATE     | host2.tivoli.edu | 4300 |
| <b>NYC_AGG_P</b> | host1.tivoli.edu | 4100 |

**Server**

Name: **NYC\_AGG\_P**

Host: **host1.tivoli.edu**

Port: **4100**

SSL: **4110**

**Add** **Update**

- c. Click **NYC\_AGG\_B** to select the entry.
- d. Enter **4110** for the SSL port number, and click **Update**.



**Hint:** You can use the same port number because the NYC\_AGG\_B ObjectServer runs on host2.

You must also add SSL access to the *virtual* ObjectServer.

- e. Click **AGG\_V** to select the entry.
- f. Enter **4110** for the SSL port number, and click **Update**.
- g. Click the *second entry* for **AGG\_V** to select it.

- h. Enter **4110** for the SSL port number, and click **Update**.

The screenshot shows a table of ObjectServers with columns: Server, Hostname, Port, and SSL. A row for 'Backup1: host2.tivoli.edu' is selected and highlighted with a red box. In the bottom panel, the 'SSL' field for this server is set to '4110' and is also highlighted with a red box. A red arrow points from the 'SSL' field in the bottom panel to the 'Update' button, indicating the action to be taken.

| Server                    | Hostname         | Port | SSL  |
|---------------------------|------------------|------|------|
| AGG_V                     | host1.tivoli.edu | 4100 | 4110 |
| Backup1: host2.tivoli.edu |                  | 4100 |      |
| HOST1_PA                  | host1.tivoli.edu | 4200 |      |
| HOST2_PA                  | host2.tivoli.edu | 4200 |      |
| NYC_AGG_B                 | host2.tivoli.edu | 4100 | 4110 |
| NYC_AGG_GATE              | host2.tivoli.edu | 4300 |      |
| NYC_AGG_P                 | host1.tivoli.edu | 4100 | 4110 |

**Server**

Name: AGG\_V  
Host: host2.tivoli.edu  
Port: 4100  
SSL: 4110

**Priority**

Generate All

- i. Click **Apply**.

- j. Click **Close**.

The interfaces file on host1 is modified to reflect SSL access for all ObjectServers.

## Creating the key database

The ObjectServers are running in FIPS 140-2 mode. Therefore, you must use the nc\_gskcmd utility to create the key database.

- Enter the following command to create the database with FIPS compliance:

```
nc_gskcmd -keydb -create -db "$NCHOME/etc/security/keys/omni.kdb"
-pw Object00netcool -stash -expire 7300
```



**Important:** The key database password for this exercise is *Object00netcool* (*Object-zero-zero-netcool*). You need this information to complete the remaining exercises.

Use the IBM Key Management utility to verify that you successfully created the key database.

- Start the utility.

```
nc_ikeyman &
```

- Click **Key Database File > Open**.
- Select **CMS** from the **Key database type** list.

5. Browse to the location of the key database.

/opt/IBM/tivoli/netcool/etc/security/keys/omni.kdb

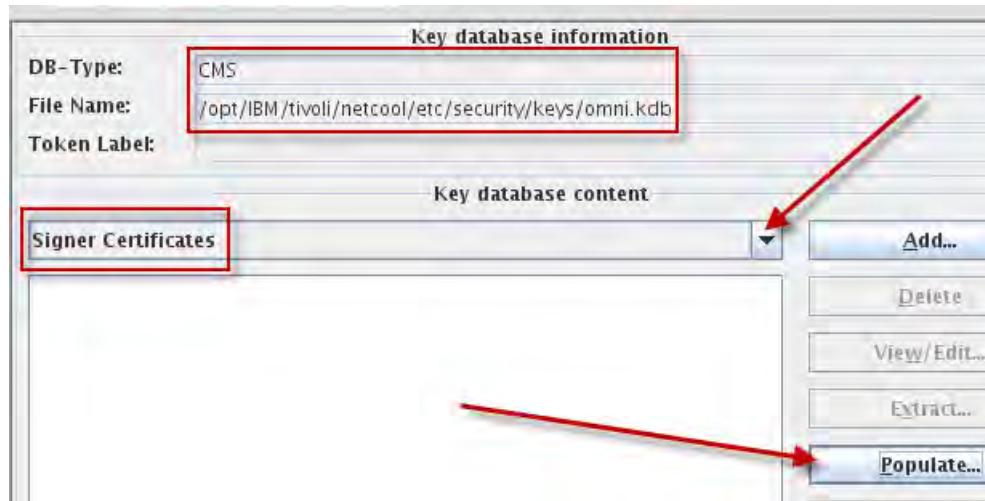
6. Click **OK**.



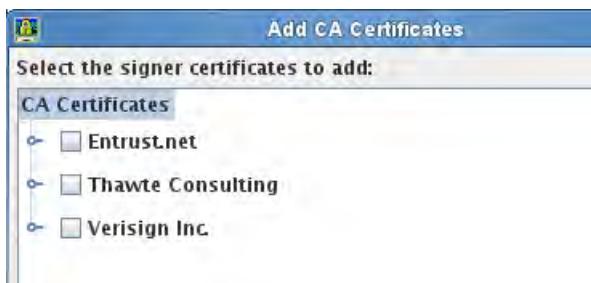
7. Enter **Object00netcool** for the password, and click **OK**.



8. Select Signer Certificates in the selection box. Click **Populate**, and observe the available commercial certificate authorities.



The list of certificate authorities opens.



9. Click **Cancel** to close list of certificate authorities.

Leave the key management utility open. You use it again shortly.

## **Creating self-signed certificates**

Public key cryptography is premised upon someone with the authority to verify the identity of a specific party to the communications process. You can rely on an external certificate authority to sign the keys. Or you can assume that the network is secure enough for an internal certificate authority to sign the keys.

For this exercise, you create an internal certificate authority, which is a key that is generated, and used to sign server certificates.

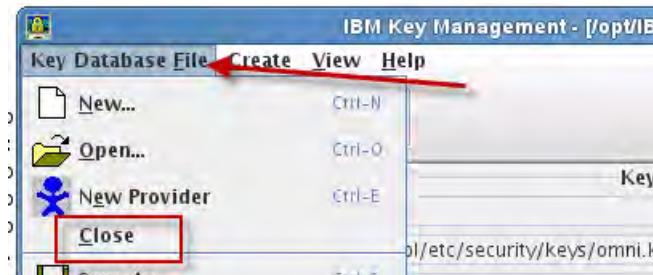
1. Create a self-signed certificate as follows:

```
nc_gskcmd -cert -create -db $KEYDB -pw Object00netcool -label host1_CA
-dn "C=US,ST=ILLINOIS,L=CHICAGO,O=IBM SWG,OU=Education,CN=host1_CA" -size 1024
-expire 3650 -ca true
```

**Note:** Notice the use of **-db \$KEYDB**. When you use the environment variable, \$KEYDB, you eliminate the need to type the entire path for the key database.

After you enter the command, you can view the certificate details within the key management utility. Close and reopen the **omni.kdb** database to see the changes.

2. Return to the key management utility.
3. Click **Key Database File > Close** to close the key database.



4. Click **Key Database File > Open**.
5. Select **CMS** from the **Key database type** list.
6. Click **Browse**. Locate and select **omni.kdb**.

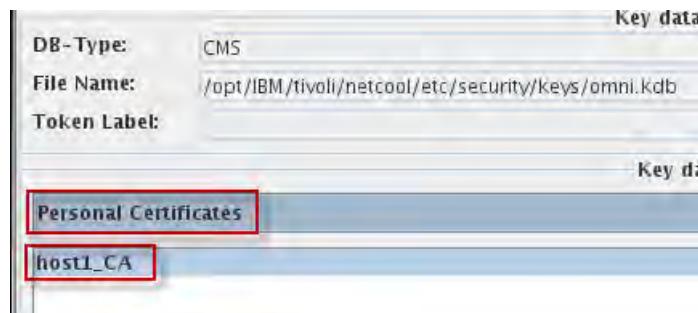
7. Click **OK**.



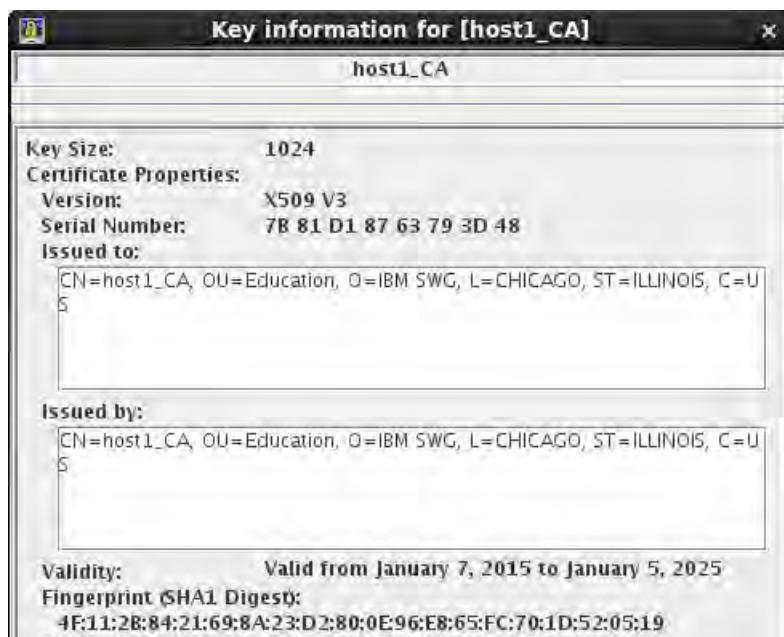
8. Enter **Object00netcool** for the password, and click **OK**.



9. Click the arrow, and select **Personal Certificates**.



10. Double-click **host1\_CA** to view the certificate details.



11. Click **OK** to close the certificate window.

You must export the certificate to a file and copy the file to the host2 image. You import the certificate on host2 in a subsequent exercise.

12. Export the certificate as follows:

```
nc_gskcmd -cert -extract -db $KEYDB -pw Object00netcool -label "host1_CA"
-target "$NCHOME/etc/security/keys/host1_CA.arm"
```

13. FTP the file to the host2 image as follows:

- a. Change to the local target directory.

```
cd /opt/IBM/tivoli/netcool/etc/security/keys
```

- b. Open a session to host2.

```
sftp host2
```

Connecting to host2...

```
netcool@host2's password: object00
```

```
sftp>
```

- c. Change to the key file directory on host2.

```
sftp> cd /opt/IBM/tivoli/netcool/etc/security/keys
```

- d. Copy the file.

```
sftp> put host1_CA.arm
```

Uploading host1\_CA.arm to

```
/opt/IBM/tivoli/netcool/etc/security/keys/host1_CA.arm
host1_CA.arm 100% 981 1.0KB/s 00:00
sftp>
```

- e. Exit.

```
sftp> quit
```

Leave the key management utility open. You use it again shortly.

## **Generating ObjectServer keys**

In the following step, you generate the ObjectServer certificate request for NYC\_AGG\_P. The output from this step is a certificate that is unsigned (not trusted). The certificate request is stored in the key management database. The request is also created as a file.

1. Create the certificate request for NYC\_AGG\_P with the following command:

```
nc_gskcmd -certreq -create -db $KEYDB -pw Object00netcool -label NYC_AGG_P
-dn "C=US,ST=ILLINOIS,L=CHICAGO,O=IBM SWG,OU=Education,CN=AGG_V" -size 1024
-file NYC_AGG_P_certreq.arm
```



**Important:** The **label** value corresponds to the primary ObjectServer: NYC\_AGG\_P. The common name (**CN**) is set to the *virtual* ObjectServer: AGG\_V. This configuration is important because clients (native clients, probes, and gateways) connect to the *virtual* ObjectServer when running in high availability. However, they physically connect to the primary or backup ObjectServer depending upon whether the primary ObjectServer is available or not.

