Netcool/Impact
Version 6.1.0.2

Administration Guide

SC14-7551-00

Netcool/Impact
Version 6.1.0.2

Administration Guide

SC14-7551-00

Note
Before using this information and the product it supports, read the information in "Notices".

Edition notice

This edition applies to version 6.1.0.2 of IBM Tivoli Netcool/Impact and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Administration Guide

The Netcool/Impact *Administration Guide* contains instructions on installing, configuring, running, and monitoring Netcool/Impact.

Intended audience

This publication is for users who are responsible for installing, configuring, running and monitoring Netcool/Impact.

Publications

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

Netcool/Impact library

- *Quick Start Guide*, CF39PML

  Provides concise information about installing and running Netcool/Impact for the first time.
- *Administration Guide*, SC14755100

  Provides information about installing, running and monitoring the product.
- *User Interface Guide*, SC14755400

  Provides instructions for using the Graphical User Interface (GUI).
- *Policy Reference Guide*, SC14755300

  Contains complete description and reference information for the Impact Policy Language (IPL).
- *DSA Reference Guide*, SC14755500

  Provides information about data source adaptors (DSAs).
- *Operator View Guide*, SC14755600

  Provides information about creating operator views.
- *Solutions Guide*, SC14755200

  Provides end-to-end information about using features of Netcool/Impact.
- *Integrations Guide*, SC14755700

  Contains instructions for integrating Netcool/Impact with other IBM® software and other vendor software.
- *Troubleshooting Guide*, GC14755800

  Provides information about troubleshooting the installation, customization, starting, and maintaining Netcool/Impact.

Accessing terminology online

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

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

Accessing publications online

Publications are available from the following locations:

- The *Quick Start* DVD contains the publications that are in the product library. The format of the publications is PDF, HTML, or both. Refer to the readme file on the DVD for instructions on how to access the documentation.
- Tivoli Information Center web site at http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.netcoolimpact.doc6.1/welcome.html. IBM posts publications for all Tivoli products, as they become available and whenever they are updated to the Tivoli Information Center Web site.

  Note: If you print PDF documents on paper other than letter-sized paper, set the option in the **File → Print** window that allows Adobe Reader to print letter-sized pages on your local paper.
- Tivoli Documentation Central at http://www.ibm.com/developerworks/wikis/display/tivolidoccentral/Impact. You can also access publications of the previous and current versions of Netcool/Impact from Tivoli Documentation Central.
- The Netcool/Impact wiki contains additional short documents and additional information and is available at https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/Tivoli%20Netcool%20Impact.

Ordering publications

You can order many Tivoli publications online at http://www.elink.ibmlink.ibm.com/publications/servlet/pbi.wss.

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

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

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

1. Go to http://www.elink.ibmlink.ibm.com/publications/servlet/pbi.wss.
2. Select your country from the list and click **Go**.
3. Click **About this site** in the main panel to see an information page that includes the telephone number of your local representative.

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully. With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate all features of the graphical user interface.

For additional information, see Appendix A, “Accessibility,” on page 143.

Tivoli technical training

For Tivoli technical training information, refer to the following IBM Tivoli Education Web site at http://www.ibm.com/software/tivoli/education.

Support for problem solving

If you have a problem with your IBM software, you want to resolve it quickly. This section describes the following options for obtaining support for IBM software products:

- "Obtaining fixes"
- "Receiving weekly support updates"
- "Contacting IBM Software Support" on page x

Obtaining fixes

A product fix might be available to resolve your problem. To determine which fixes are available for your Tivoli software product, follow these steps:

1. Go to the IBM Software Support Web site at http://www.ibm.com/software/support.
2. Navigate to the **Downloads** page.
3. Follow the instructions to locate the fix you want to download.
4. If there is no **Download** heading for your product, supply a search term, error code, or APAR number in the search field.

For more information about the types of fixes that are available, see the *IBM Software Support Handbook* at http://www14.software.ibm.com/webapp/set2/sas/f/handbook/home.html.

Receiving weekly support updates

To receive weekly e-mail notifications about fixes and other software support news, follow these steps:

1. Go to the IBM Software Support Web site at http://www.ibm.com/software/support.
2. Click the **My IBM** in the toobar. Click **My technical support**.
3. If you have already registered for **My technical support**, sign in and skip to the next step. If you have not registered, click **register now**. Complete the registration form using your e-mail address as your IBM ID and click **Submit**.
4. The **Edit profile** tab is displayed.
5. In the first list under **Products**, select **Software**. In the second list, select a product category (for example, **Systems and Asset Management**). In the third list, select a product sub-category (for example, **Application Performance & Availability** or **Systems Performance**). A list of applicable products is displayed.
6. Select the products for which you want to receive updates.
7. Click **Add products**.
8. After selecting all products that are of interest to you, click **Subscribe to email** on the **Edit profile** tab.
9. In the **Documents** list, select **Software**.
10. Select **Please send these documents by weekly email**.
11. Update your e-mail address as needed.
12. Select the types of documents you want to receive.
13. Click **Update**.

If you experience problems with the **My technical support** feature, you can obtain help in one of the following ways:

Online

Send an e-mail message to erchelp@u.ibm.com, describing your problem.

By phone

Call 1-800-IBM-4You (1-800-426-4409).

World Wide Registration Help desk

For word wide support information check the details in the following link: https://www.ibm.com/account/profile/us?page=reghelpdesk

Contacting IBM Software Support

Before contacting IBM Software Support, your company must have an active IBM software maintenance contract, and you must be authorized to submit problems to IBM. The type of software maintenance contract that you need depends on the type of product you have:

- For IBM distributed software products (including, but not limited to, Tivoli, Lotus®, and Rational® products, and DB2® and WebSphere® products that run on Windows or UNIX operating systems), enroll in Passport Advantage® in one of the following ways:

  Online

  Go to the Passport Advantage Web site at http://www-306.ibm.com/software/howtobuy/passportadvantage/pao_customers.htm .

  By phone

  For the phone number to call in your country, go to the IBM Worldwide IBM Registration Helpdesk Web site at https://www.ibm.com/account/profile/us?page=reghelpdesk.
- For customers with Subscription and Support (S & S) contracts, go to the Software Service Request Web site at https://techsupport.services.ibm.com/ssr/login.
- For customers with IBMLink, CATIA, Linux, OS/390®, iSeries®, pSeries, zSeries, and other support agreements, go to the IBM Support Line Web site at http://www.ibm.com/services/us/index.wss/so/its/a1000030/dt006.
- For IBM eServer™ software products (including, but not limited to, DB2 and WebSphere products that run in zSeries, pSeries, and iSeries environments), you can purchase a software maintenance agreement by working directly with an IBM sales representative or an IBM Business Partner. For more information about support for eServer software products, go to the IBM Technical Support Advantage Web site at http://www.ibm.com/servers/eserver/techsupport.html.

If you are not sure what type of software maintenance contract you need, call 1-800-IBMSERV (1-800-426-7378) in the United States. From other countries, go to the contacts page of the *IBM Software Support Handbook* on the Web at http://www14.software.ibm.com/webapp/set2/sas/f/handbook/home.html and click the name of your geographic region for phone numbers of people who provide support for your location.

To contact IBM Software support, follow these steps:

1. “Determining the business impact” on page xi
2. “Describing problems and gathering information” on page xi
3. “Submitting problems” on page xi

Determining the business impact

When you report a problem to IBM, you are asked to supply a severity level. Use the following criteria to understand and assess the business impact of the problem that you are reporting:

Severity 1
: The problem has a *critical* business impact. You are unable to use the program, resulting in a critical impact on operations. This condition requires an immediate solution.

Severity 2
: The problem has a *significant* business impact. The program is usable, but it is severely limited.

Severity 3
: The problem has *some* business impact. The program is usable, but less significant features (not critical to operations) are unavailable.

Severity 4
: The problem has *minimal* business impact. The problem causes little impact on operations, or a reasonable circumvention to the problem was implemented.

Describing problems and gathering information

When describing a problem to IBM, be as specific as possible. Include all relevant background information so that IBM Software Support specialists can help you solve the problem efficiently. To save time, know the answers to these questions:

- Which software versions were you running when the problem occurred?
- Do you have logs, traces, and messages that are related to the problem symptoms? IBM Software Support is likely to ask for this information.
- Can you re-create the problem? If so, what steps were performed to re-create the problem?
- Did you make any changes to the system? For example, did you make changes to the hardware, operating system, networking software, and so on.
- Are you currently using a workaround for the problem? If so, be prepared to explain the workaround when you report the problem.

Submitting problems

You can submit your problem to IBM Software Support in one of two ways:

Online
: Click **Submit and track problems** on the IBM Software Support site at http://www.ibm.com/software/support/probsub.html. Type your information into the appropriate problem submission form.

By phone
: For the phone number to call in your country, go to the contacts page of the *IBM Software Support Handbook* at http://www14.software.ibm.com/webapp/set2/sas/f/handbook/home.html and click the name of your geographic region.

If the problem you submit is for a software defect or for missing or inaccurate documentation, IBM Software Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail. Whenever possible, IBM Software Support provides a workaround that you can implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the Software Support Web site daily, so that other users who experience the same problem can benefit from the same resolution.

Conventions used in this publication

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

Typeface conventions

This publication uses the following typeface conventions:

Bold

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

Italic

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

Monospace

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

Operating system-dependent variables and paths

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

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

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

Chapter 1. Introduction

A Netcool/Impact deployment is an installation of theNetcool/Impact components in your environment.

Overview of deployments

Netcool/Impact can be deployed as a single system installation, or a distributed installation.

To set up the deployment, you run the installer programs on systems in your environment. The installer copies the program files to the systems, and sets the minimal required configuration options. After a successful installation of all the components, you can then customize the configuration according to your needs. To run the deployment, you use a set of administration scripts, or services administration tools, depending on your operating system.

A centralized logging feature is provided that is used by both the Impact Server and the GUI Server, to monitor application events and status during application runtime. In addition, you can use application subcomponents, like for example services, to configure them to log their activity to files.

You can use the ObjectServer as an external authentication and authorization source for your deployment.

Deployment components

A Netcool/Impact deployment consists of the Impact Server, Name Server, and the GUI Server.

Impact Server
The Impact Server is responsible for managing the data model, running services, and policies that make up your Netcool/Impact implementation, and runs the policies in real time in response to events that occur in your environment.

Name Server
The Name Server is a subcomponent of the Impact Server that provides registration functionality for the GUI Server and the Impact Server.

GUI Server
The GUI Server is responsible for generating the dynamic Web-based user interface of Netcool/Impact.

The Impact Server overview

The Impact Server is responsible for managing the data model, running services, and policies that make up your Netcool/Impact implementation, and runs the policies in real time in response to events that occur in your environment.

The Impact Server runs as an application instance inside a Java™ application server, by default the embedded version of WebSphere Application Server. The application server is deployed automatically as part of the Netcool/Impact installation.

Important: It is possible to run multiple Impact Server instances per system for testing and validation purposes. In a production environment it is recommended to run one instance per system to achieve better resiliency and performance.

You can run Impact Servers in a clustered server configuration. Such a configuration consists of multiple instances of the Impact Server installed on separate systems but configured in such a way that they act as a single server. This type of configurations provides failover, failback, replication, and load balancing functions.

Although the Impact Server starts and stops automatically when the application server, where it resides, is started or stopped, you can also start and stop it independently of the application server. You change the configuration of the Impact Server by editing its properties file.

A subcomponent of the Impact Server, the Name Server, provides application registry functionality for the Netcool/Impact components. You can use its another subcomponent, the JRExec Server, to run external commands, scripts, and applications from within a policy.

The Impact Server uses a centralized logging feature, that incorporates Apache Log4j logging functionality.

The Name Server overview

The Name Server is a subcomponent of the Impact Server that provides registration functionality for the GUI Server and the Impact Server.

Applications that use the GUI Server register with the Name Server at startup. The GUI Server uses the information stored in the Name Server when brokering HTTP requests between users' Web browsers and the applications. The Name Server also plays a central role in server clustering.

The GUI Server overview

The GUI Server is responsible for generating the dynamic Web-based user interface of Netcool/Impact.

It brokers requests between end users' web browsers and Netcool/Impact, and returns the graphical views that you use to work with the data model, services, and policies.

The default GUI Server for Netcool/Impact 6.1 and later is the Tivoli Integrated Portal server.

An instance of the Tivoli Integrated Portal server is created during the installation, if you chose to install the GUI Server as one of the deployment components. The installer sets all of the default configuration properties for the server. After the installation, you can change the configuration of the GUI Server by editing its properties files.

The Tivoli Integrated Portal server starts and stops automatically when the application server, where it resides, is started or stopped, but you can also start, and stop it independently of the application server.

Deployment types

The supported deployment types are a single-system deployment, and a distributed deployment.

Single-system deployment

A single-system deployment consists of the Impact Server and the GUI Server installed on a single system in your environment. A single-system deployment is suitable for testing and demonstration purposes.

Distributed deployment

A distributed deployment consists of one or more instances of the Impact Server and the GUI Server installed across different systems in your environment. A distributed deployment is suitable for most real-world implementations of Netcool/Impact.

A typical distributed deployment can consist of two or more instances of the Impact Server installed on separate systems and configured as part of a server cluster, and an instance of the GUI Server installed on a system that is configured to allow users' Web browsers to access the GUI. Server clustering provides failover and load-balancing functionality for the Impact Server instances.

Setting up a deployment

Before you start setting up Netcool/Impact, you must have an understanding of how Netcool/OMNIbus and other Netcool products are installed and used in your environment.

Specifically, you must know what type of alerts are collected by Netcool probes and monitors, and how the alerts are stored in the Netcool/OMNIbus ObjectServer database. You must also have an understanding of your network topology, including the types of systems, devices, and applications that exist on the network and how they are monitored by Netcool/OMNIbus.

Planning an installation

After you understand how Netcool/Impact, and Netcool® OMNIbus are installed and used in your environment, you can plan your deployment.

For testing or demonstration purposes, a good practice is to install Netcool/Impact, and its components on a single system. This type of deployment requires little planning and is the easiest to create and maintain.

For real life production deployments, you must take into account your goals, requirements, and available resources before you install the software. One best practice is to create a diagram of the installation you want to create before you begin. IBM technical support can help you determine what type of hardware you need to run the deployment and how to configure it to fit your requirements.

Prerequisite scanner

Download and run the IBM Prerequisite Scanner as part of your installation planning.

IBM Prerequisite Scanner is a stand-alone prerequisite checking tool that analyzes system environments before the installation or upgrade of a Tivoli product or IBM solution. The scanner includes configuration files for Netcool/Impact.

Download the scanner

Download the scanner for your operating system.

About this task

This procedure describes where to find the scanner and the information needed to install the scanner.

Procedure

1. Download the scanner version for your operating system from this page:
   http://www-933.ibm.com/support/fixcentral/swg/selectFixes?parent=ibm~Tivoli&product=ibm/Tivoli/Prerequisite+Scanner&release=All&platform=All&function=all
2. For general instructions on downloading, installing, and running the scanner, click **More Information** for the version you are downloading.
3. Extract the scanner files to a location on your Netcool/Impact host. This location is called the *ips_root* directory in this topic.

What to do next

Run the scanner on your host system.

Running the scanner for Netcool/Impact

You need to use the Netcool/Impact specific options described here when you run the scanner.

Procedure

1. Open a command window:
2. Run the command:
   - Windows: `prereq_checker.bat "NCI 06100000" detail`
   - UNIX: `prereq_checker.sh "NCI 06100000" detail`

Results

Possible results are as follows:

FAIL: If the target server does not meet the prerequisites specified in the `cfg` files, the scanner returns `FAIL` for the Netcool/Impact check. The failed prerequisites are displayed in the screen output. To resolve the failure, take the appropriate actions, for example, install the missing operating system packages, or increase disk space.

PASS:
If the target server has all the prerequisite specified in the `cfg` files, the scanner returns PASS for the Netcool/Impact check. If the scanner returns PASS for the Netcool/Impactcheck you can install, configure and start the product on the target server.

- For more information about the prerequisite scanner, see the IBM Prerequisite Scanner Information Center, available from the following URL:

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp?topic=%2Fcom.ibm.ips.doc_1.1.1.7%2Fips_ic-homepage.html

System requirements

Before you install IBM Tivoli Netcool/Impact version 6.1.0.2, read the system requirements.

Software Product Compatibility Reports

The most up-to-date information about supported hardware, software, browsers, and operating systems is provided by IBM Software Product Compatibility Reports.

For more information about running the Software Product Compatibility Reports, see the Overview and Planning section of the Tivoli Netcool/Impact wiki at: https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli Netcool Impact/page/Prerequisites for Tivoli Netcool Impact 6.1.

Assigning ports

Methods to assign ports in a Netcool/Impact installation.

The Netcool/Impact installer checks for port availability in the following ways:

- GUI and console mode:
  - Simple installation: The Netcool/Impact installer has an internal default start port address (9080) for the Impact Server and (16310) for the GUI Server and generates the other port addresses. In a simple installation, if the default start ports are not available, the Netcool/Impact installer checks incrementally for the next available port and uses this port as the new starting port. The Netcool/Impact installer generates the remaining ports from the new starting port.
  - Advanced installation: You can assign the starting address for the ports for the Impact Server and for the GUI Server. The Netcool/Impact installer uses these ports to generate the ports to be used. If any of the provided ports are already in use the Netcool/Impact installer checks incrementally for the next available port and uses this port as the new starting port. The Netcool/Impact installer generates the remaining ports from the new starting port.
- Silent installation: The Netcool/Impact installer requires a set of sequentially numbered ports. The sequence of ports is supplied during installation in the response file. The Netcool/Impact installer checks that the number of required ports (starting with the initial port value) are available before assigning them. If one of the ports in the sequence is already in use, the Netcool/Impact installer automatically terminates the installation process. You must specify a different range of ports in the response file.

Ports used by Netcool/Impact and Tivoli Integrated Portal

Use the following tables to identify the ports required to install Netcool/Impact and the Tivoli Integrated Portal.

Default ports to be defined for the Impact Server (Impact Profile) are shown in the Impact Server silent response file `installSilent_response.txt`. Default ports for the Impact Server apply to all operating systems except where otherwise stated.

Table 1. Default ports to be defined for Netcool/Impact

Parameter	Port Number
IAGLOBAL_EWAS_HTTP_PORT **Note:** The Netcool/Impact installer uses port 9080 for nameserver configuration by default for all operating systems except AIX. When the wsmserver is running on AIX it uses port 9090. The Netcool/Impact installer does not use the default port 9080 for nameserver configuration but the next available port which is 9110. In the nameserver panel, choose `localhost:9110` as the nameserver port. Before you install Netcool/Impact, you can check if the wsmserver is using port 9090. Execute `nestat -na | grep wsmserver`. If port 9090 is not being used, the Netcool/Impact installer can use port 9080 as the default port.	9080
IAGLOBAL_EWAS_WC_defaulthost_secure	9081
IAGLOBAL_EWAS_BOOTSTRAP_ADDRESS	9082
IAGLOBAL_EWAS_SOAP_CONNECTOR_ADDRESS	9083
IAGLOBAL_EWAS_IPC_CONNECTOR_ADDRESS	9084
IAGLOBAL_EWAS_ADMIN_PORT	9085
IAGLOBAL_EWAS_WC_adminhost_secure. Use this port to import SSL certificates to be used in the Impact Server.	9086
IAGLOBAL_EWAS_DCS_UNICAST_ADDRESS	9088
IAGLOBAL_EWAS_ORB_LISTENER_ADDRESS	9090
IAGLOBAL_EWAS_SAS_SSL_SERVERAUTH_LISTENER_ADDRESS	9091
IAGLOBAL_EWAS_CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS	9092
IAGLOBAL_EWAS_CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS	9093
IAGLOBAL_EWAS_REST_NOTIFICATION_PORT	9094

Netcool/Impact also uses RMI ports, see "Configuring RMI ports" on page 39 for a list of ports.

For information about viewing the assigned ports, see "Viewing the Impact Server profile" on page 7.

The following default ports are defined for the GUI Server (Tivoli Integrated Portal Profile) and are shown in the GUI Server silent response file:

Table 2. Default ports to be defined for the GUI Server

Parameter	Port Number
IAGLOBAL_WC_defaulthost	16310
IAGLOBAL_WC_defaulthost_secure. Make sure this port is open because the Netcool/Impact UI is accessed through `http://hostname:16311/ibm/console`.	16311
IAGLOBAL_BOOTSTRAP_ADDRESS	16312
IAGLOBAL_SOAP_CONNECTOR_ADDRESS	16313
IAGLOBAL_IPC_CONNECTOR_ADDRESS	16314
IAGLOBAL_WC_adminhost	16315

Table 2. Default ports to be defined for the GUI Server (continued)

Parameter	Port Number
IAGLOBAL_WC_adminhost_secure. This port is used when you launch the administration console from the TIP profile.	16316
IAGLOBAL_DCS_UNICAST_ADDRESS	16318
IAGLOBAL_ORB_LISTENER_ADDRESS	16320
IAGLOBAL_SAS_SSL_SERVERAUTH_LISTENER_ADDRESS	16321
IAGLOBAL_CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS	16322
IAGLOBAL_CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS	16323
IAGLOBAL_REST_NOTIFICATION_PORT	16324

For information about viewing the assigned ports, see "Viewing the application server profile" on page 8.

Viewing the Impact Server profile

The Impact Server profile lists the ports assigned and provides additional information about the profile.

About this task

Open the Impact Server profile to review the port number assignments and other information. The profile is available as a text file on the computer where it is installed. To view the ports assigned to Netcool/Impact:

Procedure

1. Go to the `tip_home_dir\profiles\ImpactProfile\logs` directory.
2. Open `AboutThisProfile.txt` in a text editor.

Example

The profile for an installation on in a Windows environment as it appears in the Impact Server profile `tip_home_dir\profiles\ImpactProfile\logs\AboutThisProfile.txt`:

```
Application server environment to create: Application server
Location: C:\Program Files\IBM\tivoli\<localpath>
Disk space required: 200 MB
Profile name: ImpactProfile
Make this profile the default: True
Node name: ImpactNode
Host name: <localhost>
Enable administrative security (recommended): True
Administrative console port: 9085
Administrative console secure port: 9086
HTTP transport port: 9080
HTTPS transport port: 9081
Bootstrap port: 9082
SOAP connector port: 9083
Run application server as a service: False
Create a Web server definition: False
Performance tuning setting: Standard
```

What to do next

To see the complete list of defined ports on the Impact Server:

1. Go to tip_home_dir\profiles\ImpactProfile\properties\portdef.props.
2. Open portdef.props in a text editor:

```
#Create the required WAS port properties for Impact
WC_defaulthost=9080
WC_adminhost=9085
WC_defaulthost_secure=9081
WC_adminhost_secure=9086
BOOTSTRAP_ADDRESS=9082
SOAP_CONNECTOR_ADDRESS=9083
IPC_CONNECTOR_ADDRESS=9084
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS=9091
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS=9093
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS=9092
ORB_LISTENER_ADDRESS=9090
DCS_UNICAST_ADDRESS=9088
REST_NOTIFICATION_PORT=9094
```

Viewing the application server profile

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

About this task

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

Procedure

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

Example

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

```
Application server environment to create: Application server
Location: C:\IBM\tivoli\tipv2\profiles\TIPProfile
Disk space required: 200 MB
Profile name: TIPProfile
Make this profile the default: True
Node name: TIPNode Host name: tivoliadmin.usca.ibm.com
Enable administrative security (recommended): True
Administrative consoleport: 16315
Administrative console secure port: 16316
HTTP transport port: 16310
HTTPS transport port: 16311
Bootstrap port: 16312
SOAP connector port: 16313
Run application server as a service: False
Create a Web server definition: False
```

What to do next

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

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

Installing components

You use the Netcool/Impact installer to install the Impact Server and the GUI Server.

The installer is a program that can be run in GUI mode, console mode, and silent mode. In GUI mode a series of windows guide you through the installation process. In console mode, the installer prompts you to enter the required information from the command line. If you are running the installers remotely using telnet or another command line application, you must run the installer in console mode.

Configuring components

The installer program sets the minimum required configuration properties during installation.

You can change the configuration of a component at any time by manually editing the properties files. Depending on the component, you might need to stop and restart after making configuration changes.

Chapter 2. Installation and migration

The Netcool/Impact installer is an application that copies the program files to the target system and sets the minimum required configuration properties.

The installer deploys the Impact Server, and the GUI Server files into the embedded version of WebSphere Application Server on the target system.

You can choose between the GUI, console, and silent mode of the installation. If you are installing Netcool/Impact remotely using telnet or another command-line application, you must run the installer in console mode. The silent mode, where you need to edit an installation properties file before the installation, is recommended for advanced users.

Using the launchpad is a convenient way to run the Netcool/Impact installer in GUI mode, especially if you are planning the integration with Tivoli Business Service Manager. For more information, see "Running the installer in the launchpad" on page 12

The installer will not migrate earlier versions of Netcool/Impact to version 6.1. You use a dedicated migration script for that. For more information, see "Migrating to version 6.1" on page 30.

Before you start the installation

Before you start the installation, make sure you meet the minimum system requirements, and that there is sufficient space on the temp volume.

The self extractor checks for free disk space 3 times the size of the installer on the volume where the temp directory is located. If there is not enough free space, the installer prompts you for an alternative location for the extraction. The value of the temp directory path is resolved based on the system environment variable, `IATEMPDIR` on UNIX platforms and `TEMP` on Windows platforms. Set this environment variable before running the installer to redirect the temporary directories of the installer to the location where there is sufficient disk space.

On UNIX systems, you can run the installer as any user that has read, write, and execute permissions to the target directory.

Remember: Once you installed a Tivoli Integrated Portal product using a certain user account, you must use the same user account to install, uninstall, or modify every subsequent Tivoli Integrated Portal product on that system.

If you are configuring the Tivoli Integrated Portal profile for a cluster of servers, you must use the same user name and password for all the servers in the cluster.

On Windows systems the userid used for the installation must have administrator permissions to be able to create a Windows Service entry.

Restriction: There is a known limitation on Windows, that you cannot successfully install Netcool/Impact under an administrator account that has national characters in its name. For the purpose of the installation use an administrator account with a name consisting only of ASCII characters, for example "impact_administrator".

Running the installer in the launchpad

Use the launchpad to install Netcool/Impact, and other software components required for the integration with Tivoli Business Service Manager (TBSM).

Procedure

1. Start the launchpad.

   Use one of the following methods to start the launchpad:

   - UNIX systems.

     From the directory containing `launchpad.sh`, issue the command:

     ```
     ./launchpad.sh
     ```

     Important: You need the X Window System to run the launchpad on UNIX systems.

   - Windows systems.

     Using the Windows Explorer, navigate to the directory containing the installation image, and double click the `launchpad.exe` file.

2. Select **Install IBM Tivoli Netcool/Impact 6.1** > **Run the Netcool/Impact 6.1 Installation Program** from the launchpad navigator.

   This will start the installation in GUI mode. Complete the installation, using the procedure outlined in "Running the installer in GUI and console mode."

3. Optional: Install the Discovery Library support.

   If you are planning to integrate Netcool/Impact with TBSM, the Discovery Library support provides a means of importing resource information into TBSM.

   a. Install the required database schema by clicking **Install DB2 Schema** > **Run the DB2 schema installation program** from the launchpad navigator.

      Attention: In a z/Linux environment, you cannot install the DB2 schema from the launchpad. Instead, you must use this command from the command line:

      ```
      launchpad/zlinux/setup-dbconfig-zlinux.bin
      ```

   b. Install the Discovery Library support, by selecting **Install Discovery Library support** > **Run the Discovery Library installation program** from the launchpad navigator.

Running the installer in GUI and console mode

This procedure provides a quick overview of the installation steps in the GUI and console mode.

Procedure

1. Start the installation.

   You can start the installation from any place in the file system by providing the full path to the installation file. For more information, see "Starting the installation" on page 14.

2. Select the language that will be used during the installation and continue with the installation.

   You can only choose from those languages that the system can support, not the full translated set.

3. Read the introduction and proceed with the installation.
4. Read and accept the license agreement.

5. If a Deployment Engine (DE) is installed on the machine, and it is older than the one on the installation image, the installer will prompt you for a directory to backup the old DE before upgrading it to the newer version.
6. Select the installation directory.

   Accept the default installation directory or type an absolute path of a directory you want to install to. For more information, see "Selecting the installation directory" on page 14.
7. Choose the type of installation.

   Select the simple or advanced installation. For more information, see "Selecting the type of installation" on page 15.
8. Configure the Tivoli Integrated Portal profile.

   For more information, see "Configuring the Tivoli Integrated Portal profile" on page 15.
9. (Advanced) Select the Netcool/Impact components that you want to install.

   For more information, see "Selecting installation components" on page 16.
10. (Advanced) Configure the user registry.

    Select the type of user registry that will be used for user management and authentication. You can choose between a local file based, or ObjectServer based user registry. For more information, see "Configuring the user registry" on page 16. You will have to go through this configuration step only if you select the advanced installation option.
11. (Advanced) Configure the event source.

    Netcool/Impact will need an event source and initially it will try to connect to the Netcool/OMNIbus ObjectServer. For more information, see "Configuring the event source" on page 16.
12. (Advanced) Configure the embedded version of WebSphere Application Server ports for the TIP profile.

    These ports will be used by the GUI Server to communicate with TIP. For more information, see "Configuring WebSphere profile ports" on page 17.
13. (Advanced) Configure embedded version of WebSphere Application Server ports for the Impact profile.

    The Impact profile ports will be used for communication with the Impact Server. For more information, see "Configuring Impact Server WebSphere profile ports" on page 17.
14. (Advanced) Configure the Name Server.