## ***Siging the ObjectServer key***

In this step, the certificate request is signed. This process results in a certificate that is trusted. If you are using an external certificate authority, you need to send the certificate request file to the external authority. In the classroom environment, you created an internal authority (private key). You use the private key to sign the certificate request.

1. Enter the following command, noting the target and file names:

```
nc_gskcmd -cert -sign -db $KEYDB -pw Object00netcool -label host1_CA
-target NYC_AGG_P_cert.arm -expire 1000 -file NYC_AGG_P_certreq.arm
```



**Note:** The label value of **host1\_CA** identifies the certificate authority that is created in a previous step.

## ***Importing the signed key***

The last step imports the signed ObjectServer certificate into the database. Again, be aware of the file names.

1. Enter the following command to receive the signed key for **NYC\_AGG\_P**:

```
nc_gskcmd -cert -receive -file NYC_AGG_P_cert.arm -db $KEYDB -pw Object00netcool
```

2. Return to the key management Database utility.
3. Close, and reopen the omni.kdb database.
4. Select **Personal Certificates**.



Leave the key management utility open. You use it again shortly.

## Configuring the host2 image

### **Adding SSL access to the ObjectServers**

1. Switch to the **host2** image.

2. Start the Server Editor utility.

```
nco_xigen &
```

3. Add SSL to all ObjectServers as follows:

- a. Click **NYC\_AGG\_P** to select the entry.

- b. Enter **4110** for the SSL port number, and click **Update**.

- c. Click **NYC\_AGG\_B** to select the entry.

- d. Enter **4110** for the SSL port number, and click **Update**.

You must also add SSL access to the *virtual* ObjectServer.

- e. Click **AGG\_V** to select the entry.

- f. Enter **4110** for the SSL port number, and click **Update**.

- g. Click the *second* entry for **AGG\_V** to select it.

- h. Enter **4110** for the SSL port number, and click **Update**.

The updated entries appear as follows.

| Server       | Hostname         | Port | SSL  |
|--------------|------------------|------|------|
| AGG_V        | host1.tivoli.edu | 4100 | 4110 |
| Backup1      | host2.tivoli.edu | 4100 | 4110 |
| HOST1_PA     | host1.tivoli.edu | 4200 |      |
| HOST2_PA     | host2.tivoli.edu | 4200 |      |
| JDBC_GATE    | host2.tivoli.edu | 4301 |      |
| LON_AGG_P    | host2.tivoli.edu | 4105 |      |
| NYC_AGG_B    | host2.tivoli.edu | 4100 | 4110 |
| NYC_AGG_GATE | host2.tivoli.edu | 4300 |      |
| NYC_AGG_P    | host1.tivoli.edu | 4100 | 4110 |

- i. Click **Apply**.

- j. Click **Close**.

The interfaces file on host2 is modified to reflect SSL access for all ObjectServers.

## Creating the key database

1. Enter the following command to create the database with FIPS compliance:

```
nc_gskcmd -keydb -create -db "$NCHOME/etc/security/keys/omni.kdb"
-pw Object00netcool -stash -expire 7300
```



**Important:** The key database password for this exercise is *Object00netcool* (*Object-zero-zero-netcool*). You need this information to complete the remaining exercises.

Use the IBM Key Management utility to verify that you successfully created the key database.

2. Start the utility.

```
nc_ikeyman &
```

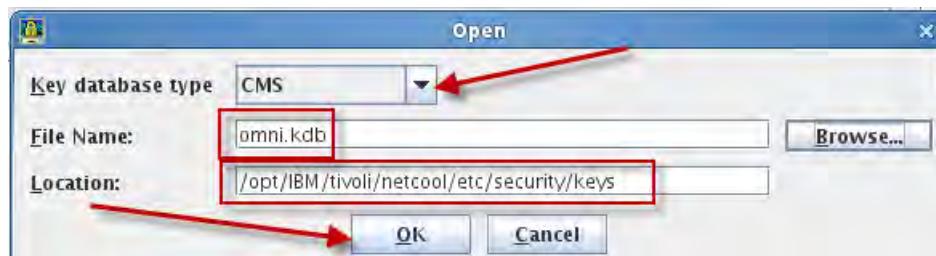
3. Click **Key Database File > Open**.

4. Select **CMS** from the **Key database type** list.

5. Browse to the location of the key database.

```
/opt/IBM/tivoli/netcool/etc/security/keys/omni.kdb
```

6. Click **OK**.



7. Enter **Object00netcool** for the password, and click **OK**.



Leave the key management utility open. You use it again shortly.

## Creating self-signed certificates

For this exercise, you create a self-signed certificate for host2, and import the self-signed certificate from host1.

1. Create a self-signed certificate for host2 as follows:

```
nc_gskcmd -cert -create -db $KEYDB -pw Object00netcool -label host2_CA
-dn "C=US,ST=ILLINOIS,L=CHICAGO,O=IBM SWG,OU=Education,CN=host2_CA" -size 1024
-expire 3650 -ca true
```

2. Enter the following commands to receive the **host1\_CA** self-signed key:

```
cd /opt/IBM/tivoli/netcool/etc/security/keys
```

```
nc_gskcmd -cert -add -file host1_CA.arm -db $KEYDB -pw Object00netcool
-label host1_CA
```

3. Return to the key management utility.

4. Click **Key Database File > Close** to close the key database.

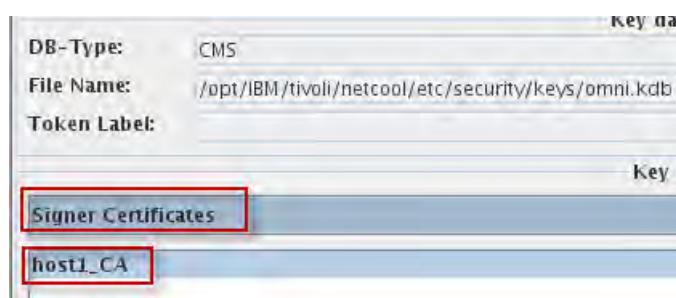
5. Click **Key Database File > Open**.

6. Select **CMS** from the **Key database type** list.

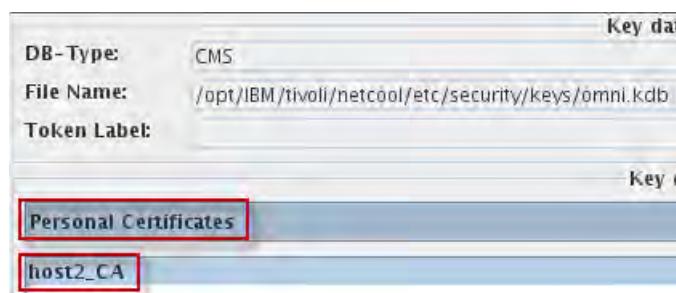
7. Click **Browse**, locate, and select **omni.kdb**.

8. Click **OK**.

9. Click the arrow, and select **Signer Certificates**.



10. Click the arrow, and select **Personal Certificates**.



You must export the host2 self-signed certificate to a file, and copy the file to the host1 image. You import the certificate on host1 in a subsequent exercise.

### 11. Export the certificate as follows:

```
nc_gskcmd -cert -extract -db $KEYDB -pw Object00netcool -label "host2_CA"
-target "$NCHOME/etc/security/keys/host2_CA.arm"
```

Leave the key management utility open. You use it again shortly.

## Generating ObjectServer keys

In the following step, you generate the ObjectServer certificate request for NYC\_AGG\_B.

### 1. Create the certificate request for **NYC\_AGG\_B** with the following command:

```
nc_gskcmd -certreq -create -db $KEYDB -pw Object00netcool -label NYC_AGG_B
-dn "C=US,ST=ILLINOIS,L=CHICAGO,O=IBM SWG,OU=Education,CN=AGG_V" -size 1024
-file NYC_AGG_B_certreq.arm
```



**Important:** The **label** value corresponds to the backup ObjectServer: NYC\_AGG\_B. The common name (**CN**) is set to the *virtual* ObjectServer: AGG\_V.

## Signing the ObjectServer key

### 1. Enter the following command, noting the target and file names:

```
nc_gskcmd -cert -sign -db $KEYDB -pw Object00netcool -label host2_CA
-target NYC_AGG_B_cert.arm -expire 1000 -file NYC_AGG_B_certreq.arm
```



**Note:** The label value of **host2\_CA** identifies the certificate authority that is created in a previous step.

## Importing the signed key

The last step imports the signed ObjectServer certificate into the database. Again, be aware of the file names.

### 1. Enter the following command to receive the signed key for **NYC\_AGG\_B**:

```
nc_gskcmd -cert -receive -file NYC_AGG_B_cert.arm -db $KEYDB -pw Object00netcool
```

### 2. Return to the key management Database utility.

### 3. Close, and reopen the omni.kdb database.

#### 4. Select Personal Certificates.



Leave the key management utility open. You use it again shortly.

## Distributing ObjectServer keys

Every computer that hosts a Netcool/OMNibus component requires the ObjectServer key. In a high availability configuration, both ObjectServer keys must be distributed. The signed key files for each ObjectServer must be copied to each system, and imported into the corresponding key database.

### **Importing keys on host1**

You can distribute both key files with one FTP session.

1. Switch to the **host1** image.
2. Open a Terminal window if necessary.
3. Change to the directory of the key files.

```
cd /opt/IBM/tivoli/netcool/etc/security/keys
```

4. FTP the file to the host2 image as follows:

- a. Open a session to host2.

```
sftp host2
Connecting to host2...
netcool@host2's password: object00
sftp>
```

- b. Change to the key file directory on host2.

```
sftp> cd /opt/IBM/tivoli/netcool/etc/security/keys
```

- c. Copy the **NYC\_AGG\_P** ObjectServer certificate file to host2.

```
sftp> put NYC_AGG_P_cert.arm
Uploading NYC_AGG_P_cert.arm to
/opt/IBM/tivoli/netcool/etc/security/keys/NYC_AGG_P_cert.arm
NYC_AGG_P_cert.arm 100% 948 0.9KB/s 00:00
sftp>
```

- d. Retrieve the **NYC\_AGG\_B** ObjectServer certificate file from host2.

```
sftp> get NYC_AGG_B_cert.arm
Fetching /opt/IBM/tivoli/netcool/etc/security/keys/NYC_AGG_B_cert.arm to
NYC_AGG_B_cert.arm
/opt/IBM/tivoli/netcool/etc/security/keys/NY 100% 859 0.8KB/s 00:00
sftp>
```

- e. Retrieve the **host2\_CA** self-signed certificate file from host2.

```
sftp> get host2_CA.arm
Fetching /opt/IBM/tivoli/netcool/etc/security/keys/host2_CA.arm to
host2_CA.arm
/opt/IBM/tivoli/netcool/etc/security/keys/host2_CA. 100% 891 0.9KB/s
00:00
sftp>
```

- f. Exit.

```
sftp> quit
```

5. Enter the following command to receive the signed key for **NYC\_AGG\_B**:

```
nc_gskcmd -cert -add -file NYC_AGG_B_cert.arm -db $KEYDB -pw Object00netcool
-label NYC_AGG_B
```

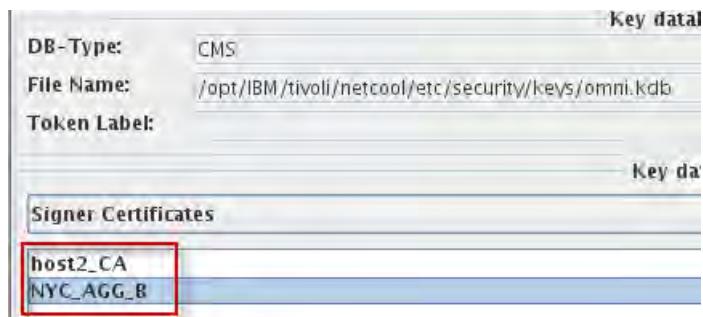
6. Enter the following command to receive the self-signer key for **host2**:

```
nc_gskcmd -cert -add -file host2_CA.arm -db $KEYDB -pw Object00netcool
-label host2_CA
```

7. Return to the key management Database utility.

8. Close, and reopen the omni.kdb database.

9. Select **Signer Certificates**.



The certificates are listed in the IBM Key Management window, as entries in the Signer Certificates list. The labels that you entered are used to identify the certificates.