    This configuration step is only required if you chose to install the GUI Server as one of the deployment components. For more information, see "Configuring the Name Server" on page 18.
15. (Advanced) Configure the Impact Server instance.

    You will need to perform this configuration step only if you are installing the Impact Server as one of the installation components. For more information, see "Configuring the Impact Server instance" on page 19.
16. (Advanced) Select and configure a version control system.

    The version control manager uses Impact Subversion (SVN) as the default version control which is installed automatically with the Impact Server if you leave the default selection. For more information, see "Selecting and configuring a version control" on page 19.
17. If you are installing into an existing directory (reuse scenario), the installation program will prompt you for a directory to backup the existing directory into.
18. Review the pre-installation summary.

Make sure that the pre-installation summary reflect your choices and proceed with the installation.

19. Finish the installation.

    Make sure the installation completed without errors. It may be a good idea to review the installation logs.

Starting the installation

You can start the installation from any place in the file system by providing the full path to the installation file.

Procedure

- To start the installation in GUI mode:
  - UNIX systems

    Run `launchpad.sh` to start Launchpad, and select **Install IBM Tivoli Netcool/Impact 6.1.** > **Run the Netcool/Impact 6.1 Installation Program**.
  - Windows systems

    Double click `launchpad.exe` to start Launchpad, and select **Install IBM Tivoli Netcool/Impact 6.1.** > **Run the Netcool/Impact 6.1 Installation Program**.
- To start the installation in console mode:
  - UNIX systems

    Use the following command syntax to start the installation:

    ```
    ./setup<system>.bin -i console
    ```

    or

    ```
    /path/setupsystem.bin -i console
    ```

    path is a fully qualified path to the installation file, and *system* is the name of the operating system. An example of the command on Linux:

    ```
    ./setuplinux.bin -i console
    ```
  - Windows systems

    Use the following command syntax to start the installation:

    ```
    path\setupwin.exe -i console
    ```

    For example:

    ```
    E:\6.1.0.0-TIV-NCI-win\Impact\setupwin.exe -i console
    ```

  The installer will only ask you to provide values for some installation parameters, and it will use default values for others, like for example the JVM memory allocated to the application server. Still, you can modify any other default installation parameter even at the installation stage. To do that, use the `-D` attribute followed by the installation parameter and its new value on the command line. For example:

  ```
  setupwin.exe -i console -D$IAGLOBAL_EWAS_MIN_HEAP_SIZE$=512
  ```

  This command will set the minimum JVM heap size for the application server to 512 MB (half of the default size, 1200, for 1.2 GB). For more information about what installation parameters you can modify, see "Silent installation response file" on page 20.

Selecting the installation directory

Accept the default installation directory, `C:\Program Files\IBM\tivoli` on Windows systems, and `/opt/IBM/tivoli` on UNIX systems, or type an absolute path of a directory you want to install to.

If you are running a new installation you cannot install into a directory that has an instance of Netcool/Impact. On UNIX systems you can install into any directory to which you have write privileges (+755 rights on UNIX platforms).

In case of the default location on UNIX, however, note that even if you have +755 (drwxr-xr-x) permissions to `/opt` the installer will not create `IBM/tivoli` in it. To be able to use the default directory go to the `/opt` directory, create the `IBM` directory and give it permissions of 777 (drwxrwxrwx). After that the installer will be able to create the `/opt/IBM/tivoli` directory. An alternative method, is to add the non-root user to run the installer to an appropriate group that allows them to create the default directory.

Restriction: On UNIX systems do not use special characters or spaces in the install path name. Do not specify an installation directory path that includes parenthesis, such as `c:\Program Files (x86)`. The install may succeed with this path, but other utilities and components will fail when you attempt to run the application using a path with parenthesis.

Selecting the type of installation

Choose between the simple or advanced type of installation.

The simple installation is not recommended for production environments. If you select the simple installation, after the TIP Information the installer takes you directly to the pre-installation summary and runs the installation with a default set of values.

Important: If you select the simple installation, you will not be able to configure the primary event source for Netcool/Impact. Still, the installation program will create a default ObjectServer data source file that you have to configure after the installation. For more information about configuring the default ObjectServer data source, see “Configuring the default ObjectServer data source” on page 27.

The advanced installation is recommended for production environments. It gives you an opportunity to choose which deployment components to install, and also to change other default values. If you select the advanced mode, you need to be prepared to go through a series of advanced configuration steps, and to provide the installer with some answers.

Configuring the Tivoli Integrated Portal profile

Provide a user id and a password for your Tivoli Integrated Portal profile.

You use these credentials to log on to Netcool/Impact in Tivoli Integrated Portal, and to manage Netcool/Impact. The default user name is “tipadmin”, but you can change it.

Restriction: The installation will fail, if you use the Object Server based user registry, with the “tipadmin” user already existing on the Object Server. To prevent that, you can either remove the “tipadmin” user from the Object Server before running the installation, or you can use a different user name for your TIP profile.
Attention: If you are configuring the Tivoli Integrated Portal profile for a cluster of servers, you must use the same user name and password for all the servers in the cluster.

Selecting installation components

Select the components of Netcool/Impact that you would like to install.

If you accept the default selection, both the GUI Server and the Impact Server will be installed on the same computer. In production environments the Impact Server and GUI Server are usually installed on separate computers. So, for example, if you already installed the Impact Server on another machine, you can choose to install the GUI Server alone.

Configuring the user registry

Select the type of user management and authentication.

During the installation, you can choose either the ObjectServer based user registry, or the file-based user registry (default).

ObjectServer

Select this option to use the Netcool/OMNIbus ObjectServer as you user registry.

If you are familiar with the ObjectServer as a user registry, you may want to use this option.

You can use this option with failover or single sign-on installations.

Local File Based

Useful for proof-of-concept installation. Cannot be used with failover, load balancing, or single sign-on installations.

Important: If you selected the ObjectServer based user registry, the installer will run successfully even if the ObjectServer information you provided is incorrect. However, the ObjectServer users may not get created in Netcool/Impact. To fix this run the `$TIP_HOME/bin/confvmm4ncos` script after the installation, and restart the server.

Configuring the event source

Netcool/Impact will need an event source and initially it will try to connect to the Netcool/OMNIbus ObjectServer.

Other types of event sources require manual configuration.

Important: To use the ObjectServer as an event source for Netcool/Impact you need to install it separately. The Netcool/Impact installer will not install it.

You can keep the default values or use a different fully qualified host name of the host that will be used as the ObjectServer host.

You can choose not to use a password as it is valid for an ObjectServer not to have a password.

ObjectServer Name

The name of the event source server. Default is NCOMS.

ObjectServer Host

By default the installer uses the current host name as the ObjectServer host and its default port.

ObjectServer Port
The ObjectServer port. Default is 4100.

ObjectServer User
The user name, that Netcool/Impact uses to authenticate to the ObjectServer. Default is root.

ObjectServer Password
Password used to authenticate. No password by default

Important: The default ObjectServer user name, and password are typical values that the Netcool/OMNIbus ObjectServer is installed with. For better security, consider changing these values after the ObjectServer installation.

Configuring WebSphere profile ports

Configure the embedded version of WebSphere Application Server ports for the Tivoli Integrated Portal (TIP) profile.

These ports will be used by the GUI Server to communicate with TIP.

Starting port number provided

Use the starting port number provided if you want the installer to reserve a set of ports for the TIP profile automatically. After you select this option you can use the starting port number value, that the installer suggests, or change it according to your preferences in the screen that follows.

Modified port value file

If you select the modified port value file, you have to have a file with your preferred port values located on your disk, and point the installer to this location in the screen that follows. Because the installer cannot verify if these ports are available, you need to ensure all your preferred ports are available.

Attention: Your configuration is saved in the `<installDir>/tipv2/profiles/TIPProfile/properties/portdef.props` file.

Configuring Impact Server WebSphere profile ports

Configure the embedded version of WebSphere Application Server (eWAS) ports for the Impact profile.

The Impact profile ports are used for communication with the Impact Server. By default, eWAS uses port 9080 for the HTTP port (GUI Server) and 9085 for the administrator port (administration console). If these ports are available the installer reserves them. If not it reserves the next vacant port.

Restriction: On AIX system, wsmserver uses port 9090. You can check that by running the command `nestat -na | grep wsmserver`. In this situation, the installer cannot reserve the default set of ports starting from 9080 so a next available port will be used as the starting port. If this is the case, use the next available port, 9110, for your Name Server HTTP port.

Starting port number provided

Use the starting port number provided if you want the installer to reserve a set of ports for the Impact profile automatically. After you select this

option you can use the starting port number value, that the installer suggests, or change it according to your preferences in the screen that follows.

Modified port value file

If you select the modified port value file, you need a file with your preferred port values located on your disk, and point the installer to this location in the screen that follows. Because the installer cannot verify if these ports are available, you need to ensure all your preferred ports are available.

Remember: Your configuration is saved in the `$TIP_HOME/profiles/ImpactProfile/properties/portdef.props` file.

Configuring the Name Server

Configure the Name Server to provide registration functionality for the deployment components.

You can define multiple Name Server-port pairs and then associate your current installation to multiple name servers. Each Name Server port that you provide will be tested to check if it is active. If the port is active you will proceed to the next step of the installation. If the port is not active you will see a message indicating that the port is not active but you will still be able to continue with the installation. For the installation to be successful, however, make sure that the server-port pair as you have defined is available.

Important: If you are defining an Impact Server, the current host name and port number must be in the list. You will get an error if it is not in the list.

The Name Server is also used to store information used for server clustering. In a clustered configuration all cluster members should be added to the list, the first server in the list acting as the primary. For each member that you are installing the list must be exactly the same.

If you have a file that contains a list of your defined name servers, you can import that list. The file must be well-formed, so the best option is to use the `nameserver.props` file that is created in the `$IMPACT_HOME/etc` directory during the first installation.

Attention: Do not rename the `nameserver.props` file. Provide a fully qualified path to the file.

You can update the Name Server list after the installation by running `nci_configuration_utility`. For more information, see “Post installation utility” on page 30.

When the Impact Server is started during the installation, the following message is displayed in the `netcool.log` file:

```
Unable to login to configured name servers host [<hostname>], port [<port>]
```

The message occurs because the Name Server can take time to initialize while Netcool/Impact starts. You can ignore this message. It is normal and it disappears when the Name Server server is initialized.

Configuring the Impact Server instance

Configure the instance of Impact Server that you are installing.

You will need to perform this configuration step only if you are installing the Impact Server as one of the installation components.

You may need to modify the default instance name, and cluster group if you are installing a member of a cluster. If you are installing a stand-alone Impact Server you can go with the default values.

Instance Name

The name of the Impact Server instance. You can use any unique identifying string.

Cluster Group

If you are installing a member of a cluster provide the name of the instance and the cluster group to which you want it to belong.

Command Line Port

The command-line manager service tool allows you to access the Impact Server from the command-line interface to start and stop services as well as configure their parameters. The default value of the command-line service port is 2000.

DB Port

The DB port is the port that you can use to access the Netcool/Impact database and by default it is 5435.

Selecting and configuring a version control

The version control manager uses Impact Subversion (SVN) as the default version control which is installed automatically with the Impact Server if you leave the default selection.

No additional configuration steps will be required if you decide to go with the default setting and immediately after the installation you will be able to save policies, data sources, data types, and configuration properties as revisions in a source control archive.

Choose another option if you want to use a different version control system. Be prepared to provide additional configuration information in the steps to follow:

Version Control Path

Provide the path to your version control. You have to provide this information if the Impact Server or both the Impact Server and the GUI Server are selected together from the components to be installed and Subversion, or CVS, or RCS or ClearCase® is selected in the version Choose Version Control System step.

Version Control Repository

Set the repository for your version control. You have to provide this information if the Impact Server or both the Impact Server and the GUI Server are selected together from the components to be installed and Subversion or CVS is selected in the Version Control System step.

Silent mode

You use the silent installation mode in an environment where neither the GUI mode, or console mode is available.

Before running the installer in silent mode you need to edit the installation properties file. A template of the properties file, `installSilent_response.txt`, is provided with the installation file and you can edit it with your favorite text editor. The following rules apply to the response file:

- The path to the installer properties file can be either absolute, or relative to the directory in which the installer resides.
- Use the response file, `installSilent_response.txt`, shipped with the installer. You can change the name of the file.
- If an installer properties file is specified but does not exist, the default properties file, if present, is used. Otherwise, any supplied command-line options are used, or if no additional options are specified, the installer is run using the default settings.

Running the installer in silent mode

Use this procedure to install Netcool/Impact in silent mode.

Procedure

1. Copy the installation properties file to a directory on your file system alongside the installation files.
2. The command you use to start the installation in silent mode has the following syntax:
   - UNIX systems: `./setup<system>.bin -i silent -f <path>` or `/<path>/setup<system>.bin -i silent -f <pathToInstallerPropertiesFile>`

     where *path* is a fully qualified path to the installer properties file, and *system* is the name of the operating system. An example of the command on Linux:

     ```
     ./setuplinux.bin -i silent -f installer.properties
     ```
   - Windows systems: `setupwin.exe -i silent -f <path>/<installer.properties>`

     For example:

     ```
     setupwin.exe -i silent -f E:\6.1.0.0-TIV-NCI-win\Impact\installer.properties
     ```
3. (Optional) Follow the installation progress by viewing the installation logs.

   The installation log files, `IMPACT6.1_install-xx.log` (where *xx* can be 00 or 01 or 02) are created in your home directory. The installation program also creates a lock file, `IMPACT6.1_install-xx.log.lck`, that is removed when the installation is complete.

Silent installation response file

You can use the silent installation response file template, `installSilent_response.txt` that is shipped with the installation image, to create your silent installation response file.