10. Click **Key Database File > Exit** to close the key management utility.

You must import the NYC\_AGG\_P ObjectServer key into the key database on the host2 image.

## Importing keys on host2

1. Switch to **host2**.
2. Enter the following command to receive the signed key for **NYC\_AGG\_P**:
 

```
nc_gskcmd -cert -add -file NYC_AGG_P_cert.arm -db $KEYDB -pw Object00netcool
-label NYC_AGG_P
```
3. Return to the key management Database utility.
4. Close, and reopen the omni.kdb database.
5. Select **Signer Certificates**.



6. Click **Key Database File > Exit** to close the key management utility.

ObjectServer keys are generated, signed, and distributed. The ObjectServers are configured to support access over SSL. In the next exercise, you configure the client components to access the ObjectServers over SSL.

## Exercise 3 SSL property value encryption

A major topic about FIPS cryptography modules is the encryption of property values within any property file. You can use property value encryption to encrypt string values in a properties file or configuration file so that the strings cannot be read without a key. When the process that uses the properties file or configuration file starts up, the strings are decrypted. You can use **nco\_keygen**, and **nco\_aes\_crypt** to do property value encryption, even if you are not enabling FIPS 140-2 compliance.



**Important:** Any string value within a Netcool/OMNIbus properties file can be encrypted except for the following two main properties and the **MessageLog** field in the **\*.props** file:

```
ConfigCryptoAlg : 'AES_FIPS'
ConfigKeyFile : '$NCHOME/etc/security/keys/netcool.keygen'
```

The client decrypts the cipher-text property value into plain-text. Properties such as passwords can then be further protected by SSL (FIPS-compliant or not).

## Configuring the host1 image

1. Switch to the **host1** image.

The first step creates the key file that is used to encrypt the property values.

2. Run the following command to build the key file:

```
nco_keygen -o $NCHOME/etc/security/keys/netcool.keygen
```

Perform the following steps to encrypt the password **object00**, and to store it. You use the encrypted text in several subsequent steps.

3. Encrypt the **object00** password as follows:

```
cd $NCHOME/etc/security/keys
```

```
nco_aes_crypt -o $NCHOME/etc/security/keys/encrypted_output.txt -c AES_FIPS
-k $NCHOME/etc/security/keys/netcool.keygen object00
```

The command places the encrypted text in a file.

4. View the encrypted text.

```
more $NCHOME/etc/security/keys/encrypted_output.txt
```

```
@44:3WZnAsSKpuWSlgrxx0W+g+C0crevWmcJKPbmryCmlm4=@
```

You copy, and paste the cipher text into various locations in subsequent steps.



**Hint:** Leave the Terminal window as is, and open a new Terminal window. Use the second Terminal window for the remaining steps.

## ObjectServer properties

1. Modify the ObjectServer property file as follows.

- a. Change to the target directory.

```
cd $OMNIHOME/etc
```

- b. Save a copy of the file before modifications.

```
cp NYC_AGG_P.props NYC_AGG_P.props.orig
```

- c. Edit the file with the gedit utility.

```
gedit NYC_AGG_P.props &
```



**Important:** There are property lines at the top of the file that are commented out. Do not change these lines.

- d. Scroll down in the file and locate the first property line with no comment character.

```
UniqueLog: FALSE # BOD
ActingPrimary: TRUE # BOOLEAN (Acting Primary ObjectServer)
AlertSecurityModel: 0 # INTEGER (Desktop security model)
AllowConnections: TRUE # BOOLEAN (Specifies whether or not no
AllowISQL: TRUE # BOOLEAN (Specifies whether or not isql conn
AllowISQLWrite: TRUE # BOOLEAN (Specifies whether or not modi
AllowTimedRefresh: FALSE # BOOLEAN (Allow desktops to apply t
Auto.Debug: FALSE # BOOLEAN (Automation debug)
Auto.Enabled: TRUE # BOOLEAN (Automation enable)
```

- e. Modify the following property values as shown:

ConfigCryptoAlg: '**AES\_FIPS**'

ConfigKeyFile: '\$NCHOME/etc/security/keys/netcool.keygen'

PA.Password: '@44:3WZnAssKpuWS1grxx0W+g+C0crevWmcJKPbmryCm1m4=@'

SecureMode: **TRUE**



**Note:** Copy the cipher text from the encrypted\_output.txt file and paste it into the PA.Password field.

- f. Save the changes, and exit the gedit utility.

The property file modifications implement two basic changes:

- Configure the ObjectServer to use AES FIPS-compliant encryption
- Configure the ObjectServer to run in Secure mode

These changes are independent of each other.



**Important:** All probes and most gateways do not require a user name and password to connect to an ObjectServer, unless the ObjectServer is configured to run in *secure* mode. Secure mode is not a requirement for SSL or for AES encryption. You can use SSL and AES encryption without setting the ObjectServer to secure mode. By configuring the ObjectServer to run in secure mode, the subsequent exercises can demonstrate the use of user names and passwords in the probe and gateway configurations.

## Probe files properties

You do not generate keys for probes because probes are not servers. Instead, probes use the signed public key of the server to which they connect as a means for both server authentication, and communications encryption.

1. Modify the Simnet probe as follows.

- a. Change to the target directory.

```
cd $OMNIHOME/probes/linux2x86
```

- b. Save a copy of the file before modifications.

```
cp simnet.props simnet.props.orig
```

- c. Edit the file.

```
gedit simnet.props &
```



**Important:** Use your encrypted password from  
**\$NCHOME/etc/security/keys/encrypted\_output.txt** for AuthPassword.

- d. Add the following lines:

```
AuthPassword : '@44:3WZnAsSKpuWS1grxX0W+g+C0crevWmcJKPbmryCmlm4=@'
AuthUserName : 'root'
ConfigCryptoAlg : 'AES_FIPS'
ConfigKeyFile : '$NCHOME/etc/security/keys/netcool.keygen'
SSLServerCommonName : 'AGG_V'
Server : 'NYC_AGG_P'
ServerBackup : 'NYC_AGG_B'
```

```
NHttpd.ListeningPort : 4190
AuthPassword : '@44:3WZnAsSKpuWS1grxX0W+g+C0crevWmcJKPbmryCmlm4=@'
AuthUserName : 'root'
ConfigCryptoAlg : 'AES_FIPS'
ConfigKeyFile : '$NCHOME/etc/security/keys/netcool.keygen'
SSLServerCommonName : 'AGG_V'
Server : 'NYC_AGG_B'
ServerBackup : 'NYC_AGG_B'
```

- e. Save the changes, and exit gedit.

The **Server** property contains the name of the primary ObjectServer. The **ServerBackup** property identifies the backup ObjectServer. **SSLServerCommonName** identifies the *virtual* ObjectServer. Recall that each ObjectServer has its own key. The label (NYC\_AGG\_P, and NYC\_AGG\_B) uniquely identifies each key. But both keys contain the same Common Name (CN) of AGG\_V.

```
ConfigCryptoAlg : 'AES_FIPS'
ConfigKeyFile : '$NCHOME/etc/security/keys/netcool.keygen'
```

Because the password property value is encrypted, the probe must find the key (ConfigKeyFile). Also, the probe must have encryption information (ConfigCryptoAlg).

```
AuthPassword : '@44:ndtg41+/K3+TYbbdVf3LpGoxQODM1M0Ea1GbBM0CGEU=@'
AuthUserName : 'root'
```

These two properties are required because the ObjectServers are configured to run in *Secure Mode*. The probe uses these values to authenticate with the ObjectServer.

## 2. Make the same modifications to the **mttrapd.props**.

```
#####
#Server : "AGG_V"
#RulesFile : "$OMNIHOME/probes/linux2x86/mttrapd.rules"
#RulesFile : "$NCF_RULES_HOME/snmptrap.rules"
NHttpd.EnableHTTP : TRUE
NHttpd.ListeningHostname : "host1.tivoli.edu"
NHttpd.ListeningPort : 4198

AuthPassword : '@44:3WZnAsSKpuWS1grxxX0W+g+CDcrevWmcJKPbmryCmlm4=@'
AuthUserName : 'root'
ConfigCryptoAlg : 'AES_FIPS'
ConfigKeyFile : '$NCHOME/etc/security/keys/netcool.keygen'
SSLServerCommonName: 'AGG_V'
Server : 'NYC_AGG_P'
ServerBackup : 'NYC_AGG_B'
```



**Important:** The probe is configured with Server set to ACC\_V. You must remove, or comment out, the existing line.

## 3. Make the same modifications to the **syslog.props** file.

```
#####
#Server : "AGG_V"
#FifoName : "/var/log/netcool"
#RulesFile : "$OMNIHOME/probes/linux2x86/syslog.rules"
#RulesFile : "$NCF_RULES_HOME/syslog.rules"
NHttpd.EnableHTTP : TRUE
NHttpd.ListeningHostname : "host1.tivoli.edu"
NHttpd.ListeningPort : 4199

AuthPassword : '@44:3WZnAsSKpuWS1grxxX0W+g+CDcrevWmcJKPbmryCmlm4=@'
AuthUserName : 'root'
ConfigCryptoAlg : 'AES_FIPS'
ConfigKeyFile : '$NCHOME/etc/security/keys/netcool.keygen'
SSLServerCommonName: 'AGG_V'
Server : 'NYC_AGG_P'
ServerBackup : 'NYC_AGG_B'
```

## Process Activity properties

### 1. Modify the process activity configuration file as follows:

```
cd $OMNIHOME/etc
cp nco_pa.conf nco_pa.conf.orig
gedit nco_pa.conf
```

In the routing section, incorporate the changes as follows. In the last field, use the netcool user's encrypted password **object00** stored in:

\$NCHOME/etc/security/keys/encrypted\_output.txt:

```
ROUTING TABLE ENTRIES.

'user' - (optional) only required for secure mode PAD on target host
'user' must be member of UNIX group 'ncoadmin'
'password' - (optional) only required for secure mode PAD on target host
use nco_pa_crypt to encrypt.
nco_routing
{
 host 'host1.tivoli.edu' 'HOST1_PA' 'netcool' '@44:3WZnAsSKpuWSlgrxX0W+g+C0crevWmcJKPbmryCmlm4=@'
 host 'host2.tivoli.edu' 'HOST2_PA' 'netcool' '@44:3WZnAsSKpuWSlgrxX0W+g+C0crevWmcJKPbmryCmlm4=@'
}
```

2. Save and close the file.

## Configuring the host2 image

1. Switch to the **host2** image.

The first step creates the key file that is used to encrypt the property values.

2. Run the following command to build the key file:

```
nco_keygen -o $NCHOME/etc/security/keys/netcool.keygen
```

Perform the following steps to encrypt the password **object00** and to store it. You use the encrypted text in several subsequent steps.

3. Encrypt the **object00** password as follows:

```
cd $NCHOME/etc/security/keys
```

```
nco_aes_crypt -o $NCHOME/etc/security/keys/encrypted_output.txt -c AES_FIPS
-k $NCHOME/etc/security/keys/netcool.keygen object00
```

The command places the encrypted text in a file.

4. View the encrypted text.

```
more $NCHOME/etc/security/keys/encrypted_output.txt
```

```
@44:3WZnAsSKpuWSlgrxX0W+g+C0crevWmcJKPbmryCmlm4=@
```

You copy and paste the cipher text into various locations in subsequent steps.



**Hint:** Leave the Terminal window as is, and open a new Terminal window. Use the second Terminal window for the remaining steps.

## ObjectServer properties

1. Modify the ObjectServer property file as follows.

- a. Change to the target directory.

```
cd $OMNIHOME/etc
```

- b. Save a copy of the file before modifications.

```
cp NYC_AGG_B.props NYC_AGG_B.props.orig
```

- c. Edit the file with the gedit utility.

```
gedit NYC_AGG_B.props &
```



**Important:** There are property lines at the top of the file that are commented out. Do not change these lines.

- d. Scroll down in the file and locate the first property line with no comment character.

```
UniqueLog: FALSE # BOO
ActingPrimary: TRUE # BOOLEAN (Acting Primary ObjectServer)
AlertSecurityModel: 0 # INTEGER (Desktop security model)
AllowConnections: TRUE # BOOLEAN (Specifies whether or not no
AllowISQL: TRUE # BOOLEAN (Specifies whether or not isql conn
AllowISQLWrite: TRUE # BOOLEAN (Specifies whether or not modi
AllowTimedRefresh: FALSE # BOOLEAN (Allow desktops to apply t
Auto.Debug: FALSE # BOOLEAN (Automation debug)
Auto.Enabled: TRUE # BOOLEAN (Automation enable)
```

- e. Modify the following property values as shown:

```
ConfigCryptoAlg: 'AES_FIPS'
```

```
ConfigKeyFile: '$NCHOME/etc/security/keys/netcool.keygen'
```

```
PA.Password: '@44:rTIJHizvpc/7I+uHCOpkzpXAUwtbvy007d+Yi9WBO/o=@'
```

```
SecureMode: TRUE
```

**Note:** Copy the cipher text from the encrypted\_output.txt file and paste it into the PA.Password field.

- f. Save the changes, and exit the gedit utility.

The property file modifications implement two basic changes:

- Configure the ObjectServer to use AES FIPS-compliant encryption
- Configure the ObjectServer to run in Secure mode

These changes are independent of each other.

## Probe files properties

1. Modify the Simnet probe as follows.

- a. Change to the target directory.

```
cd $OMNIHOME/probes/linux2x86
```

- b. Save a copy of the file before modifications.

```
cp simnet.props simnet.props.orig
```

- c. Edit the file.

```
gedit simnet.props &
```



**Important:** Use your encrypted password from  
**\$NCHOME/etc/security/keys/encrypted\_output.txt** for AuthPassword.

- d. Add the following lines:

```
AuthPassword : '@44:rTIJHizvpc/7I+uHCOpkzpXAUwtbvy007d+Yi9WBO/o=@'
AuthUserName : 'root'
ConfigCryptoAlg : 'AES_FIPS'
ConfigKeyFile : '$NCHOME/etc/security/keys/netcool.keygen'
SSLSERVERCommonName : 'AGG_V'
Server : 'NYC_AGG_P'
ServerBackup : 'NYC_AGG_B'
```

```
NHttpd.ListeningPort : 4190
AuthPassword : '@44:rTIJHizvpc/7I+uHCOpkzpXAUwtbvy007d+Yi9WBO/o=@'
AuthUserName : 'root'
ConfigCryptoAlg : 'AES_FIPS'
ConfigKeyFile : '$NCHOME/etc/security/keys/netcool.keygen'
SSLSERVERCommonName : 'AGG_V'
Server : 'NYC_AGG_P'
ServerBackup : 'NYC_AGG_B'
```

- e. Save the changes, and exit gedit.

## Process Activity properties

1. Modify the process activity configuration file as follows:

```
cd $OMNIHOME/etc
cp nco_pa.conf nco_pa.conf.orig
gedit nco_pa.conf
```

In the routing section, incorporate the changes as follows. In the last field, use the netcool user's encrypted password **object00** stored in:

\$NCHOME/etc/security/keys/encrypted\_output.txt:

```
Routing Table Entries.
#
'user' - (optional) only required for secure mode PAD on target host
'user' must be member of UNIX group 'ncoadmin'
'password' - (optional) only required for secure mode PAD on target host
use nco_pa_encrypt to encrypt.
nco_routing
{
 host 'host1.tivoli.edu' 'HOST1_PA' 'netcool' '@44:rTIJHizvpc/7I+uHCOpkzpXAUwtbvy007d+Yi9WBO/o=@'
 host 'host2.tivoli.edu' 'HOST2_PA' 'netcool' '@44:rTIJHizvpc/7I+uHCOpkzpXAUwtbvy007d+Yi9WBO/o=@'
}
```

- Save and close the file.

## Bidirectional gateway properties

- Modify the bidirectional gateway.

```
cd $OMNIHOME/etc
cp NYC_AGG_GATE.props NYC_AGG_GATE.props.orig
gedit NYC_AGG_GATE.props &
```

- Add the following lines to the end of the file.

```
Gate.ObjectServerA.CommonNames : 'AGG_V'
Gate.ObjectServerA.Username : 'root'
Gate.ObjectServerA.Password :
'@44:rTIJHizvpc/7I+uHCOpkzpXAUwtbvy007d+Yi9WBO/o=@'
Gate.ObjectServerB.CommonNames : 'AGG_V'
Gate.ObjectServerB.Username : 'root'
Gate.ObjectServerB.Password :
'@44:rTIJHizvpc/7I+uHCOpkzpXAUwtbvy007d+Yi9WBO/o=@'

Gate.UsePamAuth : TRUE
ConfigCryptoAlg : 'AES_FIPS'
ConfigKeyFile : '$NCHOME/etc/security/keys/netcool.keygen'

Gate.ObjectServerA.CommonNames : 'AGG_V'
Gate.ObjectServerA.Username : 'root'
Gate.ObjectServerA.Password : '@44:rTIJHizvpc/7I+uHCOpkzpXAUwtbvy007d+Yi9WBO/o=@'
Gate.ObjectServerB.CommonNames : 'AGG_V'
Gate.ObjectServerB.Username : 'root'
Gate.ObjectServerB.Password : '@44:rTIJHizvpc/7I+uHCOpkzpXAUwtbvy007d+Yi9WBO/o=@'

Gate.UsePamAuth : TRUE
ConfigCryptoAlg : 'AES_FIPS'
ConfigKeyFile : '$NCHOME/etc/security/keys/netcool.keygen'
```



**Important:** The two password values are the same text. The gateway decrypts the password strings before it connects to each ObjectServer. The password value is passed to the ObjectServer in clear text, which is one reason to use SSL to access the ObjectServer.

- Save and close the file.

## Database gateway properties

- Edit the JDBC\_GATE.props file in a similar fashion.

```
Gate.RdrWtr.CommonNames: 'AGG_V'
Gate.RdrWtr.Username: 'root'
Gate.RdrWtr.Password: '@44:rTIJHizvpc/7I+uHCOpkzpXAUwtbvy007d+Yi9WBO/o=@'
Gate.UsePamAuth : TRUE
ConfigCryptoAlg : 'AES_FIPS'
ConfigKeyFile : '$NCHOME/etc/security/keys/netcool.keygen'

Description name - this value appears in the list of ObjectServer connections
Gate.RdrWtr.Description : 'JDBC Gateway'

Gate.RdrWtr.CommonNames : 'AGG_V'
Gate.RdrWtr.Username : 'root'
Gate.RdrWtr.Password : '@44:rTIJHizvpc/7I+uHCOpkzpXAUwtbvy007d+Yi9WBO/o=@'
Gate.UsePamAuth : TRUE
ConfigCryptoAlg : 'AES_FIPS'
ConfigKeyFile : '$NCHOME/etc/security/keys/netcool.keygen'
```

- Save and close the file.

# Exercise 4 Validating SSL with FIPS compliance

## Verifying the encryption configuration

The configuration changes are in place. However, nothing changes until the components are stopped, and started.

- Switch to the **host1** image.
- Stop the process activity daemon, and all components.

```
nco_pa_shutdown -server HOST1_PA -password object00
```

```
Connected To PA Server [HOST1_PA] Shutdown Options :-
1) Shutdown Server leaving managed processes running.
2) Shutdown Server and stop all managed processes.
3) Exit shutdown interface.
Select Option [1-3] 2
```

- Enter **2** to shut down process activity.

- Start the process activity daemon.

```
nco_pad -name HOST1_PA -authenticate PAM
-keyfile $NCHOME/etc/security/keys/netcool.keygen -cryptalgorith AES_FIPS
```



**Important:** Note the addition of the *keyfile* and *cryptalgorith*m parameters to the command.

##### 5. Verify component status.

```
nco_pa_status -server HOST1_PA -password object00
```

| Service Name | Process Name       | Hostname                | User | Status  | PID  |
|--------------|--------------------|-------------------------|------|---------|------|
| Core         | MasterObjectServer | host1.tivoli.edunetcool |      | RUNNING | 5882 |
|              | SyslogProbe        | host1.tivoli.edunetcool |      | RUNNING | 5883 |
|              | SnmpProbe          | host1.tivoli.edunetcool |      | RUNNING | 5884 |
|              | SimnetProbe        | host1.tivoli.edunetcool |      | RUNNING | 5885 |

Verify that all components are running.

The fact that the probes are running verifies their revised encryption settings. Because the ObjectServers are running in *secure mode*, the probes must specify a user name and password to connect. The password values in the configuration files are encrypted.

##### 1. Switch to the **host2** image.

##### 2. Stop the process activity daemon, and all components.

```
nco_pa_shutdown -server HOST2_PA -password object00
```

```
Connected To PA Server [HOST1_PA] Shutdown Options :-
1) Shutdown Server leaving managed processes running.
2) Shutdown Server and stop all managed processes.
3) Exit shutdown interface.

Select Option [1-3] 2
```

##### 3. Enter **2** to shut down process activity.

##### 4. Start the process activity daemon.

```
nco_pad -name HOST2_PA -authenticate PAM
-keyfile $NCHOME/etc/security/keys/netcool.keygen -cryptalgorith AES_FIPS
```



**Important:** Note the addition of the *keyfile* and *cryptalgorith*m parameters to the command.

##### 5. Verify component status.

```
nco_pa_status -server HOST2_PA -password object00
```

| Service Name | Process Name       | Hostname                | User | Status  | PID   |
|--------------|--------------------|-------------------------|------|---------|-------|
| Core         | BackupObjectServer | host2.tivoli.edunetcool |      | RUNNING | 29824 |
|              | BackupGateway      | host2.tivoli.edunetcool |      | RUNNING | 32109 |
|              | ArchiveGateway     | host2.tivoli.edunetcool |      | RUNNING | 410   |
|              | LondonObjectServer | host2.tivoli.edunetcool |      | RUNNING | 29827 |
|              | SimnetProbe        | host2.tivoli.edunetcool |      | RUNNING | 29828 |

Verify that all components are running.

The fact that the gateways are running verifies their revised encryption settings. Because the ObjectServers are running in *secure mode*, the gateways must specify a user name, and password to connect. The password values in the configuration files are encrypted.

The previous steps verify that encryption is configured correctly for all components.

## Verifying the SSL configuration

Ensure the ObjectServers are running in SSL mode.