The following table explains the silent installation response file parameters:

Table 3. Silent installation parameters

Parameter	Description
LICENSE_ACCEPTED	Uncomment this parameter and change its value to true to accept the license agreement.
IAGLOBAL_EWAS_MIN_HEAP_SIZE IAGLOBAL_EWAS_MAX_HEAP_SIZE	The minimum and maximum JVM heap size value for the Impact Server. The default value is 256 and 1200 (1.2GB), respectively. To change the defaults uncomment the parameter and provide your values.
IAGLOBAL_TIP_EWAS_MIN_HEAP_SIZE IAGLOBAL_TIP_EWAS_MAX_HEAP_SIZE	The minimum and maximum JVM heap size value for the GUI Server (TIP server). The default value is 256 and 512, respectively. To change the defaults uncomment the parameter and provide your values.
NO_BACKUP	Set this parameter to false to backup your existing installation.
BACKUP_DIR	A directory on the system to backup the existing application to. Used with NO_BACKUP=false. The directory must already exist.
USER_INSTALL_DIR	The installation directory. Default is /opt/IBM/tivoli on UNIX systems and C:\Program Files\IBM\tivoli on Windows systems.
IAGLOBAL_INSTALL_LOCATION_SELECTION	Set this parameter to reuse if you are reusing an existing TIP installation. In this case you must also set USER_INSTALL_DIR to the location where the TIP instance is installed.
IAGLOBAL_WASUserID	The user id for accessing Netcool/Impact in TIP. The default user name is “tipadmin”. **Restriction:** The installation will fail, if you use the Object Server based user registry, with the “tipadmin” user already existing on the Object Server. To prevent that, you can either remove the “tipadmin” user from the Object Server before running the installation, or you can use a different user name for your TIP profile.
IALOCAL_WASPassword	The password to use with IAGLOBAL_WASUserID.
CHOSEN_INSTALL_SET_S	To change the default values of the installation set this parameter to advanced, to use the default values set it to default.
IAGLOBAL_GUI_SERVER_RESULT	Set to 1 to install the GUI Server.
IAGLOBAL_IMPACT_SERVER_RESULT	Set to 1 to install the Impact Server.
IAGLOBAL_WC_defaulthost ... IAGLOBAL_REST_NOTIFICATION_PORT	A set of ports that are used by the GUI Server. Used with IAGLOBAL_GUI_SERVER_RESULT=1.
IAGLOBAL_IMPORT_NAMESERVER_FILE	Set this parameter to true to import a list of Nameservers from an external file. Set to false to manually provide a list of Nameservers. Used with IAGLOBAL_IMPACT_SERVER_RESULT=1.

Table 3. Silent installation parameters (continued)

Parameter	Description
IAGLOBAL_NAMESERVER_PROPS_FILE	Provide a qualified path to a file that contains a list of your Nameservers. The file must be valid, so the best option is to use the nameserver.props file that is created in the $IMPACT_HOME/etc directory during the first installation. Used with IAGLOBAL_IMPORT_NAMESERVER_FILE=true and IAGLOBAL_IMPACT_SERVER_RESULT=1.
IAGLOBAL_NAMESERVER_HP	Provide a list of your Nameservers host name (or IP address), port number pairs. A host name (or IP address) must be separated from a port with a colon. A Nameserver host name (or IP address), and port entry, must be separated from another entry with a comma. Used with IAGLOBAL_IMPORT_NAMESERVER_FILE=false, and IAGLOBAL_IMPACT_SERVER_RESULT=1.
IAGLOBAL_USER_REGISTRY_FILE_SELECTED	Set the value to 1 for the file based user registry, and 0 for the ObjectServer based user registry. **Important:** If you selected the ObjectServer based user registry, the installer will run successfully even if the ObjectServer information you provided is incorrect. However, the ObjectServer users may not get created in Netcool/Impact. To fix this run the $TIP_HOME/bin/confvmm4ncos script after the installation, and restart the server.
IAGLOBAL_OBJSERV_NAME	The name of the ObjectServer that is used as your event source. By default, it is the Netcool/OMNIbus server, NCOMS.
IAGLOBAL_OBJSERV_HOST_NAME	A fully qualified host name (or IP address) of your ObjectServer. This parameter has no default value, so you have to provide some value.
IAGLOBAL_OBJSERV_PORT	The ObjectServer port. The default is 4100.
IAGLOBAL_OBJSERV_USER	The user account used to connect as to the ObjectServer. The default is root.
IAGLOBAL_OBJSERV_PASSWORD_S	The password used with IAGLOBAL_OBJSERV_USER.
IAGLOBAL_EWAS_HTTP_PORT ... IAGLOBAL_EWAS_REST_NOTIFICATION_PORT	A set of ports that are used by the Impact Server. Used with IAGLOBAL_IMPACT_SERVER_RESULT=1.
IAGLOBAL_NCI_INSTANCE_NAME	The name of the Impact Server being installed. The default is NCI. Used with IAGLOBAL_IMPACT_SERVER_RESULT=1.
IAGLOBAL_NCI_CLUSTER_GROUP	The name of the cluster that the Impact Server is to join. The default is NCICLUSTER. Used with IAGLOBAL_IMPACT_SERVER_RESULT=1.
IAGLOBAL_NCI_CMD_LINE_PORT	The port used by the command line service for the new instance. The default is 2000. Used with IAGLOBAL_IMPACT_SERVER_RESULT=1.
IAGLOBAL_NCI_DB_PORT	The port used by the database server. The default is 5435. Used with IAGLOBAL_IMPACT_SERVER_RESULT=1.

Table 3. Silent installation parameters (continued)

Parameter	Description
IAGLOBAL_VERSION_CONTROL_TYPE	Specify which version control system you want to use. The default value is Impact Subversion, NCI_VCS_IMPACTSVN. This is installed automatically with the product.
VERCTLPATH	The version control path. You must provide a fully qualified path, because there is no default value. Required if IAGLOBAL_VERSION_CONTROL_TYPE is not of the IMPACT_SVN value.
VERCTL_REPOSITORY	The version control repository. You must provide some value, because there is no default value. Required if IAGLOBAL_VERSION_CONTROL_TYPE is not of the IMPACT_SVN value.

Installing in a shared environment

If you have an installed product, for example TBSM, that uses Tivoli Integrated Portal (TIP), you can install Netcool/Impact 6.1 into the existing TIP.

Netcool/Impact requires TIP version 2.2 or better. If your existing TIP environment is using TIP 2.1, you must install the TIP Feature Pack 2.2.0.1, before you can install Netcool/Impact.

Netcool/Impact 6.1.0.1 requires TIP version 2.2.0.7 or better.

See the Tivoli Netcool/Impact wiki for information about obtaining the TIP feature packs.

Typical scenarios where an existing TIP installation is reused are the following:

- Installing the Impact Server on a dedicated machine, and the GUI Server on the machine, where TIP is already installed. This scenario is recommended for most production environments.
- Installing both Netcool/Impact components on the machine, where TIP is already installed.

Important: If you are installing on a Windows system, where the Tivoli Monitoring Agent is installed, you must stop the agent before starting the installation.

Installing the GUI Server into an existing TIP installation

Use this procedure to install the Impact Server on a dedicated machine, and the GUI Server onto a machine where TIP is already installed.

Before you begin

It is recommended to back up any components that are already installed on TIP. To back up the Tivoli Business Service Manager component, follow the guidelines in the *Performing advanced installations*, and *Restoring system from a backup* sections, of theTivoli Business Service Manager 6.1 *Installation Guide*.

Restriction: If you are installing the GUI Server into a Tivoli Integrated Portal server with WebGUI, the Impact Server does not function until the **wimconfig.xml** file is updated after the installation. For information about updating the **wimconfig.xml** file see "Updating the wimconfig.xml file" on page 25.

Procedure

1. Install the Impact Server on a dedicated machine.

   Use the procedure "Running the installer in GUI and console mode" on page 12.

   a. Select the advanced type of installation.
   b. Install the Impact Server component only.
2. Install the GUI Server on the machine where TIP is installed.

   Use the procedure "Running the installer in GUI and console mode" on page 12.

   a. The installer will detect an existing TIP installation. Choose to reuse an existing installation directory, and point the installer to the directory where TIP is installed.
   b. Select the advanced type of installation.
   c. Choose to install the GUI Server component only.

      Restriction: If TBSM is already installed on TIP, you will not be able to install the GUI Server because it is installed as part of the TBSM dashboard.

      Remember: The TIP user name and password used must match the TIP user name and password of the existing TIP installation for reuse.
   d. When asked about the Name Server point to the machine where the Impact Server is installed.

      Note: If you provide incorrect Name Server information, you can reconfigure it after the installation, using the `nci_configuration_utility` tool. For more information, see "Post installation utility" on page 30.
3. After the installation is finished, restart the TIP server to pick up the new installed component.

What to do next

If you installed Netcool/Impact to be a part of a TBSM solution, they need to be configured to share their nameservers. Both the TBSM data server and the Impact Server are running nameservers independently but they can be configured together as a cluster, with the TIP server a member of the cluster. To do that, you must run the `nci_configuration_utility` tool on all the servers. For more information, see "Sharing nameservers with TBSM" on page 66.

Installing both components into an existing TIP installation

Use this procedure to install the Impact Server, and the GUI Server onto a machine where TIP is already installed.

Before you begin

It is recommended to back up any components that are already installed on TIP. To back up the Tivoli Business Service Manager component, follow the guidelines in the *Performing advanced installations*, and *Restoring system from a backup* sections, of theTivoli Business Service Manager 6.1 *Installation Guide*.

Restriction: If you are installing the GUI Server into a Tivoli Integrated Portal server with WebGUI, the Impact Server does not function until the **wimconfig.xml** file is updated after the installation. For information about updating the **wimconfig.xml** file see "Updating the wimconfig.xml file."

Procedure

1. Run the Netcool/Impact installer on the machine where TIP is installed.

   Use the procedure "Running the installer in GUI and console mode" on page 12.

   a. The installer will detect an existing TIP installation. Choose to reuse an existing installation directory, and point the installer to the directory where TIP is installed.
   b. Select the advanced type of installation.
   c. Choose to install the Impact Server, and the GUI Server component.

      Restriction: If TBSM is already installed on TIP, you will not be able to install the GUI Server because it is installed as part of the TBSM dashboard.

      Remember: The TIP user name and password used must match the TIP user name and password of the existing TIP installation for reuse.
2. After the installation is finished, restart the TIP server to pick up the new installed component.

What to do next

If you installed Netcool/Impact to be a part of a TBSM solution, they need to be configured to share their nameservers. Both the TBSM data server and the Impact Server are running nameservers independently but they can be configured together as a cluster, with the TIP server a member of the cluster. To do that, you must run the `nci_configuration_utility` tool on all the servers. For more information, see "Sharing nameservers with TBSM" on page 66.

Post installation steps

Perform any of the following additional configuration steps that are applicable to your installation profile.

Procedure

- Review the installation logs, especially the logs found in the `$IMPACT_HOME/logs` directory, to verify that the software has been installed correctly, or to troubleshoot installation errors.
- Log on to the deployed components to verify if they have been successfully installed.
- If you were installing an Impact Server to be a part of a server cluster follow the steps in Chapter 5, "Server clustering," on page 63.
- If you want to use SSL for communication with the GUI Server follow the steps in Chapter 10, "Secure communication," on page 113.
- (Windows platforms) Install the JRExec server.

Updating the wimconfig.xml file

If you are installing the GUI Server into a Tivoli Integrated Portal server with WebGUI, the Impact Server does not function until the **wimconfig.xml** file is updated.

About this task

After installation, you must complete the following steps to update the host name for Netcool OMNIbus authentication. If you have a split installation see the topic, *Updating the wimconfig.xml file on a split installation* for more information.

Procedure

1. Stop the Impact Server.
2. Locate the **wimconfig.xml** file in `$NCHOME/tipv2/profiles/ImpactProfile/config/cells/ImpactCell/wim/config/wimconfig.xml`.
3. Edit the **wimconfig.xml** file and add the Object server host name to the following section of the file:

   ```
   <config:CustomProperties name="host1" value=""/>
   ```

   For example:

   ```
   <config:CustomProperties name="host1" value="<test.ibm.com>"/>
   ```

4. Check that the ObjectServer password in the **wimconfig.xml** file is correct. To update the password with the correct encrypted value, complete the following steps:
   a. Run the **confvmm4ncos.sh** command. The command updates the **wimconfig.xml** file.
   b. Copy the **wimconfig.xml** file to the Impact Profile.
   c. Check that the result matches the existing encrypted password in the following property:

      ```
      <config:CustomProperties name="password"
      value="<encrypted password>"/>
      ```

5. Restart the Impact Server.

Updating the wimconfig.xml file on a split installation

Use the following steps for a split installation.

Procedure

1. Stop the Impact Server.
2. Make a back up copy of the **wimconfig.xml** file in `$NCHOME/tipv2/profiles/ImpactProfile/config/cells/ImpactCell/wim/config/wimconfig.xml`.
3. Check that the ObjectServer password in the **wimconfig.xml** file is correct. To update the password with the correct encrypted value, complete the following steps:
   a. Edit the **confvmm4ncos.sh** script. Change *TIPProfile* to *ImpactProfile*, and *TIPCell* to *ImpactCell* in the script.
   b. Run the **confvmm4ncos.sh** command. The command updates the **wimconfig.xml** file.
   c. Copy the password that is encrypted in the newly generated **wimconfig.xml** into the backup **wimconfig.xml** file.
   d. Return the original **wimconfig.xml** file back to its original location `$NCHOME/tipv2/profiles/ImpactProfile/config/cells/ImpactCell/wim/config/wimconfig.xml`.
4. Restart the Impact Server.

Restoring the system from backup

You can restore the system to the original state using the backup that was created during the installation.

About this task

The procedure is for a system that was installed to an existing Tivoli Integrated Portal directory.

Procedure

1. Stop the Impact Server.
   For details, refer to "Stopping and starting the Impact Server" on page 42.
2. Stop the Tivoli Integrated Portal server.
   For details, refer to "Stopping and starting the GUI Server" on page 47.
3. Stop all other applications that use the `$TIP_HOME` directory.
4. Run the `restore-backup` script that is located in the `Impact/impact` directory of the Netcool/Impact installation image.
   - UNIX:
     `./restore_backup.sh` *`backup_directory`*
   - Windows:
     `restore_backup.bat` *`backup_directory`*

   The *backup_directory* is the fully qualified path to the backup directory that you specified during the installation.
5. Start the Impact Server.
   For details, refer to "Stopping and starting the Impact Server" on page 42.
6. Start the Tivoli Integrated Portal server.
   For details, refer to "Stopping and starting the GUI Server" on page 47.

Increasing the memory for the Java Virtual Machine on the Impact profile

When the installation is complete, increase the Java Virtual Machine memory heap size settings from the default values.

For Netcool/Impact, the default value of `Xms` is 256 MB and the default value of `Xmx` is 1200 MB for 32 bit and 64 bit installations. It is better to increase the heap size to more than the default setting of 1200MB as it can improve performance. In a 64 bit installation, as a guideline increase the heap setting to 80% of the free memory on your system.

Consider the following factors when you are increasing the heap size in either a 32 bit or 64 bit installation:

- Event flow volume
- The number of SQL data types and the amount of data in them
- Internal data types and the size of them
- The number of event readers and event listeners
- The number of hibernations

For details about increasing the JVM memory heap size, see "Setting the memory for the Java Virtual Machine on the Impact profile" on page 90.

Configuring the default ObjectServer data source

You can configure the primary event source for your Netcool/Impact deployment after the installation, by editing the default ObjectServer data source.

The installation program creates the default ObjectServer data source in $IMPACT_HOME/etc/*NCI*_defaultobjectserver.ds, where *NCI* is the name of the server instance. You can configure the connection to the ObjectServer after the installation in one of the following ways:

- Using the data source editor to edit the ObjectServer data source.

  For more information, see "Working with data sources" in the *User Interface Guide*.
- Using the $TIP_HOME/bin/confvmm4ncos script.

Attention: Do not edit the *NCI*_defaultobjectserver.ds file manually.

Adding JDBC drivers and third-party .jar files to the shared library

Use this procedure to add a JDBC driver or third-party .jar files to the Netcool/Impact shared library.