1. Switch to the **host1** image.
2. Open the Netcool/OMNIbus Administrative utility.

nco\_config &



**Note:** You might be prompted to reimport connection data. Choose **Yes > Finish**.

Observe the ObjectServer Report window.

| Name                | Host             | Port | SSL     |
|---------------------|------------------|------|---------|
| AGG_V (backup 01)   | host1.tivoli.edu | 4110 | ✓ true  |
| AGG_V (backup 02)   | host2.tivoli.edu | 4110 | ✓ true  |
| AGG_V (backup 03)   | host2.tivoli.edu | 4100 | ✗ false |
| AGG_V (primary)     | host1.tivoli.edu | 4100 | ✗ false |
| NYC_AGG_B (backup)  | host2.tivoli.edu | 4110 | ✓ true  |
| NYC_AGG_B (primary) | host2.tivoli.edu | 4100 | ✗ false |
| NYC_AGG_P (backup)  | host1.tivoli.edu | 4110 | ✓ true  |
| NYC_AGG_P (primary) | host1.tivoli.edu | 4100 | ✗ false |

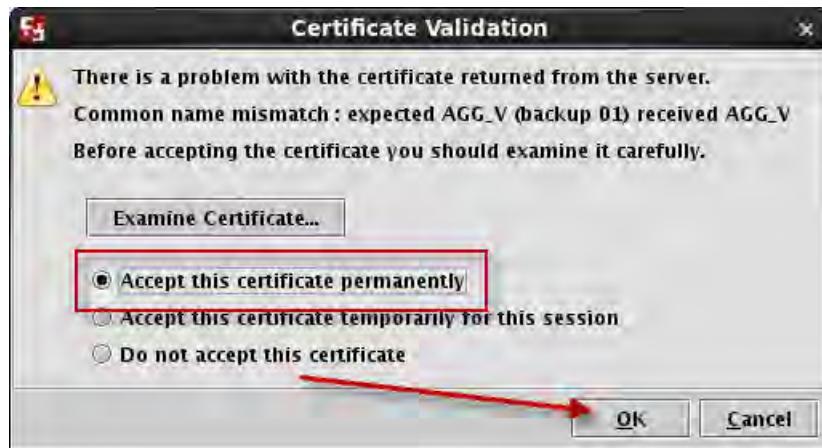
There are multiple entries for each ObjectServer. One entry identifies the non-SSL port, and the other entry identifies the SSL port. The entries appear because each ObjectServer is configured to support access through two different ports. One being SSL, and the other being non-SSL.

3. Connect to the **AGG\_V (backup 01)** ObjectServer as the **root** user with password **object00** through the SSL port **4110**.

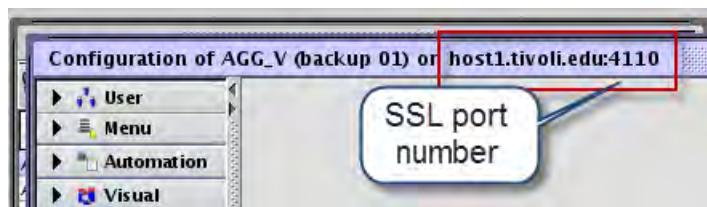
| Name                | Host             | Port | SSL     |
|---------------------|------------------|------|---------|
| AGG_V (backup 01)   | host1.tivoli.edu | 4110 | ✓ true  |
| AGG_V (backup 02)   | host2.tivoli.edu | 4110 | ✓ true  |
| AGG_V (backup 03)   | host2.tivoli.edu | 4100 | ✗ false |
| AGG_V (primary)     | host1.tivoli.edu | 4100 | ✗ false |
| NYC_AGG_B (backup)  | host2.tivoli.edu | 4110 | ✓ true  |
| NYC_AGG_B (primary) | host2.tivoli.edu | 4100 | ✗ false |
| NYC_AGG_P (backup)  | host1.tivoli.edu | 4110 | ✓ true  |

A certificate warning window opens.

- Click **Accept this certificate permanently** and click **OK**.



You are connected through the SSL port.



- Close the ObjectServer window.
- Repeat the previous steps to verify access through the other SSL ports.
- Switch to the **host2** image, and repeat the previous steps.

**Note:** The host2 image uses a separate key database and separate keys. When you connect to the ObjectServers over the SSL ports, you verify that the keys are correct.

The interfaces file currently contains two port numbers for each ObjectServer. You verified that you can connect to the primary ObjectServer through an SSL port with the Netcool/OMNIbus Administrator utility. What is not verified is whether the probes and gateways can connect through an SSL port. The only way to verify that SSL access works is to remove the non-SSL port numbers from the interfaces file.

**Important:** In a production environment, probes generally run on remote servers and have their own interfaces file. In a production environment, you can leave the interfaces file for the ObjectServer as configured, which configures the ObjectServer to support SSL, and non-SSL access. You modify the interfaces file on the probe server, and remove the non-SSL ports. Then, the probe can only access the ObjectServer over the SSL port.

- Switch to the **host1** image.

9. Stop process activity and all processes.

```
nco_pa_shutdown -server HOST1_PA -password object00
```

Enter **2** to stop everything.

10. Start the Server Editor utility.

```
nco_xigen &
```

11. Click the first entry for **AGG\_V**. Clear the non-SSL port value. **Click Update**.

12. Repeat the previous step to remove the non-SSL port numbers from all of the ObjectServers.

| Server       | Hostname         | Port | SSL |
|--------------|------------------|------|-----|
| AGG_V        | host1.tivoli.edu | 4110 |     |
| Backup1:     | host2.tivoli.edu | 4110 |     |
| HOST1_PA     | host1.tivoli.edu | 4200 |     |
| HOST2_PA     | host2.tivoli.edu | 4200 |     |
| NYC_AGG_B    | host2.tivoli.edu | 4110 |     |
| NYC_AGG_GATE | host2.tivoli.edu | 4300 |     |
| NYC_AGG_P    | host1.tivoli.edu | 4110 |     |

13. Click **Apply** and click **Close**.

14. Start the process activity daemon.

```
nco_pad -name HOST1_PA -authenticate PAM
-keyfile $NCHOME/etc/security/keys/netcool.keygen -cryptalgorith AES_FIPS
```

15. Verify component status.

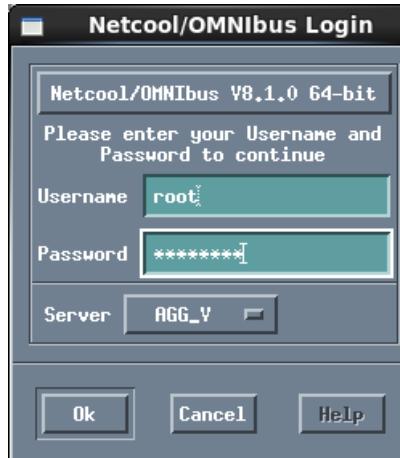
```
nco_pa_status -server HOST1_PA -password object00
```

| Service Name | Process Name       | Hostname                | User | Status  | PID   |
|--------------|--------------------|-------------------------|------|---------|-------|
| Core         | MasterObjectServer | host1.tivoli.edunetcool |      | RUNNING | 10746 |
|              | SyslogProbe        | host1.tivoli.edunetcool |      | RUNNING | 10747 |
|              | SrmpProbe          | host1.tivoli.edunetcool |      | RUNNING | 10748 |
|              | SimnetProbe        | host1.tivoli.edunetcool |      | RUNNING | 10749 |

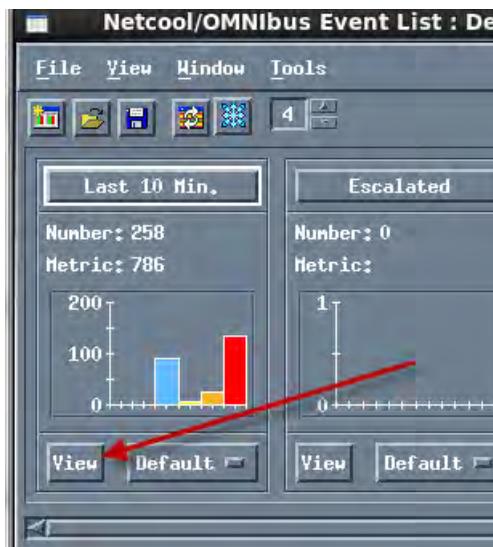
The probes are running which verifies that they are able to connect to the ObjectServers through an SSL port.

16. Start the native desktop, and connect to the **AGG\_V** ObjectServer as **root** with password **object00**.

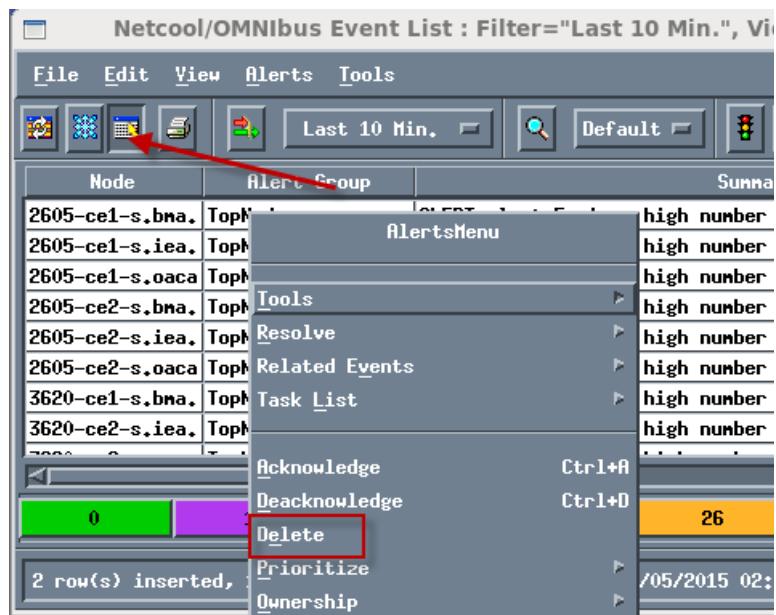
```
nco_event &
```



17. Select the **Last 10 Min** monitor box.



18. Click the icon to select all events. Right-click, and select **Delete**.



After a short time, events appear in the event list, which verifies that the probes can send events to the ObjectServer over SSL.

| Node            | Alert Group | Summary                   |
|-----------------|-------------|---------------------------|
| nsm123-c3.iea.g | Systems     | Machine has gone offline  |
| nsm123-c2.oaca  | Stats       | Diskspace alert           |
| nsm123-c2.iea.g | Stats       | Diskspace alert           |
| LON-dca.gov     | Link        | Port failure : port reset |
| 2605-ce1-s.oaca | Link        | Link Up on port           |
| 2605-ce2-s.oaca | Link        | Link Up on port           |
| BRU-CPE-dca.go  | Link        | Link Up on port           |

19. Close the event list, and exit the native client.

20. Switch to the **host2** image.

21. Stop process activity and all processes.

```
nco_pa_shutdown -server HOST2_PA -password object00
```

Enter **2** to stop everything.