About this task

You must copy the required JDBC drivers to the $IMPACT_HOME/dsalib directory. You can also copy any third-party JAR files that you require to the same directory. For example, if you have specific Java classes that you want use with Java policy functions in Netcool/Impact, you add the JAR files to this directory.

Procedure

1. Obtain the appropriate JDBC driver according to the DSA specification or the third party .jar files.
2. Copy the JDBC driver or third party .jar files to the $IMPACT_HOME/dsalib directory.

   This directory is created during the installation, and initially it is empty.
3. Restart the Impact Server.

What to do next

In a clustered configuration, you must repeat this procedure for each server in the cluster because files in the $IMPACT_HOME/dsalib directory are not replicated between cluster members. All the servers in the cluster should be stopped while you are performing this procedure.

Creating the JRExec service

The JRExec server is a component used to run external commands, scripts, and applications from within a policy. The JRExec server is automatically installed on the system when you install Tivoli Netcool/Impact. On Windows systems, however, you must manually create the JRExec service after installation.

Procedure

To create the JRExec service, enter the following command at a command prompt:

```
%NCHOME%/impact/bin/nci_jrexec -i nci_jrexec.conf
```

Configuring a Netcool/Impact server to use OMNIbus as the user repository

You can configure Netcool/Impact so that Netcool OMNIbus is used as the user repository for a Netcool/Impact server.

About this task

The user repository is selected when you install Netcool/Impact. If you want to configure an alternative user repository, you must do an advanced installation and choose File Registry as your user repository. You can then configure an alternative user repository after the installation is finished.

Procedure

1. Run the utility script. If you use a UNIX system, run the `confvmm4ncos_full.sh` script. If you use a Windows system, run the `confvmm4ncos_full.bat` script. The script is in the `$IMPACT_HOME/tipv2/bin` directory. Enter the following parameters:

Table 4. Parameters for user registry utility script

Parameter	Description
`wasdir`	The root directory location of the server installation (equivalent to $IMPACT_HOME)
`profile`	TIPProfile for a GUI Server server or ImpactProfile for an Impact Server
`cell`	TIPCell for a GUI Server or ImpactCell for an Impact Server.
`username`	Netcool OMNIbus user name to connect to Netcool OMNIbus (for example, root)
`password`	Password for the specified user name (if you specify a null or empty password, enter two double quotation marks [""])
`host1`	Host name of the Netcool OMNIbus server (the host name of the primary host if you are using a failover environment)
`port1`	Port number (for example, 4100; this number is the port number of the primary server if you are using a failover environment)
`host2`	This parameter is optional. Host name of a backup Netcool OMNIbus server if you are using a failover environment
`port2`	This parameter is optional. Port number of a backup Netcool OMNIbus server if you are using a failover environment
`Realm`	This parameter is optional. The authentication realm to configure. The default value is `defaultWIMFileBasedRealm`.

2. Restart the server.
3. Verify that the GUI Server is communicating correctly with the Netcool OMNIbus server.
   - Log in to the Tivoli Integrated Portal as an administrator.
   - Click **Users and groups**, then click **Manage users**.

- Click **Search**. The users that are defined in the Netcool OMNIbus repository are displayed in the list of users that is returned. The value for the **Base entry for the default parent** field for Netcool OMNIbus users is `o=netcoolObjectServerRepository`.

4. Verify that the Impact Server is communicating correctly with the Netcool OMNIbus server.
   - Use a telnet application to connect to the command port of the Impact Server. The default port for the Impact Server is 2000.
   - Use an Netcool OMNIbus user to log in to the Impact Server.
   - Enter `quit ();` to exit the utility

Post installation utility

You use the post installation utility to create a new instance of the Impact Server, or configure Name Servers.

The post installation utility, `nci_configuration_utility`, is located in the `$IMPACT_HOME/install` directory. You can run it in GUI mode, or from the command line. When run in GUI mode you can select one of the two options:

Create a new Impact Server

Select this option to add a new Impact Server to the application server. Not recommended for production environments.

Restriction: You cannot use this option on a machine where only the GUI Server is installed.

Configure Name Servers

Select this option to change the name servers configuration for your GUI Server, or Impact Server.

This option can also be used, in a situation when Netcool/Impact is installed "distributed" with TBSM, and you want them to form a cluster at a later time. As TBSM data server is an Impact Server with TBSM customizations, it also has the `nci_configuration_utility` installed with it. You run this utility on the TBSM server, and configure the same list of nameservers as you would on an Impact Server. For more details, see "Sharing nameservers with TBSM" on page 66.

Tip: The post installation utility can also be used from the command line. For information about the command line usage, run the `nci_configuration_utility` with the `-?` switch.

Migrating to version 6.1

If you are running a previous version of Netcool/Impact you can migrate it to version 6.1 by using the provided migration script.

You can migrate only versions 5.x directly to 6.1. You can migrate from a split 5.x server configuration to a single machine, or a split server 6.1 configuration. For earlier versions, you must complete any required intermediate upgrades first, up to version 5.x.

Before you migrate

- Web Services listener:

Any web services client that used to access Netcool/Impact in version 5.1.1 must be updated to use basic HTTP authentication in 6.1. A user name and password is required to access the web services listener.

You must assign the **impactFullAccessUser** and **bsmAdministrator** roles to the user who uses the web services. For more information about how to assign these roles, see "Using WebServices through the command line" on page 140 and "Mapping groups and users to roles" on page 142.

Also see the *Authentication for the web services listener* topic in the Netcool/Impact information center or in the *DSA Reference Guide* PDF file.

The migration script

The migration script, `migrate`, is copied to the destination drive during the installation, to the `$IMPACT_HOME/install/migration` directory.

Before you can use the migration script, extract the migration script file from the `$IMPACT_HOME/install/migration` directory.

The migration script can only be run from the command line, and uses the following syntax:

```
migrate -command arguments
```

The script takes the following commands and arguments:

Table 5. The migration script commands and arguments

Command	Arguments	Description
`-migrate`	*old_NCHOME_dir new_NCHOME_dir* `-password` *tippass*	Export and import the GUI Server, and Impact Server. This option is only used when 5.x and 6.1 are on the same host. *tippass* is your TIP password.
`-export`	*old_NCHOME_dir export_dir*	Export the GUI Server, and Impact Server.
`-import`	*new_NCHOME_dir import_dir* `-password` *tippass*	Import the GUI Server, and Impact Server.
`-exportImpactServer`	*old_NCHOME_dir export_dir*	Export the Impact Server.
`-importImpactServer`	*new_NCHOME_dir import_dir* `-password` *tippass*	Import the Impact Server.
`-exportGUIServer`	*old_NCHOME_dir export_dir*	Export the GUI Server.
`-importGUIServer`	*new_NCHOME_dir import_dir* `-password` *tippass*	Import the GUI Server.
`-help`	N/A	For help on the available commands and arguments.

Important: Windows paths that contain spaces must be enclosed within inverted commas.

Examples

This example shows how to migrate GUI Server, and Impact Server version 5.x into 6.1, on a UNIX system:

```
./migrate -migrate /opt/ibm/netcool /opt/IBM/tivoli -password netcool
```

This example shows how to export the Impact Server version 5.x, to an export directory, on Windows:

```
migrate.bat -exportImpactServer "C:\Program Files\IBM\netcool\impact" C:\temp
```

Scenario 1: Migrating 5.x servers to version 6.1 that is installed on the same machine

This procedure gives an example of migration from Netcool/Impact version 5.x. to 6.1 when both the source and the destination server are on the same machine.

About this task

For simplicity's sake, we assume a deployment of a single GUI Server, and Impact Server. In a clustered environment, you will repeat the steps of the procedure for each cluster member. Also, note that the installation paths that are used here for both version 5.x, and 6.1 are default installation paths.

Procedure

1. Stop your 5.x deployment.

   Use the `ewas` script located in the `NCHOME/bin/ewas` directory to stop the application server where both the GUI Server, and the Impact Server are running. On Windows, you can also use the Services console to stop the "IBM Netcool Impact Server" process.
2. Install Netcool/Impact 6.1 on the same machine.

   Install both deployment components, the GUI Server, and Impact Server. If you are installing in silent mode, select your installation options by editing the silent installation response file. For more information, see "Silent mode" on page 20.

   Important: You must configure the Nameservers for your 6.1 cluster correctly at this stage, because the Nameserver configuration is not migrated.
3. Use the `$IMPACT_HOME/bin/nci_removeserver.bat/sh` tool to remove the pre-configured NCI impact instance so it does not clash with the imported NCI instance from 5.x.

   For example, on a UNIX system, you use the following command:

   ```
   ./nci_removeserver NCI
   ```

   Choose 'yes' to confirm the operation.
4. Navigate to the directory that contains the migration script and run the script with the following parameters:
   - UNIX:

     ```
     ./migrate -migrate /opt/ibm/netcool /opt/IBM/tivoli -password netcool
     ```

     where *netcool* is your TIP password.
   - Windows:

     ```
     migrate.bat -migrate "C:\Program Files\IBM\netcool"
     "C:\Program Files\IBM\tivoli" -password netcool
     ```
5. Wait for the script to complete all its tasks.

   At the end of the migration, if it was successful, the script will display a `BUILD SUCCESSFUL` message to the command line.
6. Log on to the GUI Server version 6.1 (Tivoli Integrated Portal server), and double check that all your 5.x settings were migrated successfully.
7. Check the status of the server cluster, and the imported data in the configuration documenter.

For more information about using the configuration documenter, see the *Configuration documenter* chapter in the *User Interface Guide*.

8. Perform additional steps as outlined in the "Post migration steps" on page 36 procedure.

Scenario 2: Migrating 5.x servers to version 6.1, with both 6.1 servers installed on different machines

This procedure gives an example of migration from Netcool/Impact version 5.x where the destination server, 6.1, is installed on a different machine than the source server, 5.x. Moreover, the Impact Server, and the GUI Server in the 6.1 installation are installed on dedicated machines.

About this task

For simplicity's sake, we assume a deployment of a single GUI Server, and Impact Server. In a clustered environment, you will repeat the steps of the procedure for each cluster member. Also, note that the installation paths that are used here for both version 5.x, and 6.1 are default installation paths.

Procedure

1. Install Netcool/Impact 6.1.

   Install the GUI Server, and the Impact Server, each on a dedicated machine. Make sure that only one component is selected in the installation components selection screen. If you are installing in silent mode, select your installation options by editing the silent installation response file. For more information, see "Silent mode" on page 20.

   Important: During the installation, make sure to configure the Name Servers for your 6.1 cluster correctly because the Name Server configuration is not migrated from 5.x.

2. Use the `nci_removeserver` tool to remove the pre-configured NCI impact instance so it does not clash with the imported NCI instance from 5.x. For more information, see "nci_removeserver" on page 126.

3. Copy the `$IMPACT_HOME/install/migration/migration.zip` from the 6.1 installation to a temporary directory on the machine where the GUI Server 5.x is installed.

   For example, on Windows you can use the `C:\temp` directory, and on Linux `/home/impact/tmp`, or any directory to which you have full access rights.

4. Stop your 5.x deployment.

   Use the `ewas` script located in the `NCHOME/bin/ewas` directory to stop the application server where both the GUI Server, and the Impact Server are running. On Windows, you can also use the Services console to stop the "IBM Netcool Impact Server" process.

5. Go to the directory that contains the migration script and run the script with the following parameters:

   - UNIX systems:

     ```
     ./migrate -exportGUIServer NCHOME
     TEMPDIR
     ```

     Where *NCHOME* is the installation directory of Netcool/Impact 5.x, and *TEMPDIR* is a temporary directory. For example:

     ```
     ./migrate -exportGUIServer /opt/ibm/netcool /home/impact/tmp
     ```

The example works if you installed Netcool/Impact 5.x as the "impact" UNIX user, in the default directory.

- Windows:

  ```
  migrate.bat -exportGUIServer NCHOME
  TEMPDIR
  ```

  For example:

  ```
  migrate.bat -exportGUIServer "C:\Program Files\IBM\netcool" C:\temp
  ```

  The example works if you have Netcool/Impact 5.x installed in the default directory.

The migration script outputs all the settings, and saved data from your GUI Server to your temporary directory.

6. Wait for the script to complete all its tasks.

   At the end of the migration, if it was successful, the script will display a `BUILD SUCCESSFUL` message to the command line.

7. Copy the contents of the temporary directory to a temporary location on the machine where the GUI Server version 6.1 is installed.

   On Windows, you can use the `C:\temp` directory, and on Linux `/home/impact/tmp`, or any directory to which you have full access rights.

8. Go to the directory that contains the migration script and run the script with the following parameters:

   - UNIX systems:

     ```
     ./migrate -importGUIServer IMPACT_HOME
     TEMPDIR -password tippass
     ```

     Where *IMPACT_HOME* is the installation directory of Netcool/Impact 6.1, *TEMPDIR* is a temporary directory, and *tippass* is your Netcool/Impact administrator password in the Tivoli Integrated Portal. For example:

     ```
     ./migrate -importGUIServer /opt/IBM/tivoli /home/impact/tmp -password netcool
     ```

     The example works if you installed Netcool/Impact 6.1 as the "impact" UNIX user, in the default directory, and with the default Netcool/Impact administrator password in the Tivoli Integrated Portal.

   - Windows:

     ```
     migrate.bat -importGUIServer IMPACT_HOME
     TEMPDIR -password tippass
     ```

     For example:

     ```
     migrate.bat -importGUIServer "C:\Program Files\IBM\tivoli\impact"
     C:\temp -password netcool
     ```

     The example works if you have Netcool/Impact 6.1 installed in the default directory, and with the default Netcool/Impact administrator password in the Tivoli Integrated Portal.

   The migration script imports all the settings, and saved data from your temporary directory into the GUI Server.

9. Wait for the script to complete all its tasks.

   At the end of the migration, if it was successful, the script will display a `BUILD SUCCESSFUL` message to the command line.

10. Copy the contents of the `migration` directory to a temporary directory on the machine where the Impact Server 5.x is installed.

    For example, on Windows you can use the `C:\temp` directory, and on Linux `/home/impact/tmp`, or any directory to which you have full access rights.

11. Go to the directory that contains the migration script and run the script with the following parameters:

- UNIX systems:

  ```
  ./migrate -exportImpactServer NCHOME TEMPDIR
  ```

  Where *NCHOME* is the installation directory of Netcool/Impact 5.x, and *TEMPDIR* is a temporary directory. For example:

  ```
  ./migrate -exportImpactServer /opt/ibm/netcool /home/impact/tmp
  ```

  The example works if you installed Netcool/Impact 5.x as the "impact" UNIX user, in the default directory.

- Windows:

  ```
  migrate.bat -exportImpactServer NCHOME TEMPDIR
  ```

  For example:

  ```
  migrate.bat -exportImpactServer
   "C:\Program Files\IBM\netcool" C:\temp
  ```

  The example works if you have Netcool/Impact 5.x installed in the default directory.