22. Start the Server Editor utility.

```
nco_xigen &
```

23. Click the first entry for **AGG\_V**. Clear the non-SSL port value. **Click Update**.

24. Repeat the previous step to remove the non-SSL port numbers from all of the ObjectServers.

| Server       | Hostname         | Port | SSL |
|--------------|------------------|------|-----|
| AGG_V        | host1.tivoli.edu | 4110 |     |
| Backup1:     | host2.tivoli.edu | 4110 |     |
| HOST1_PA     | host1.tivoli.edu | 4200 |     |
| HOST2_PA     | host2.tivoli.edu | 4200 |     |
| NYC_AGG_B    | host2.tivoli.edu | 4110 |     |
| NYC_AGG_GATE | host2.tivoli.edu | 4300 |     |
| NYC_AGG_P    | host1.tivoli.edu | 4110 |     |

25. Click **Apply** and click **Close**.

26. Start the process activity daemon.

```
nco_pad -name HOST2_PA -authenticate PAM -keyfile
$SNHOME/etc/security/keys/netcool.keygen -cryptalgorith AES_FIPS
```

27. Verify component status.

```
nco_pa_status -server HOST2_PA -password object00
```

| [netcool@host1 log]# nco_pa_status -server HOST2_PA -password object00 |                    |                         |         |         |       |
|------------------------------------------------------------------------|--------------------|-------------------------|---------|---------|-------|
| Service Name                                                           | Process Name       | Hostname                | User    | Status  | PID   |
| Core                                                                   | BackupObjectServer | host2.tivoli.edunetcool | netcool | RUNNING | 13362 |
|                                                                        | BackupGateway      | host2.tivoli.edunetcool | netcool | RUNNING | 13363 |
|                                                                        | ArchiveGateway     | host2.tivoli.edunetcool | netcool | RUNNING | 13364 |
|                                                                        | LondonObjectServer | host2.tivoli.edunetcool | netcool | RUNNING | 13365 |
|                                                                        | SimnetProbe        | host2.tivoli.edunetcool | netcool | RUNNING | 13366 |

The gateways are running which verifies that they are able to connect to the ObjectServers through an SSL port.

## Modifying the process activity startup script

When configured for SSL access, the process activity component requires extra command line parameters. The class images contain a script that starts process activity when the server starts. You must modify that script and add the extra parameters.

1. Switch to the **host1** image.

2. Change to the **root** user.

```
su -
Password: object00
```

3. Modify the file as follows.

- a. Change to the target directory.

```
cd /etc/init.d
```

- b. Open the file with the gedit utility.

```
gedit nco &
```

- c. Locate the following section:

```
."
if ["$SECURE" = "y"]; then
 su - netcool -c "${OMNIHOME}/bin/nco_pad -name ${NCO_PA} -
te PAM -secure > /dev/null 2> /dev/null"
else
 su - netcool -c "${OMNIHOME}/bin/nco_pad -name ${NCO_PA} -
te PAM > /dev/null 2> /dev/null"
fi
```

- d. Change the two lines as shown here:

```
su - netcool -c "${OMNIHOME}/bin/nco_pad -name ${NCO_PA} -authenticate PAM
-secure -keyfile $NCHOME/etc/security/keys/netcool.keygen -cryptalgorith
AES_FIPS> /dev/null 2> /dev/null"
```

```
su - netcool -c "${OMNIHOME}/bin/nco_pad -name ${NCO_PA} -authenticate PAM
-keyfile $NCHOME/etc/security/keys/netcool.keygen -cryptalgorith AES_FIPS>
/dev/null 2> /dev/null"
```

```
."
name ${NCO_PA} -authenticate PAM -secure -keyfile $NCHOME/etc/security/keys/netcool.keygen -
cryptalgorith AES_FIPS> /dev/null 2> /dev/null"
else
 su - netcool -c "${OMNIHOME}/bin/nco_pad -
name ${NCO_PA} -authenticate PAM -keyfile $NCHOME/etc/security/keys/netcool.keygen -
cryptalgorith AES_FIPS> /dev/null 2> /dev/null"
fi
```

- e. Save the file, and exit the gedit utility.

4. Stop process activity as the **root** user.

```
./nco stop
```

5. Start process activity as the **root** user.

```
./nco start
```

6. Exit the **root** user back to the **netcool** user.

```
exit
```

7. Verify the status of process activity.

```
nco_pa_status -server HOST1_PA -password object00
```

| Service Name | Process Name       | Hostname                | User | Status  | PID   |
|--------------|--------------------|-------------------------|------|---------|-------|
| Core         | MasterObjectServer | host1.tivoli.edunetcool |      | RUNNING | 12086 |
|              | SyslogProbe        | host1.tivoli.edunetcool |      | RUNNING | 12087 |
|              | SnmpProbe          | host1.tivoli.edunetcool |      | RUNNING | 12088 |
|              | SimnetProbe        | host1.tivoli.edunetcool |      | RUNNING | 12089 |

8. Repeat the steps on **host2**.

To summarize what is accomplished:

- ObjectServers are configured to use AES FIPS 140-2 compliant encryption
- ObjectServers are configured to support access through only SSL ports
- ObjectServers are configured to run in Secure Mode
- Probes are configured to connect to the ObjectServers through SSL ports
- Gateways are configured to connect to the ObjectServers through SSL ports

In many production environments, the Web GUI server does not require SSL encrypted connections to the ObjectServer. To enable SSL access for Web GUI, more modifications are required. The changes are documented in the Web GUI administration guide.



## 10 Multitiered architecture exercises

In this unit, you learn how to use the Initial Configuration Wizard to create, and deploy a multi-tiered architecture.

### Exercise 1 Preparing for the deployment

The class images are configured with ObjectServers, probes, and gateways. These components are deployed in a *single layer* architecture, which is distributed across two images. In the following exercise, you replace the single layer architecture with a three-layer architecture. In a production environment, this architecture is distributed across multiple servers. In this exercise, you use two servers.

You must stop the existing components on both images before you create the new configuration.

#### Stopping components on host1

1. Switch to the **host1** image.
2. Open a Terminal window if necessary.
3. Enter the following command to stop process activity.

```
nco_pa_shutdown -server HOST1_PA -password object00
Connected To PA Server [HOST1_PA] Shutdown Options :-
```

- 1) Shutdown Server leaving managed processes running.
- 2) Shutdown Server and stop all managed processes.
- 3) Exit shutdown interface.

```
Select Option [1-3] 2
```

4. Enter **2** to stop the agent and all processes.

## Stopping components on host2

1. Switch to the **host2** image.
2. Open a Terminal window if necessary.
3. Enter the following command to stop process activity.

```
nco_pa_shutdown -server HOST2_PA -password object00
Connected To PA Server [HOST2_PA] Shutdown Options :-
```

- 1) Shutdown Server leaving managed processes running.
- 2) Shutdown Server and stop all managed processes.
- 3) Exit shutdown interface.

Select Option [1-3] **2**

4. Enter **2** to stop the agent, and all processes.
5. Stop Dashboard Application Services Hub as follows:

```
cd /opt/IBM/JazzSM/profile/bin
. ./stopServer.sh server1 -username smadmin -password object00
```

Wait for the server to stop.

## Exercise 2 Creating the configuration

In this exercise, you use the Initial Configuration Wizard to define the three layer architecture. Then, you use the wizard to deploy part of the components on the host1 image. You use the wizard again to deploy the remaining components on host2.

# Creating the configuration on host1

## Removing old configuration files

The wizard creates Netcool/OMNibus components when you apply the configuration. The wizard does not overwrite existing components. Before you run the wizard, you must remove the following files:

```
/opt/IBM/tivoli/netcool/etc/omni.dat
/opt/IBM/tivoli/netcool/omnibus/etc/nco_pa.conf
```

1. Switch to the **host1** image.
2. Remove the two files.

```
rm /opt/IBM/tivoli/netcool/etc/omni.dat
rm /opt/IBM/tivoli/netcool/omnibus/etc/nco_pa.conf
```



**Important:** If you do not remove these files before you run the wizard, the wizard generates an error, and fails to create the components.

In the previous unit on security, you configured Netcool/OMNibus to use FIPS 140-2 encryption. You must remove some configuration files to disable FIPS 140-2 encryption.

3. Remove the FIPS configuration file.
- ```
rm /opt/IBM/tivoli/netcool/etc/security/fips.conf
```

Creating the three-layer configuration

1. Start the wizard.
2. Click **Next**.
3. Leave the option to create a new configuration, and click **Next**.
4. Select **Aggregation backup**.
5. Enter **1** for the number of Collection ObjectServers.
6. Select **Collection backup**.

7. Enter **2** for the number of Display ObjectServers. Click **Next**.

Multitier ObjectServers

Tivoli Netcool/OMNibus can be deployed in a multitiered configuration to increase performance and event handling capacity. Select one or more Collection or Display ObjectServers only if your operating environment requires them.

Aggregation backup

The Aggregation layer is the central point where events from all sources are aggregated and processed. A primary Aggregation ObjectServer is always created. A backup Aggregation ObjectServer is strongly recommended in production environments.

Primary Collection ObjectServers

Collection ObjectServers collect incoming events from probes and forward them to the Aggregation layer.

Collection backup

Include a backup Collection ObjectServer for every primary Collection ObjectServer.

Display ObjectServers

Display ObjectServers ease the load on the Aggregation layer by forwarding events to clients such as the Web GUI. If Display ObjectServers are to be deployed a minimum of 2 are recommended for resiliency.

8. Enter **host1.tivoli.edu** for the host.

9. Enter **/opt/IBM/tivoli/netcool** for NCHOME, and click **Add**.

Enter a host name. To ensure that all computers in your deployment names (FQDN) such as myhost.example.com.

* Host

Enter the Tivoli Netcool/OMNibus installation directory. The environment variable.

* NCHOME



10. Repeat the previous steps to add an entry for **host2.tivoli.edu**. Click **Next**.

host1.tivoli.edu - /opt/IBM/tivoli/netcool
host2.tivoli.edu - /opt/IBM/tivoli/netcool

11. Accept the default configuration for the two process agents, and click **Next**.

Computer	host1.tivoli.edu:/opt/IBM/tivoli/netcool
PA name	HOST1_PA
Name prefix	HOST1
PA port	4200
Computer	host2.tivoli.edu:/opt/IBM/tivoli/netcool
PA name	HOST2_PA
Name prefix	HOST2
PA port	4200

12. Select **host1.tivoli.edu** for the primary Aggregate ObjectServer.

13. Enter **MULTI** for the prefix.

14. Enter **4110** for the port.

ObjectServer name	MULTI_AGG_P
Computer	host1.tivoli.edu - /opt/IBM/tivoli/r
Name prefix	MULTI
Server port	4110

15. Select **host2.tivoli.edu** for the backup Aggregate ObjectServer.

16. Enter **MULTI** for the prefix.

17. Enter **4110** for the port.

Backup Aggregate

ObjectServer name MULTI_AGG_B

* Computer host2.tivoli.edu - /opt/IBM/tivoli/r

Name prefix MULTI

* Server port 4110

18. Scroll down, and enter **MULTI** for the gateway prefix. Leave the port as **4300**. Click **Next**.

Gateway name MULTI_AGG_GATE

Gateway prefix MULTI

* Gateway port 4300

19. Select **host1.tivoli.edu** for the primary Collection 1 ObjectServer.

20. Enter **MULTI** for the prefix.

21. Enter **4120** for the port.

Primary Collection 1

ObjectServer name MULTI_COL_P_1

* Computer host1.tivoli.edu - /opt/IBM/tivoli/r

Name prefix MULTI

* Server port 4120

22. Scroll down, and enter **MULTI** for the gateway prefix. Enter **4301** for the port number.

Gateway name MULTI_C_TO_A_GATE_P_1

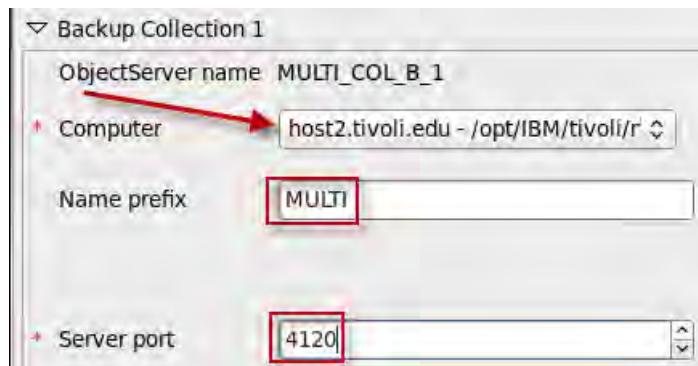
Gateway prefix MULTI

* Gateway port 4301

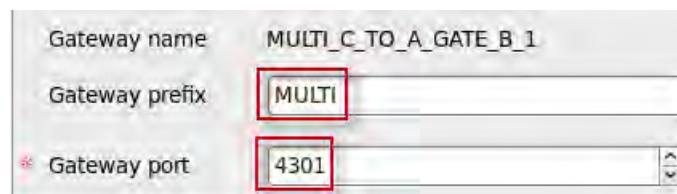
23. Select **host2.tivoli.edu** for the backup Collection 1 ObjectServer.

24. Enter **MULTI** for the prefix.

25. Enter **4120** for the port.



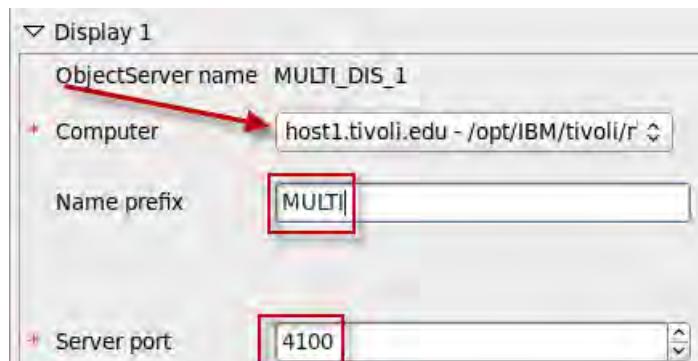
26. Scroll down, and enter **MULTI** for the gateway prefix. Leave the port number as **4301**, and click **Next**.



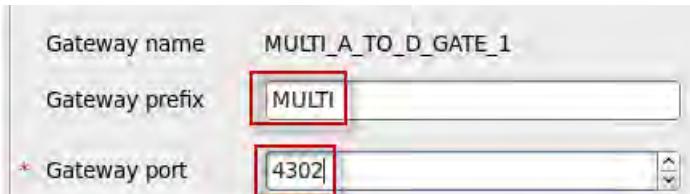
27. Select **host1.tivoli.edu** for the Display 1 ObjectServer.

28. Enter **MULTI** for the prefix.

29. Leave the port as **4100**.



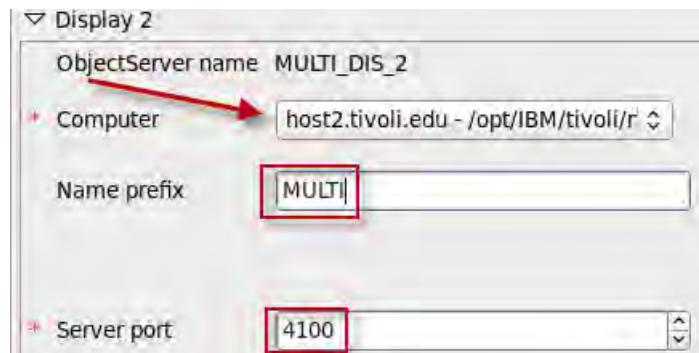
30. Scroll down, and enter **MULTI** for the gateway prefix. Enter **4302** for the port number.



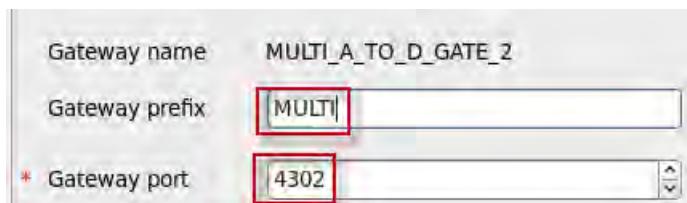
31. Select **host2.tivoli.edu** for the Display 2 ObjectServer.

32. Enter **MULTI** for the prefix.

33. Leave the port as 4100.

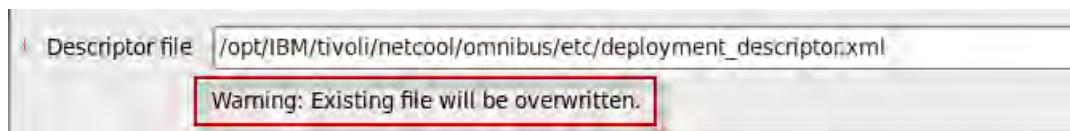


34. Scroll down, and enter **MULTI** for the gateway prefix. Leave **4302** for the port number. Click **Next**.



35. Review the summary information, and click **Next**.

36. Leave the default file name, and path for the descriptor file. Click **Next**.



A warning message appears because there is an existing file.

f. Review the list of components for host1, and click **Next**.

Component	Action
Interfaces file	Update /opt/IBM/tivoli/netcool/etc/omni.dat and run nco_igen
Process agent	Create configuration for this computer
Process agent	Add entry for ObjectServer MULTI_DIS_1
Process agent	Add entry for ObjectServer MULTI_COL_P_1
Process agent	Add entry for ObjectServer MULTI_DIS_2
Process agent	Add entry for ObjectServer gateway MULTI_C_TO_A_GATE_P_1
Process agent	Add entry for ObjectServer gateway MULTI_A_TO_D_GATE_1
ObjectServer	Create properties file and database for MULTI_DIS_1
ObjectServer	Create properties file and database for MULTI_COL_P_1
ObjectServer	Create properties file and database for MULTI_DIS_2
ObjectServer gateway	Create configuration files for MULTI_C_TO_A_GATE_P_1
ObjectServer gateway	Create configuration files for MULTI_A_TO_D_GATE_1

37. Verify success, click **Exit** to close the wizard.

Successful application

The current configuration has been successfully applied to this computer. Follow the instructions and checklist contained in the file below to set up the other computers. Click **Exit** to leave the wizard.

/opt/IBM/tivoli/netcool/omnibus/etc/icw_instructions.txt

Modifying the process agent configuration

The wizard assumes that all components run as the **root** user. You must modify the process activity configuration file, and change the commands to use the **netcool** user.

1. Modify the process activity configuration file as follows.

- a. Change to the correct directory.

```
cd /opt/IBM/tivoli/netcool/omnibus/etc
```

- b. Edit the file with the gedit utility.

```
gedit nco_pa.conf
```

- c. Locate the first line that contains **runs as 0**.

```
Command '$OMNIHOME/bin/nco_objserv -name MULTI_AGG_P -pa HOST1_PA' run as 0
```

- d. Change the line to use **run as 500**.

```
Command '$OMNIHOME/bin/nco_objserv -name MULTI_AGG_P -pa HOST1_PA' run as  
500
```

- e. Repeat this step, and change all references of run as 0 to **run as 500**.



Note: You must change five lines.

- f. Save the modifications, and exit the gedit utility.

2. Enter the following command to verify that all lines are modified.

```
grep "run as" nco_pa.conf
```

Starting the components

1. Enter the following command to start process activity.

```
nco_pad -name HOST1_PA
```

- Verify the component status as follows:

```
nco_pa_status -server HOST1_PA -password object00
```

Service Name	Process Name	Hostname	User	Status	PID
Core	MasterObjectServer	host1.tivoli.edunetcool		RUNNING	11592
	PrimaryCollectionObjectServer_1	host1.tivoli.edunetcool		RUNNING	11593
	DisplayObjectServer_1	host1.tivoli.edunetcool		RUNNING	11594
	PrimaryCollectionGateway_1	host1.tivoli.edunetcool		RUNNING	11595
	DisplayGateway_1	host1.tivoli.edunetcool		RUNNING	11596

The host1 image contains three ObjectServers, and two gateways. All components are running.

The new configuration uses a display ObjectServer. The wizard created this ObjectServer with a root user and no password.

- Modify the **root** user, and set the password to **object00**.

```
nco_sql -server MULTI_DIS_1 -user root -password ''  
1> alter user 'root' set password 'object00';  
2> go  
1> quit
```

- Verify the **root** user password.

```
nco_sql -server MULTI_DIS_1 -user root -password 'object00'  
1> quit
```



Important: The Web GUI component is configured to connect to an ObjectServer on host1 with port number 4100. The component is also configured to connect to an ObjectServer on host2 with port number 4100. The component is configured to use the root user with password object00 for this access. The display ObjectServers are configured to use port 4100. You add the object00 password to these ObjectServers to allow the Web GUI component access.

Creating the configuration on host2

Removing old configuration files

The wizard creates Netcool/OMNibus components when you apply the configuration. The wizard does not overwrite existing components. Before you run the wizard, you must remove the following files:

```
/opt/IBM/tivoli/netcool/etc/omni.dat  
/opt/IBM/tivoli/netcool/omnibus/etc/nco_pa.conf
```

- Switch to the **host2** image.
- Remove the two files.

```
rm /opt/IBM/tivoli/netcool/etc/omni.dat  
rm /opt/IBM/tivoli/netcool/omnibus/etc/nco_pa.conf
```



Important: If you do not remove these files before you run the wizard, the wizard generates an error, and fails to create the components.

In the previous unit on security, you configured Netcool/OMNibus to use FIPS 140-2 encryption. You must remove some configuration files to disable FIPS 140-2 encryption.

3. Remove the FIPS configuration file.

```
rm /opt/IBM/tivoli/netcool/etc/security/fips.conf
```

Creating the three-layer configuration

1. Retrieve the wizard descriptor file from host1 as follows.

- a. Change to the local directory.

```
cd /opt/IBM/tivoli/netcool/omnibus/etc/
```

- b. Connect to host1 with SFTP.

```
sftp host1
```

- c. Change to the remote directory location.

```
sftp> cd /opt/IBM/tivoli/netcool/omnibus/etc/
```

- d. Retrieve the file.

```
sftp> get deployment_descriptor.xml
```

- e. Exit SFTP.

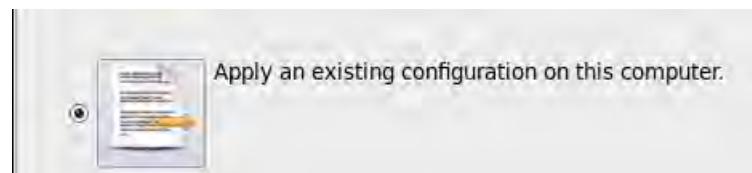
```
sftp> quit
```

2. Start the wizard.

```
nco_icw &
```

3. Click **Next**.

4. Select the option to apply an existing configuration, and click **Next**.



5. Browse to the location of the descriptor file, and click **Next**.

Load an existing configuration

Load an existing configuration by selecting a deployment descriptor file. The deployment descriptor file defines the components and their properties for a previously created configuration.

Filename /opt/IBM/tivoli/netcool/omnibus/etc/deployment_descriptor.xml

6. Review the list of components, and click **Next**.

Apply the configuration

The components listed here will be configured on this computer. Click **Next** to apply the configuration now.

Component	Action
Interfaces file	Update /opt/IBM/tivoli/netcool/etc/omni.dat and run nco_igen
Process agent	Create configuration for this computer
Process agent	Add entry for ObjectServer MULTI_AGG_B
Process agent	Add entry for ObjectServer MULTI_COL_B_1
Process agent	Add entry for ObjectServer MULTI_DIS_2
Process agent	Add entry for ObjectServer gateway MULTI_AGG_GATE
Process agent	Add entry for ObjectServer gateway MULTI_C_TO_A_GATE_B_1
Process agent	Add entry for ObjectServer gateway MULTI_A_TO_D_GATE_2
ObjectServer	Create properties file and database for MULTI_AGG_B
ObjectServer	Create properties file and database for MULTI_COL_B_1
ObjectServer	Create properties file and database for MULTI_DIS_2
ObjectServer gateway	Create configuration files for MULTI_AGG_GATE
ObjectServer gateway	Create configuration files for MULTI_C_TO_A_GATE_B_1
ObjectServer gateway	Create configuration files for MULTI_A_TO_D_GATE_2

7. Verify success, and click **Exit**.

Successful application

The current configuration has been successfully applied to this computer. Press **Exit** to leave the wizard.