The migration script outputs all the settings, and saved data from your Impact Server to your temporary directory.

12. Copy the contents of the temporary directory to a temporary location on the machine where the Impact Server version 6.1 is installed.

    For example, on Windows you can use the `C:\temp` directory, and on Linux `/home/impact/tmp`, or any directory to which you have full access rights.

13. Go to the directory that contains the migration script and run the script with the following parameters:

    - UNIX systems:

      ```
      ./migrate -importImpactServer IMPACT_HOME
      TEMPDIR -password tippass
      ```

      Where *IMPACT_HOME* is the installation directory of Netcool/Impact 6.1. The default directory is `/opt/IBM/tivoli`, *TEMPDIR* is a temporary directory, and *tippass* is your Netcool/Impact administrator password in the Tivoli Integrated Portal.

      ```
      ./migrate -importImpactServer /opt/IBM/tivoli /home/impact/tmp -password netcool
      ```

      The example works if you installed Netcool/Impact 6.1 as the "impact" UNIX user, in the default directory, and with the default Netcool/Impact administrator password in the Tivoli Integrated Portal.

    - Windows:

      ```
      migrate.bat -importImpactServer IMPACT_HOME
      TEMPDIR -password tippass
      ```

      For example:

      ```
      migrate.bat -importImpactServer "C:\Program Files\IBM\tivoli"
      C:\temp -password netcool
      ```

      Where *IMPACT_HOME* is the default directory, `C:\Program Files\IBM\tivoli`, *TEMPDIR* is a temporary directory, and *tippass* is your Netcool/Impact administrator password in the Tivoli Integrated Portal.

      The example works if you have Netcool/Impact 6.1 installed in the default directory, and with the default Netcool/Impact administrator password in the Tivoli Integrated Portal.

    The migration script imports all the settings, and saved data from your temporary directory into the Impact Server.

14. Wait for the script to complete all its tasks.

    At the end of the migration, if it was successful, the script will display a `BUILD SUCCESSFUL` message to the command line.

15. Log on to the GUI Server version 6.1 (Tivoli Integrated Portal server), and double check that all your 5.x settings were migrated successfully.
16. Check the status of the server cluster, and the imported data in the configuration documenter.

    For more information about using the configuration documenter, see the *Configuration documenter* chapter in the *User Interface Guide*.
17. Perform additional steps as outlined in the "Post migration steps" procedure.

Post migration steps

You must complete some of the migration-related tasks manually.

Procedure

- Event Isolation and Correlation (EIC), and Maintenance Window Management (MWM).

  If you installed 6.1 using a non-default cluster name, NCICLUSTER, the EIC, and MWM features will not work after the migration. In this case, you must run the `$IMPACT_PROFILE/bin/nc_ant` tool on the GUI Server to which you migrated. Use this command to create the operator view files for the EIC and MWM:

  ```
  nc_ant  -f installAddOnOpview.xml  -DCLUSTERNAME=ClusterName NewCluster
  ```

  Where *ClusterName* is the cluster name that is being migrated from 5.x. The NewCluster parameter is a hardcoded parameter and must not be changed.
- Migrating the JMS configuration from 5.1.x to 6.1.

  Migrating the JMS configuration from 5.1.x to 6.1 is not supported. After the migration, the JMSMessageListener service fails to start, because there is no JMS data source. You must manually reconfigure your policies, add a data source, and reconfigure the JMS listener after the migration.

What to do next

After you make sure that your settings, and saved data were imported successfully to 6.1, you can uninstall Netcool/Impact 5.x. Alternatively, you can disable automatic startup on your 5.x servers and keep them as backup.

Installation logs

Review the installation logs to make sure that the installation was successful.

Attention:

When the Impact Server is started during the installation, the following message is displayed in the `netcool.log` file:

```
Unable to login to configured name servers host [<hostname>], port [<port>]
```

The message occurs because the Name Server can take time to initialize while Netcool/Impact starts. You can ignore this message. It is normal and it disappears when the Name Server server is initialized.

IMPACT6.1.0.0_install-*xx*.log
: *xx* can be 00 or 01 or 02. The installer puts this log in the user home directory. This log file contains messages generated during the installation process. You can also use it to troubleshoot installation problems.

$IMPACT_HOME/logs
For additional debugging information review the logs found in this directory. These logs will help to troubleshoot installation problems.

Uninstalling the deployment

You can uninstall both Netcool/Impact components, the Impact Server, and the GUI Server using the same system tools.

On UNIX systems, you run the uninstaller script, `uninstall` which is located in the `$IMPACT_HOME/_uninst/IMPACTInstall61` directory.

On Windows systems, you use the **Add/Remove Programs** tool.

Important:

If for any reason you need to uninstall TBSM or Netcool/Impact, do not uninstall the external registry, unless you are sure no other applications need that user registry. The TBSM Data server, Dashboard server, and Netcool/Impact servers can all use the same user registry. For example, if you use a Netcool/OMNIbus ObjectServer as your user registry for a TBSM dashboard server and Netcool/Impact, you will disable Netcool/Impact if you uninstall the ObjectServer when you uninstall a TBSM server. Use the same caution if you use an LDAP server as your user registry.

Uninstalling the deployment on Windows platforms

On Windows platforms, you uninstall Netcool/Impact components, using the **Add/Remove Programs** tool.

Procedure

1. Select **Start** > **Control Panel** > **Add/Remove Programs**, and then **IBM Tivoli Netcool/Impact** > **Remove** to remove both Netcool/Impact deployment components.

   Restriction: To remove the GUI Server from a TBSM deployment run the TBSM uninstallation. When Netcool/Impact is running as part of a TBSM deployment, TBSM will block the uninstallation of the GUI Server, but it will remove the Netcool/Impact uninstallation routine.
2. Follow the on-screen prompts to remove the software.
3. Remove the directories where Netcool/Impact components were installed.

   Use Windows Explorer, or the `rmdir` command at the command prompt.

Uninstalling the deployment on UNIX platforms

On UNIX systems, you uninstall Netcool/Impact components, by running the uninstaller script, in GUI or console mode.

Procedure

1. At the command prompt, navigate to the `$IMPACT_HOME/_uninst/IMPACTInstall61`.
2. Run the uninstaller script.
   - To run the uninstaller in GUI mode:

     ```
     ./uninstall
     ```

The uninstaller presents a series of graphical windows that guide you through the uninstallation process.

- To run the uninstaller in console mode:

  ```
  ./uninstall -i console
  ```

  Remember: If you are running the uninstaller remotely, using telnet, or another command-line application, you must run it in console mode.

3. Follow the on-screen prompts to remove the software.
4. Remove the directories where Netcool/Impact was installed.

   Use the `rm -r` command.

Chapter 3. Managing the Impact Server

You manage your Impact Server deployment using various command line scripts.

Creating Impact Server instances

An instance of the Impact Server is automatically created and deployed during the installation but you can create additional instances of the server using the post installation utility.

The post installation utility creates an EAR file for the Impact Server and deploys it into the Java application server where Netcool/Impact components reside. The application server automatically detects the new EAR file after deployment and starts the instance of the Impact Server.

Important: You must create a new server under the same user account that you used to install the first instance of the Impact Server.

A set of properties files and other supporting files are also created that define the behavior of the server instance. These files are located in `$IMPACT_HOME/etc`. Files related to the server instance have the prefix *NCI_* where *NCI* is the name of the Impact Server. For example, `NCI_server.props` is the name of the server properties file for the instance named NCI.

For more information about running the post installation utility, see "Post installation utility" on page 30.

Using Impact Server administration scripts

You can start and stop the Impact Server independent of the application server using two scripts, `nci_server`, and `nci_shutdown`, that are located in the `$IMPACT_HOME/bin` directory.

Before you begin

Make sure the embedded version of WebSphere Application Server is running before you start and stop the Impact Server.

Procedure

- To stop a server instance, enter the following command at a command prompt:

  `nci_shutdown` *`server`*

  where *server* is the name of the Impact Server instance.
- To start a server instance, enter the following command at a command prompt:

  `nci_server` *`server`*

Configuring RMI ports

The Impact Server opens a number of ports on a machine for Java Remote Method Invocation (RMI) operations.

There are a few exposed properties that allow you to control which ports are used. By default, a random port is opened on the host system that is used by the RMI registry. You can control RMI ports where a firewall exists on a network and the exact control of which ports should be opened is necessary. You specify the properties in the `$IMPACT_HOME/etc/NCI_server.props` properties file.

To control which ports are used by Impact when exporting RMI objects, you need to provide values for the following properties:

impact.server.rmiport
This property specifies the port that Impact uses when starting its RMI registry.

impact.rmiPortRangeStart
This property specifies the minimum value of the port number that Impact uses for its RMI registry.

impact.rmiPortRangeEnd
This property specifies the maximum value of the port number that Impact uses for its RMI registry.

These values should span a range of approximately 100 ports on your machine and they force Impact to open the ports in this range only. Once specified, Impact Server uses this range of ports when exporting RMI objects.

Important: In a cluster configuration, if Impact Servers are communicating through a firewall, the secondary Impact Server must be able to access the Name Server on the primary, and the other way round. To achieve that, open the Name Server port on the firewall, by default 9080, or 9081 for SSL. Also, open the port on which TIP is running (by default 16310, or 16311 for SSL), if the browser is outside the firewall.

Monitoring deployment components

Netcool/Impact provides a centralized logging feature that incorporates Apache Log4j logging functionality.

Monitoring server instances

You can monitor and log the status and activity of both the Impact Server and the GUI Server.

The server log contains status, debugging, and error messages generated during runtime. The log file, `netcool.log`, is located in the `$IMPACT_HOME/logs` directory.

When the log file reaches 10 MB in size the contents of the log are copied to a backup file and a new, empty copy of the file is created. Backup log files are named `netcool.log.n`, where *n* is an index number that identifies the file. When the maximum number of backup files is reached (by default, three) the oldest file (for example, `netcool.log.3`) is deleted and the other two files are renamed so that the oldest copy always has the largest index number of all of the files.

You can customize the logging properties by modifying the contents of the Log4j properties file, `log4j.properties`, that is located in the `$IMPACT_PROFILE/properties` directory.

Log4j properties file

This is a list of properties that you can modify in the `log4j.properties` file to change the default logging behavior.

Table 6. Log4j properties

Property	Description
`log4j.appender.NETCOOL.threshold`	Specifies that the deployment components must log all messages with a severity of `DEBUG` (default) or higher.
`log4j.appender.NETCOOL.file`	Specifies the path and filename of the log file. Default is `$IMPACT_HOME/logs/netcool.log`.
`log4j.appender.NETCOOL.append`	Specifies whether to append to an existing log file at startup. Default is `true`.
`log4j.appender.NETCOOL.maxBackupIndex`	Specifies the maximum number of backup files. Default is 3.
`log4j.appender.NETCOOL.maxFileSize`	Specifies the maximum size of log files. Default is `10MB`.

Monitoring services

Services running in the Impact Server can also print their activity and status messages to a log file.

You set the maximum number of backup files that can be created by services in the server properties file, `$IMPACT_HOME/impact/etc/`*`NCI`*`_server.props`, where *`NCI`* is the name of the Impact Server.

To set the maximum number of log files, modify the value of the following property:

`impact.server.logging.maxlogfiles=`*`n`*

where *n* is the maximum number of log files for each service. The default is 10. The maximum file size for a service log file is 10 MB.

You can override this value on a per service basis by specifying the `impact.servicename.maxbackupindex` property in the individual *`servicename`*`.props` files to set this value for each service. For more information about configuring logging for single services, refer to the *User Interface Guide*.

Deleting Impact Server instances

You can delete an Impact Server instance from by using the remove server script, `$IMPACT_HOME/bin/nci_removeserver`.

Procedure

- To remove server Impact Server run the following command at a command prompt:

  `./nci_removeserver` *`server_name`*

  where *server_name* is the instance of the server you want to remove. The script checks to see if the server instance is running before removing it from the installation. If the instance is running, the script automatically shuts it down before continuing with operations.

By default, the remove server script does not delete information about policies, data sources, data types, and services from the version control system. If you are using SVN as your version control system, you can run the remove SVN archives script to completely remove all archives related to a server instance. The remove SVN archives script is named `nci_svn_remove` and is located in the `$IMPACT_HOME/bin` directory.

- Optional: To run the remove SVN archives script, enter the following command at a command prompt:

  `./nci_svn_remove server_name`

  where *server_name* is the name of the deleted server instance.

Read-only mode

You use read-only mode in production environment to ensure that no changes can be made to the Netcool/Impact configuration in real time.

In read-only write access to a Impact Server is blocked and users cannot make any changes to data sources, data types, policies, or services. In read-only mode, the server can read, write, update, and delete data stored in data items, including those stored in internal data types. Read-only mode does not interfere with the operation of policies and services during server runtime. Read-only mode only prevents changes to the Netcool/Impact configuration. In addition, read-only mode does not affect the ability to change or use operator views.

You set read-only mode by manually editing the version control properties file, `$IMPACT_HOME/etc/NCI_versioncontrol.props`, where *servername* is the name of the Impact Server.

After you enable read-only mode, users are notified that data sources, data types, policies, and services are locked to a user named “read-only” when they try to make changes.

Enabling read-only mode

Follow this procedure to enable read-only mode.

Procedure

1. Edit the `impact.versioncontrol.readonlymode` property as follows:

   ```
   impact.versioncontrol.readonlymode=true
   ```

2. Restart the Impact Server for the change to take effect.

Stopping and starting the Impact Server

You stop and start the Impact Server by stopping and starting the application server where it is running.

Procedure

- On UNIX systems you use the application server administration script, `ewasImpactStartStop.sh`, to start and stop the application server.

  This script is located in the `$IMPACT_HOME/bin` directory.

  1. To stop eWAS, enter the following command at the command prompt:

     ```
     $IMPACT_HOME/bin/ewasImpactStartStop.sh stop
     ```

When prompted, provide the user name and password for the embedded version of WebSphere Application Server. The default credentials are `wasadmin` and `netcool`.

2. To start eWAS, enter the following command at the command prompt:

   ```
   $IMPACT_HOME/bin/ewasImpactStartStop.sh start
   ```

- Windows systems.

  In the **Start** menu, select **Control Panel** > **Administrative Tools** > **Services**.

  1. To stop eWAS, in the Services window, right-click **IBM Netcool Impact Server** and select **Properties**. In the Properties dialog box, click **Stop** and then click **OK**.
  2. To start eWAS, in the Services window, right-click **IBM Netcool Impact Server** and select **Properties**. In the Properties dialog box, click **Start** and then click **OK**.

  Remember: A version of the `ewasImpactStartStop.sh` script for Windows is the `ewasImpactStartStop.bat` file. It takes the same parameter as `ewasImpactStartStop.sh`.

Connection request to mail server

Connections made by the e-mail reader to a back-end mail server is a blocking activity.

As a result, connection attempts are subject to TCP/IP timeout in the event of any connectivity issues between Netcool/Impact and the mail server. Since this value varies depending on the operating system, by default Netcool/Impact waits a minimum of 15 seconds, before abandoning a connection attempt. If this value is insufficient, you can change the value manually in the e-mail reader service configuration file.

Changing the time interval on a connection request to the mail server

You can adjust the time interval on a connection request to the mail server.

Procedure

1. Open the `$IMPACT_HOME/etc/`*`NCI_servicename`*`.props` file and specify the property `impact.`*`servicename`*`.timeoutseconds`.
2. Restart the Impact Server server to implement the changes.

The Impact EAR file

The Impact Server program files are packaged into an enterprise archive file (EAR) that is managed by the Java Application Server (by default, the embedded version of WebSphere Application Server) where the Netcool/Impact components reside.

The EAR file is named according to the server instance, for example `instanceName-impact-6.1.0.0.ear`. For example, if the name of the Impact server is NCI, then the EAR file is named `NCI-impact-6.1.0.0.ear`. The ear file is located in the `$IMPACT_HOME/integration/installableApps` directory by default.

The new Impact EAR file is deployed into the running embedded version of WebSphere Application Server. The newly created EAR file is also copied into the `$IMPACT_HOME/integration/installableApps` directory. The embedded version of

WebSphere Application Server, however, does not run the EAR file from this location. This file represents a copy of the EAR file deployed in the server. If other changes are required to the Impact Server, this file is to be used as a baseline. The Impact Server stops and restarts itself after you install new libraries into it. You can confirm that the Impact Server has restarted again by viewing the `$IMPACT_HOME/logs/netcool.log` file and looking for the message indicating that Impact was started successfully.

Adding a new JAR file to an existing EAR file

You can use the `nci_ear_editor` command to add a Java Archive (JAR) file to an existing Enterprise Archive (EAR) file.

About this task

Before you begin, make sure that the Netcool/Impact server is running.

If you deploy the EAR file on multiple servers in a cluster, you must complete this procedure for each server in the cluster.

Procedure

1. To open the `nci_ear_editor` command, enter the following command:

   ```
   $NCHOME/bin/nci_ear_editor
   $NCHOME/eWAS/profiles/ImpactProfile/installableApps/
   <instance_name>-impact-6.1.0.ear/<temporary_directory>/
   <instance_name>-impact-6.1.0.ear
   ```

   where <*instance_name*> is the name of the server instance where the JAR file is deployed. <*temporary_directory*> is the temporary directory where you store the modified EAR file temporarily.

   For example:

   ```
   $NCHOME/bin/nci_ear_editor
   $NCHOME/eWAS/profiles/ImpactProfile/installableApps/
   NCI-impact-6.1.0.ear/tmp/NCI-impact-6.1.0.ear
   ```

2. For each JAR file that you want to add, perform the following tasks:
   a. To add an artifact to the EAR file, enter `1`.
   b. To add a jar artifact to the EAR file, enter `3`.
   c. Specify the fully qualified path and file name of the client jar file.
3. Enter `5` to finish the installation process and write the new EAR file to the temporary location that you specified when you started the script.
4. To deploy the modified EAR file again, use the `redeploy` command.

   For example, in the following command it is assumed that the temporary location of the modified EAR file is the `/tmp` directory:

   ```
   $NCHOME/bin/nc_ant -f $NCHOME/impact/install/new-server.xml redeploy
   -Dimpact.ewas.pw=<tipadminpw>
   -Dear.filename=/<temporary_directory>/<instance_name>
   -Impact-6.1.0.ear
   ```

   where <*instance_name*> is the name of the server instance where the EAR file is deployed. <*tipadminpw*> is the Tivoli Integrated Portal password.

   Note: For Windows, the `-Dear.filename` value takes a path with two forward slashes, for example:

   ```
   -Dimpact.ewas.pw=abc123 -Dear.filename=C:\\tmp\NCI-impact-6.1.0.ear
   ```

Results

After you deploy the EAR file again, the new JAR file and the modified EAR file are available in the server immediately. You do not have to restart WebSphere Application Server.

FIPS Compliance

Netcool/Impact supports the US Federal Information Processing Standard 140-2 (FIPS 140-2) when using cryptographic algorithms to encrypt and decrypt passwords.

For more information about FIPS, see http://csrc.nist.gov/publications/fips/

Enabling FIPS on the Impact Server

Netcool/Impact password encryption algorithms on the Impact Server use FIPS-approved cryptographic providers regardless of whether FIPS is enabled for the entire Impact Server.

About this task

If you want to configure SSL communications between the Netcool/Impact servers and you want the SSL encryption to use a FIPS-approved cryptographic provider, enable FIPS on all servers first and then configure SSL. The following procedure can be used to enable FIPS for the Impact Server:

Procedure

1. From `tip_home_dir/bin`, login to the Impact Server using the following command:

   ```
   wsadmin -username <username>  -password <password> -profileName ImpactProfile
   ```

2. To view the current security settings including the current value for `com.ibm.security.useFIPS`, run the following command:

   ```
   $AdminTask getActiveSecuritySettings
   ```

3. Modify the current setting using the command:

   ```
   $AdminTask setAdminActiveSecuritySettings  {-customProperties
   {"com.ibm.security.useFIPS=true"}  }
   ```

4. Configure the Impact Server to use FIPS algorithms for Java clients that must access enterprise beans:
   a. Open the `tip_home_dir/profiles/ImpactProfile/properties/ssl.client.props` file in a text editor.
   b. Change the `com.ibm.security.useFIPS` property value from `false` to `true`.
5. Configure the Impact Server to use FIPS algorithms for SOAP-based administrative clients that must access enterprise beans:
   a. Open the `tip_home_dir/profiles/ImpactProfile/properties/soap.client.props` file in a text editor.
   b. Add this line: `com.ibm.ssl.contextProvider=IBMJSSEFIPS`.
6. Configure `java.security` to enable IBMJCEFIPS:
   a. Open the `tip_home_dir/java/jre/lib/security/java.security` file in a text editor.
   b. Insert the IBMJCEFIPS provider (`com.ibm.crypto.fips.provider.IBMJCEFIPS`) before the IBMJCE provider,

and also renumber the other providers in the provider list. The IBMJCEFIPS provider must be in the `java.security` file provider list. See the following example:

Example

The IBM SDK `tip_home_dir/java/jre/lib/security/java.security` file is similar to this example when IBMJCEFIPS is enabled.

```
security.provider.1=com.ibm.crypto.fips.provider.IBMJCEFIPS
security.provider.2=com.ibm.crypto.provider.IBMJCE
security.provider.3=com.ibm.jsse.IBMJSSEProvider
security.provider.4=com.ibm.jsse2.IBMJSSEProvider2
security.provider.5=com.ibm.security.jgss.IBMJGSSProvider
security.provider.6=com.ibm.security.cert.IBMCertPath
security.provider.7=com.ibm.crypto.pkcs11.provider.IBMPKCS11
security.provider.8=com.ibm.security.cmskeystore.CMSProvider
security.provider.9=com.ibm.security.jgss.mech.spnego.IBMSPNEGO
```

Chapter 4. Managing the GUI Server

The default GUI Server for Netcool/Impact 6.1 is the Tivoli Integrated Portal server.

If you chose to install the GUI Server as one of the deployment components, the installer sets all of the default configuration properties for the Tivoli Integrated Portal server, so you can log on to, and start working with Netcool/Impact immediately after the installation.

The Tivoli Integrated Portal server starts and stops automatically when the application server, where it resides, is started or stopped, but you can also start, and stop it independently of the application server.

Stopping and starting the GUI Server

Use this procedure to stop, and start the GUI Server.

Procedure

- On UNIX systems you use the Tivoli Integrated Portal server (TIP) administration script, `$IMPACT_HOME/bin/ewasGUIStartStop.sh`.
  1. To stop TIP, enter the following command at the command prompt:

     ```
     $IMPACT_HOME/bin/ewasGUIStartStop.sh stop
     [-username adminuser -password adminpassword]
     ```

     where *adminuser* is the TIP profile user (tipadmin by default), and *adminpassword* is the TIP profile password. You provided them during the installation. These arguments are optional.
  2. To start TIP, enter the following command at the command prompt:

     ```
     $IMPACT_HOME/bin/ewasGUIStartStop.sh start
     [-username adminuser -password adminpassword]
     ```

  Tip: You can also use the `stopServer.sh` and `startServer.sh` scripts in the `$TIP_HOME/profiles/TIPProfile/bin` directory.
- On Windows systems, you use the Services Extension in the Microsoft Management Console.
  1. To stop TIP, in the Services Extension window right-click **Tivoli Integrated Portal**, and select **Properties**. In the Properties dialog box, click **Stop**, and then click **OK**.
  2. To start TIP, in the Services Extension window right-click **Tivoli Integrated Portal**, and select **Properties**. In the Properties dialog box, click **Start**, and then click **OK**.

Enabling FIPS on the GUI Server

You can configure the GUI Server to use a Federal Information Processing Standard (FIPS) approved cryptographic provider.

About this task

The GUI password encryption algorithms on the GUI Server use FIPS approved cryptographic providers regardless of whether FIPS is enabled for the entire GUI

Server. However, enabling FIPS on the GUI Server ensures that the encryption used to support SSL communications, as well as Single Sign On, uses a FIPS-approved cryptographic provider.

Follow these steps to enable FIPS 140–2 for the GUI Server.

Procedure

1. Configure the GUI Server to use FIPS.
   a. Log in to the GUI Server using the Tivoli Integrated Portal.
   b. In the navigation pane, click **Settings** > **Websphere Administrative Console** and click **Launch Websphere administrative console**.
   c. In the WebSphere Application Server administrative console navigation pane, click **Security** > **SSL certificate and key management**.
   d. Select the **Use the United States Federal Information Processing Standard (FIPS) algorithms** option and click **Apply**. This option makes IBMJSSE2 and IBMJCEFIPS the active providers.
2. Configure the GUI Server to use FIPS algorithms for Java clients that must access enterprise beans:
   a. Open the `tip_home_dir/profiles/TIPProfile/properties/ssl.client.props` file in a text editor.
   b. Change the `com.ibm.security.useFIPS` property value from `false` to `true`.
3. Configure the GUI Server to use FIPS algorithms for SOAP-based administrative clients that must access enterprise beans:
   a. Open the `tip_home_dir/profiles/TIPProfile/properties/soap.client.props` file in a text editor.
   b. Add the following line, `com.ibm.ssl.contextProvider=IBMJSSEFIPS`.
4. Configure `java.security` to enable IBMJCEFIPS:
   a. Open the `tip_home_dir/java/jre/lib/security/java.security` file in a text editor.
   b. Insert the IBMJCEFIPS provider (`com.ibm.crypto.fips.provider.IBMJCEFIPS`) before the IBMJCE provider, and also renumber the other providers in the provider list. The IBMJCEFIPS provider must be in the `java.security` file provider list. See the example at the end of this topic.
5. Enable your browser to use Transport Layer Security (TLS) 1.0:
   a. Microsoft Internet Explorer: Start Internet Explorer and click **Tools > Internet Options**. On the **Advanced** tab, select the **Use TLS 1.0** option.
   b. Firefox: TLS 1.0 is enabled by default.
6. Export Lightweight Third Party Authentication keys so applications that use these LTPA keys can be reconfigured.
   a. In the navigation pane, click **Settings** > **Websphere Admin Console** and click **Launch Websphere Admin Console**.
   b. In the WebSphere Application Server administrative console, select **Security** > **Global security**.
   c. In the Global security page, from the Authentication area, click the **LTPA** link.
   d. Under **Cross-cell single sign-on**, specify a key file and provide a filename and password for the file that will contain the exported LTPA keys.
   e. Click **Export keys**. By default the exported file is saved to `tip_home_dir/profiles/TIPProfile/`

7. Reconfigure any applications that use GUI Server LTPA keys: To reconfigure the Tivoli SSO service with the updated LTPA keys, run this script: `tip_home_dir/profiles/TIPProfile/bin/setAuthnSvcLTPAKeys.jacl`.
   a. Change directory to `tip_home_dir/profiles/TIPProfile/bin/`
   b. If the GUI Server is not running, start it using the following command:
      - Windows,`startServer.bat server1`
      - Unix, or Linux,`startServer.sh server1`
   c. Run the following command:

      `wsadmin -username` *`tipadmin`* `-password` *`tipadmin_password`* `-f setAuthnSvcLTPAKeys.jacl` *`exported_key_path key_password`*

      Where:

      `exported_key_path` is name and full path to the key file that was exported.

      `key_password` is the password that was used to export the key.
8. For SSO, enable FIPS for any other GUI Server instances, then import the updated LTPA keys from the first server into these servers:
   a. Copy the LTPA key file from step 6 on page 48 to another GUI Server computer.
   b. In the navigation pane, click **Settings** > **Websphere Admin Console** and click **Launch Websphere Admin Console**.
   c. In the WebSphere Application Server administrative console, select **Security** > **Global security**.
   d. In the Global security page, from the Authentication area, click the **LTPA** link.
   e. Under **Cross-cell single sign-on**, provide the filename and password from above for the file that contains the exported LTPA keys.
   f. Click **Import keys**.
9. Run the **`ConfigureCLI`** command:

   Windows, `tip_home_dir\profiles\TIPProfile\bin\tipcli.bat`

   Unix, or Linux, `tip_home_dir/profiles/TIPProfile/bin/tipcli.sh`

Example

The IBM SDK `tip_home_dir/java/jre/lib/security/java.security` file is similar to this example when IBMJCEFIPS is enabled.

```
security.provider.1=com.ibm.crypto.fips.provider.IBMJCEFIPS
security.provider.2=com.ibm.crypto.provider.IBMJCE
security.provider.3=com.ibm.jsse.IBMJSSEProvider
security.provider.4=com.ibm.jsse2.IBMJSSEProvider2
security.provider.5=com.ibm.security.jgss.IBMJGSSProvider
security.provider.6=com.ibm.security.cert.IBMCertPath
security.provider.7=com.ibm.crypto.pkcs11.provider.IBMPKCS11
security.provider.8=com.ibm.security.cmskeystore.CMSProvider
security.provider.9=com.ibm.security.jgss.mech.spnego.IBMSPNEGO
```

Central user registry

As a post-installation task you can configure a central user registry for user management and authentication. You can configure an LDAP server or Tivoli Netcool/OMNIbus ObjectServer registry (or both).

Note: When you add a new user, you should check that the user ID you specify does not already exist in any of the user repositories to avoid difficulties when the new user attempts to log in.

In a network environment that includes a user registry on an LDAP server or Tivoli Netcool/OMNIbus ObjectServer, you can configure *Tivoli Integrated Portal* to use either or both types. In fact, these functions require a central user registry:

- Load balancing, which requires that each *Tivoli Integrated Portal* server instance in the cluster use the same central user repository, whether that be anLDAP server or an ObjectServer.
- Single sign-on, which authenticates users at the central repository during login and whenever they launch into other authorized Tivoli applications.

Before configuring a central user registry, be sure that the user registry or registries that you plan to identify are started and can be accessed from the computer where you have installed the *Tivoli Integrated Portal*.