Modifying the process agent configuration

The wizard assumes that all components run as the **root** user. You must modify the process activity configuration file, and change the commands to use the **netcool** user.

1. Modify the process activity configuration file as follows.

- a. Change to the correct directory.

```
cd /opt/IBM/tivoli/netcool/omnibus/etc
```

- b. Edit the file with the gedit utility.

```
gedit nco_pa.conf
```

- c. Locate the first line that contains **runs as 0**.

Command '\$OMNIHOME/bin/nco_objserv -name MULTI_ AGG_B -pa HOST2_PA' **run as 0**

- d. Change the line to use **run as 500**.

Command '\$OMNIHOME/bin/nco_objserv -name MULTI_ AGG_B -pa HOST2_PA' **run as 500**

- e. Repeat this step, and change all references of run as 0 to **run as 500**.



Note: You must change six lines.

- f. Save the modifications, and exit the gedit utility.

2. Enter the following command to verify that all lines are modified.

```
grep "run as" nco_pa.conf
```

Starting the components

1. Enter the following command to start process activity.

```
nco_pad -name HOST2_PA
```

2. Verify the component status as follows:

```
nco_pa_status -server HOST2_PA -password object00
```

Service Name	Process Name	Hostname	User	Status	PID
Core	BackupObjectServer	host2.tivoli.edunetcool	RUNNING	16706	
	BackupCollectionObjectServer_1	host2.tivoli.edunetcool	RUNNING	16707	
	DisplayObjectServer_2	host2.tivoli.edunetcool	RUNNING	16708	
	BackupGateway	host2.tivoli.edunetcool	RUNNING	16709	
	BackupCollectionGateway_1	host2.tivoli.edunetcool	RUNNING	16710	
	DisplayGateway_2	host2.tivoli.edunetcool	RUNNING	16711	

The host2 image contains three ObjectServers, and three gateways. All components are running.

The new configuration uses a display ObjectServer. The wizard created this ObjectServer with a root user, and no password.

3. Modify the **root** user, and set the password to **object00**.

```
nco_sql -server MULTI_DIS_2 -user root -password ''  
1> alter user 'root' set password 'object00';  
2> go  
1> quit
```

4. Verify the **root** user password.

```
nco_sql -server MULTI_DIS_2 -user root -password 'object00'  
1> quit
```



Important: There are two display ObjectServers in the architecture. MULTI_DIS_1 runs on host1, and MULTI_DIS_2 runs on host2.

The Web GUI component does not use the interfaces file to locate ObjectServers. Instead, the access information is hardcoded in the Web GUI data source definition file. The Web GUI data source definition file is configured to access ObjectServers that listen on port 4100. When you created the multitier architecture, you used port 4100 for the new display ObjectServers. You can start Dashboard Application Services Hub, and the Web GUI component can connect to the display ObjectServers on port 4100, with the root user, and password object00.

5. Start Dashboard Application Services Hub.

```
cd /opt/IBM/JazzSM/profile/bin  
. ./startServer.sh server1
```

Wait for the server to start.

6. Start the Netcool/OMNIbus Administrator utility.

```
nco_config &
```

7. Click **Yes** to import the omnit.dat file.

8. Click **Finish**.

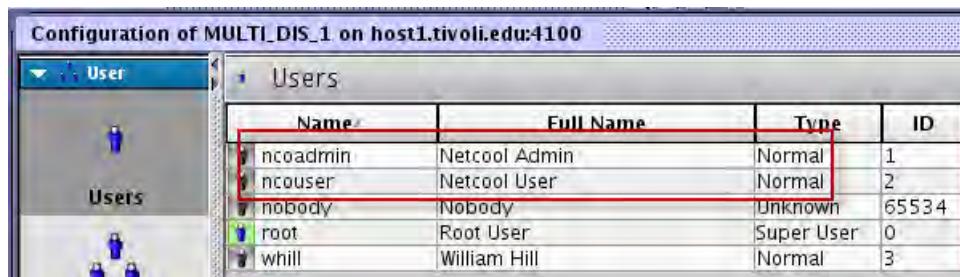
9. Examine the list of ObjectServers.

Name	Host	Port	SSL
AGG_V (backup)	host2.tivoli.edu	4110	false
AGG_V (primary)	host1.tivoli.edu	4110	false
MULTI_A_TO_D_GATE_1	host1.tivoli.edu	4302	false
MULTI_A_TO_D_GATE_2	host2.tivoli.edu	4302	false
MULTI_COL_B_1	host2.tivoli.edu	4120	false
MULTI_COL_P_1	host1.tivoli.edu	4120	false
MULTI_DIS_1	host1.tivoli.edu	4100	false
MULTI_DIS_2	host2.tivoli.edu	4100	false

Based on the names, it is easy to identify the aggregation, collection, and display ObjectServers.

10. Connect to **MULTI_DIS_1** as the **root** user with password **object00**.

11. Examine the list of users.



Name	Full Name	Type	ID
ncoadmin	Netcool Admin	Normal	1
ncouser	Netcool User	Normal	2
nobody	Nobody	Unknown	65534
root	Root User	Super User	0
whill	William Hill	Normal	3

The ncoadmin and ncouser entries are created in the ObjectServer by the Web GUI synchronization process. The synchronization process was configured in the Web GUI component in a previous unit.



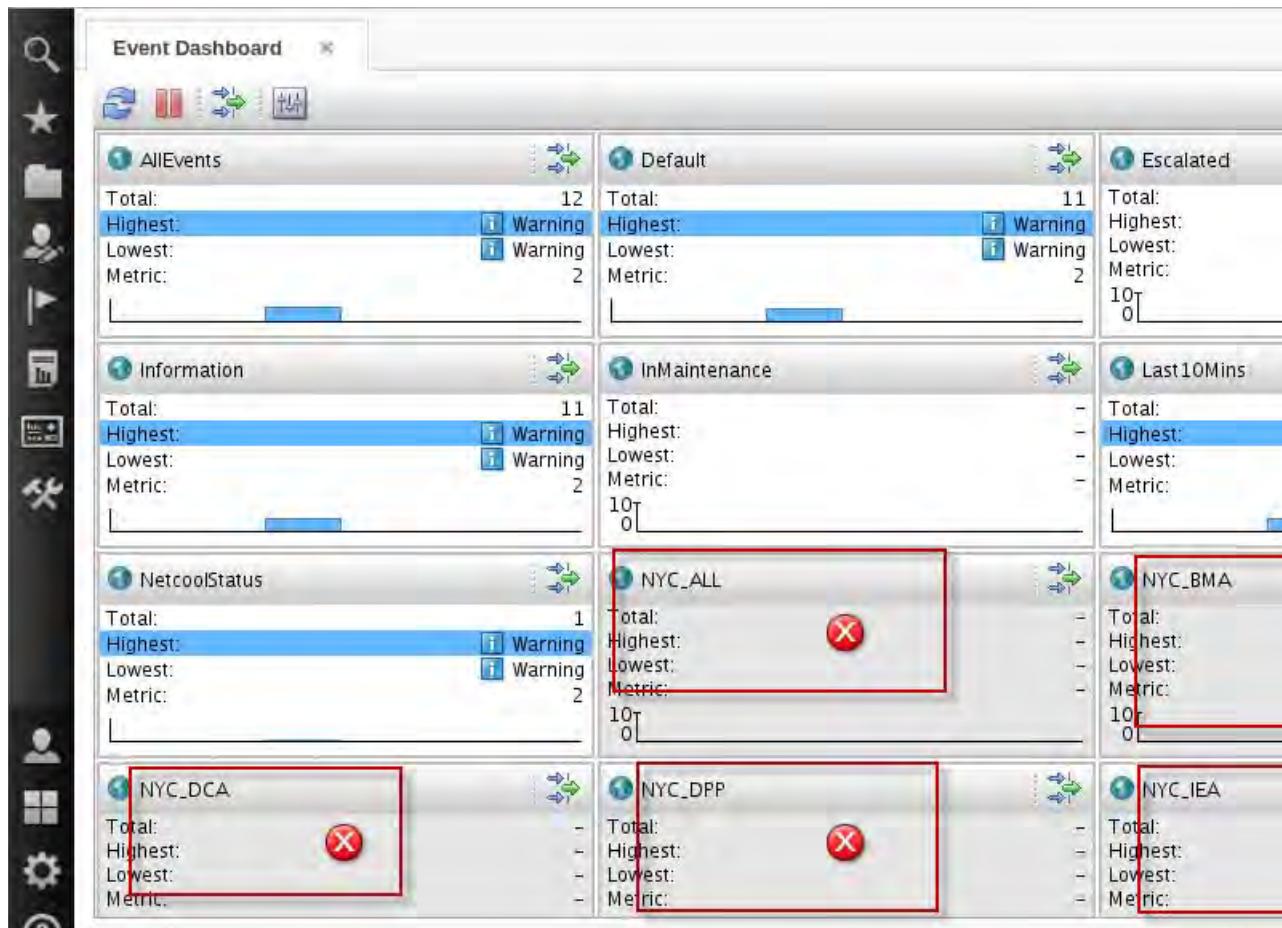
Important: If the list of users does not contain the LDAP users, then check MULTI_DIS_2. Web GUI synchronization only synchronizes a single ObjectServer. In the initial class images, the bidirectional gateway is configured to replicate users between the two aggregation ObjectServers. The display ObjectServers are not a high availability pair. Therefore, there is no bidirectional gateway between the two ObjectServers.

12. Close the Netcool/OMNIbus Administration utility.

13. Open a Firefox browser.

14. Log in to Dashboard Application Services Hub as the **ncoadmin** user with password **object00**.

15. Open the Event Dashboard.



Several monitor boxes contain a red X. Each of these boxes is based on a filter that uses an ObjectServer column name that does not exist in the display ObjectServers. Each of these boxes is unusable.

16. Click the box that is labeled **AllEvents**.

AllEvents@OMNIBUS - Active Event List (host2.tivoli.edu:16311)					
AllEvents			Default		
Sev	Ack	Node	Alert Group	Summary	
!	No	host1.tivoli.edu	MemstoreStatus	table_store soft limit: used 1 MB of capacity 4	
!	No	host1.tivoli.edu	TriggerStatus	Time for all triggers in profiling period (59.45	
!	No	host1.tivoli.edu	ConnectionStatus	GATEWAY: display_gate connected from host	
!	No	host1.tivoli.edu	nco_objserv	Average time to display events: 5 seconds.	
!	No	host1.tivoli.edu	ClientStatus	Time for all clients in granularity period (60s):	
!	No	host1.tivoli.edu	nco_objserv	ObjectServer MULTI_1 Profiler enabled a	
!	No	host1.tivoli.edu	nco_objserv	ObjectServer MULTI_COL_1 Profiler enabled	

The Active Event List opens, and you see events from the MULTI_DIS_1 ObjectServer.

17. Close the Active Event List window.

18. Log out of Dashboard Application Services Hub.

19. Close the Firefox browser.

The following is a summary of some important points from this exercise:

- You used the Initial Configuration Wizard to define a three layer architecture.
- You used the wizard to create the architecture components on the host1, and host2 images.
- You reused the Web GUI configuration settings, and successfully connected to the display layer ObjectServers.
- Web GUI synchronization created users in the display ObjectServers.
- You logged in to Dashboard Application Services Hub with a Web GUI-authorized user.
- You opened an Active Event List and saw events.

You created a three layer architecture that includes the recommended best practice configurations in a matter of a few minutes. You verified that you can send events to the collection layer, and view the events from the display layer.

TN035 1.0



ibm.com/training

Authorized
IBM | Training