Attention: When *Tivoli Integrated Portal* is configured with multiple central user repositories, you cannot login if one remote user repository becomes inaccessible from *Tivoli Integrated Portal*, even if your user ID exists in one of the other repositories. If you need access is this situation, you have to run WebSphere Application Server commands to allow access when all repositories are available, or the federated repositories will not function properly. For more information, refer to the following links:

- http://www-01.ibm.com/support/docview.wss?uid=swg1PK78677
- http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.web20fep.multiplatform.doc/info/ae/ae/rxml_atidmgrrealmconfig.html

Adding an external LDAP repository

After installation, you can add an IBM Tivoli Directory Server or Active Directory Microsoft Active Directory Server as an LDAP repository for *Tivoli Integrated Portal*.

About this task

To add a new LDAP repository:

Procedure

1. Log in to the *Tivoli Integrated Portal*.
2. In the navigation pane, click **Settings** > **Websphere Admin Console** and click **Launch Websphere Admin Console**.
3. In the WebSphere Application Server administrative console, select **Security** > **Global security**.
4. From the **Available realm definitions** list, select **Federated repositories** and click **Configure**.
5. In the Related Items area, click the **Manage repositories** link and then click **Add** to add a new LDAP repository.
6. In the **Repository identifier** field, provide a unique identifier for the repository. The identifier uniquely identifies the repository within the cell, for example, `LDAP1`.
7. From the **Directory type** list, select the type of LDAP server. The type of LDAP server determines the default filters that are used by WebSphere Application Server.

Note: IBM Tivoli Directory Server users can choose either IBM Tivoli Directory Server or SecureWay as the directory type. For better performance, use the IBM Tivoli Directory Server directory type.

8. In the **Primary host name** field, enter the fully qualified host name of the primary LDAP server. The primary host name and the distinguished name must contain no spaces. You can enter either the IP address or the domain name system (DNS) name.
9. In the **Port** field, enter the server port of the LDAP directory.

   The host name and the port number represent the realm for this LDAP server in a mixed version nodes cell. If servers in different cells are communicating with each other using Lightweight Third Party Authentication (LTPA) tokens, these realms must match exactly in all the cells.

   Note:

   The default port value is `389`, which is not a Secure Sockets Layer (SSL) connection port. Use port `636` for a Secure Sockets Layer (SSL) connection. For some LDAP servers, you can specify a different port. If you do not know the port to use, contact your LDAP server administrator.
10. Optional: In the **Bind distinguished name** and **Bind password** fields, enter the bind distinguished name (DN) (for example, `cn=root`) and password.

    Note: The bind DN is required for write operations or to obtain user and group information if anonymous binds are not possible on the LDAP server. In most cases, a bind DN and bind password are needed, except when an anonymous bind can satisfy all of the required functions. Therefore, if the LDAP server is set up to use anonymous binds, leave these fields blank.
11. Optional: In the **Login properties** field, enter the property names used to log into the WebSphere Application Server. This field takes multiple login properties, delimited by a semicolon (`;`). For example, `cn`.
12. Optional: From the **Certificate mapping** list, select your preferred certificate map mode. You can use the X.590 certificates for user authentication when LDAP is selected as the repository.

    Note: The **Certificate mapping** field is used to indicate whether to map the X.509 certificates into an LDAP directory user by `EXACT_DN` or `CERTIFICATE_FILTER`. If you select `EXACT_DN`, the DN in the certificate must match the user entry in the LDAP server, including case and spaces.
13. Click **OK**.
14. In the Messages area at the top of the Global security page, click the **Save** link and log out of the WebSphere Application Server console.

What to do next

Configure the Tivoli Integrated Portal Server to communicate with an external LDAP repository.

Configuring an external LDAP repository

You can configure the Tivoli Integrated Portal Server to communicate with an external LDAP repository.

About this task

In a load balanced environment, all Tivoli Integrated Portal Server instances must be configured separately for the LDAP server. To configure an application server to communicate with an external LDAP repository:

Procedure

1. Log in to *Tivoli Integrated Portal*.
2. In the navigation pane, click **Settings** > **Websphere Administrative Console** and click **Launch Websphere Administrative Console**.
3. In the WebSphere Application Server administrative console, select **Security** > **Global security**.
4. From the **Available realm definitions** list, select **Federated repositories** and click **Configure**.
5. To add an entry to the base realm:
   a. Click **Add Base entry to Realm**.
   b. Enter the distinguished name (DN) of a base entry that uniquely identifies this set of entries in the realm. This base entry must uniquely identify the external repository in the realm.

      Note: If multiple repositories are included in the realm, use the DN field to define an additional distinguished name that uniquely identifies this set of entries within the realm. For example, repositories LDAP1 and LDAP2 might both use `o=ibm,c=us` as the base entry in the repository. So `o=ibm,c=us` is used for LDAP1 and `o=ibm2,c=us` for LDAP2. The specified DN in this field maps to the LDAP DN of the base entry within the repository (such as `o=ibm,c=us b`). The base entry indicates the starting point for searches in this LDAP directory server (such as `o=ibm,c=us c`).
   c. Click **OK**.
   d. In the Messages area at the top of the Global security page, click the **Save** link and log out of the WebSphere Application Server console.
6. In the WebSphere Application Server administrative console, select **Security** > **Global security**.
7. From the **Available realm definitions** list, select **Federated repositories** and click **Set as current** to mark the federated repository as the current realm.
8. Stop and restart the Tivoli Integrated Portal Server:
   a. In the *tip_home_dir*`/profiles/TIPProfile/bin` directory, depending on your operating system, enter one of the following commands:
      - `stopServer.bat server1`
      - `stopServer.sh server1`

      Note: On UNIX and Linux systems, you are prompted to provide an administrator username and password.
   b. In the *tip_home_dir*`/profiles/TIPProfile/bin` directory, depending on your operating system, enter one of the following commands:
      - `startServer.bat server1`
      - `startServer.sh server1`
9. Verify that the federated repository is correctly configured:
   a. In the portal navigation pane, click **Users and Groups** > **Manage Users**.
   b. Select **User ID** from the **Search by** list.

c. Click **Search** to search for users in the federated repository.
d. Confirm that the list includes users from both the LDAP repository and the local file registry.

On the Tivoli Integrated Portal Server, LDAP users are queried only by the `userid` attribute. When users are imported into LDAP using an LDAP Data Interchange Format (LDIF) file, an auxiliary class of type `eperson` and an `uid` attribute is added to the LDAP user ID. Note that this is to be done only if you want to search the LDAP repository using VMM from the server.

What to do next

To be able to create or manage users in the portal that are defined in your LDAP repository, in the WebSphere Application Server administrative console, you must specify the supported entity types.

Managing LDAP users in the console

To create or manage users in the portal that are defined in your LDAP repository, in the WebSphere Application Server administrative console specify the supported entity types.

About this task

To create or manage LDAP users in the portal:

Procedure

1. Log in to the *Tivoli Integrated Portal*.
2. In the navigation pane, click **Settings** > **Websphere Admin Console** and click **Launch Websphere Admin Console**.
3. In the WebSphere Application Server administrative console, select **Security** > **Global security**.
4. From the **Available realm definitions** list, select **Federated repositories** and click **Configure**.
5. In the Additional Properties area, click **Supported entity types**, to view a list of predefined entity types.
6. Click the name of a predefined entity type to change its configuration.
7. In the **Base entry for the default parent** field, provide the distinguished name of a base entry in the repository. This entry determines the default location in the repository where entities of this type are placed on write operations by user and group management.
8. In the **Relative Distinguished Name properties** field, provide the relative distinguished name (RDN) properties for the specified entity type.

   Possible values are `cn` for **Group**, `uid` or `cn` for **PersonAccount**, and `o`, `ou`, `dc`, and `cn` for **OrgContainer**.

   Delimit multiple properties for the **OrgContainer** entity with a semicolon (**;**).
9. Click **OK** to return to the Supported entity types page.
10. In the Messages area at the top of the Global security page, click the **Save** link and log out of the WebSphere Application Server console.
11. For the changes to take effect, stop, and restart the Tivoli Integrated Portal Server. In a load balanced environment, you must stop and restart each Tivoli Integrated Portal Server instance.
12. Stop and restart the Tivoli Integrated Portal Server:

a. In the *tip_home_dir*/profiles/TIPProfile/bin directory, depending on your operating system, enter one of the following commands:
   - stopServer.bat server1
   - stopServer.sh server1

   Note: On UNIX and Linux systems, you are prompted to provide an administrator username and password.
b. In the *tip_home_dir*/profiles/TIPProfile/bin directory, depending on your operating system, enter one of the following commands:
   - startServer.bat server1
   - startServer.sh server1

Results

You can now manage your LDAP repository users in the portal through the **Users and Groups** > **Manage Users** menu items.

Note: When you add a new user, you should check that the user ID you specify does not already exist in any of the user repositories to avoid difficulties when the new user attempts to log in.

Restriction: You cannot currently update user IDs through the **Users and Groups** > **Manage Users** portlet that have been created in Microsoft Active Directory repositories.

Configuring an SSL connection to an LDAP server

If your implementation of *Tivoli Integrated Portal* uses an external LDAP-based user repository, such as Microsoft Active Directory, you can configure it to communicate over a secure SSL channel.

Before you begin

This task assumes that you have already an existing connection to an LDAP server set up.

Your LDAP server (for example, an IBM Tivoli Directory Server Version 6 or an Microsoft Active Directory server), must be configured to accept SSL connections and be running on secured port number (636). Refer to your LDAP server documentation if you need to create a signer certificate, which as part of this task, must be imported from your LDAP server into the trust store of the Tivoli Integrated Portal Server.

About this task

Follow these instructions to configure the Tivoli Integrated Portal Server to communicate over a secure (SSL) channel with an external LDAP repository. All application server instances must be configured for the LDAP server.

Procedure

1. Log in to the portal.
2. Follow these steps to import your LDAP server's signer certificate into the application server trust store.

a. In the navigation pane, click **Settings** > **Websphere Admin Console** and click **Launch Websphere Admin Console**.
b. In the WebSphere Application Server administrative console navigation pane, click **Security** > **SSL certificate and key management**.
c. In the Related Items area, click the **Key stores and certificates** link and in the table click the **NodeDefaultTrustStore** link.
d. In the Additional Properties area, click the **Signer certificates** link and click the**Retrieve from port** button.
e. In the relevant fields, provide hostname, port (normally 636 for SSL connections), SSL configuration details, as well as the alias of the certificate for your LDAP server and click the **Retrieve signer information** button and then click **OK**.

3. Follow these steps to enable SSL communications to your LDAP server:
   a. In the navigation pane, click **Security** > **Secure administration, applications, and infrastructure**.
   b. Select **Federated repositories** from the **Available realm definitions** drop down list and click **Configure**.
   c. Select your LDAP server from the **Repository** drop down list.
   d. Enable the **Require SSL communications** check box and the select the **Centrally managed** option.
   e. Click **OK**.
4. For the changes to take effect, save, stop, and restart all Tivoli Integrated Portal Server instances.

What to do next

If you intend to enable single sign-on (SSO) so that users can log in once and then traverse to other applications without having to re-authenticate, configure SSO.

Configuring an SSL connection to the ObjectServer

For environments that include a Tivoli Netcool/OMNIbus ObjectServer user registry, you need to set up encrypted communications on the Tivoli Integrated Portal Server.

About this task

Follow these steps to establish a secure channel for communications between the Tivoli Integrated Portal Server and the ObjectServer.

Procedure

1. Retrieve the ObjectServer certificate information, as follows:
   a. In the navigation pane, click **Settings** > **Websphere Admin Console** and click **Launch Websphere Admin Console**.
   b. In the WebSphere Application Server administrative console navigation pane, click **Security** > **SSL certificate and key management**.
   c. On the SSL certificate and key management page, click **Key stores and certificates** and on the page that is displayed, click **NodeDefaultTrustStore**.
   d. On the NodeDefaultTrustStore page, click **Signer certificates** and on the page that is displayed, click **Retrieve from port**.
   e. In the relevant fields, enter **Host**, **Port**, and **Alias** values for the ObjectServer and click **Retrieve signer information**.

The signer information is retrieved and stored. For your reference, when the signer information has been retrieved, the following details are displayed:

Serial number
Specifies the certificate serial number that is generated by the issuer of the certificate.

Issued to
Specifies the distinguished name of the entity to which the certificate was issued.

Issued by
Specifies the distinguished name of the entity that issued the certificate. This name is the same as the issued-to distinguished name when the signer certificate is self-signed.

Fingerprint (SHA digest)
Specifies the Secure Hash Algorithm (SHA hash) of the certificate, which can be used to verify the certificate's hash at another location, such as the client side of a connection.

Validity period
Specifies the expiration date of the retrieved signer certificate for validation purposes.

2. Open *tip_home_dir*/profiles/TIPProfile/etc/com.sybase.jdbc3.SybDriver.props in a text editor and change these parameters:
   a. Enable SSL for ObjectServer primary host: `USESSLPRIMARY=TRUE`
   b. Enable SSL for ObjectServer backup host: `USESSLBACKUP=TRUE`
3. Stop and restart the Tivoli Integrated Portal Server:
   a. In the *tip_home_dir*/profiles/TIPProfile/bin directory, depending on your operating system, enter one of the following commands:
      - `stopServer.bat server1`
      - `stopServer.sh server1`

      Note: On UNIX and Linux systems, you are prompted to provide an administrator username and password.
   b. In the *tip_home_dir*/profiles/TIPProfile/bin directory, depending on your operating system, enter one of the following commands:
      - `startServer.bat server1`
      - `startServer.sh server1`

Configuring LDAP or Active Directory

After the installation, you can configure an LDAP server through the administrative console.

Before you begin

If you want all LDAP communications to be encrypted, you can specify SSL communications. If so, be sure to import the LDAP signer's certificate into the trust store of the embedded version of WebSphere Application Server before starting this task.

About this task

Follow these instructions to configure the embedded version of WebSphere Application Server to communicate over a secure (SSL) channel with an external repository such as Microsoft Active Directory. All application server instances must be configured for the LDAP server.

Procedure

1. Log in to the administrative console.
   `http://host:adminport/ibm/console`

   where *hostname* is the name of the system where the application server is running and *port* is the admin port. The default port is `9060`.

   The application server console prompts you for login information. The default administration username is `wasadmin` and the default password is `netcool`.
2. If you need to add a new LDAP repository, complete the following steps from the administrative console.
   a. In the navigation pane, select **Security** > **Secure administration, applications, and infrastructure.**
   b. From the available realm definitions, select **Federated repositories** and then click **Configure**.
   c. Under Related Items click **Manage repositories**. To add a new LDAP Repository, click **Add**.
   d. Enter the LDAP security setting information. The primary host name and the distinguished name must contain no spaces.
   e. If all LDAP communications must be encrypted, select **Require SSL communications**. If users authenticate to a Microsoft Active Directory Server, select this to enable password changes and creating users from the administrative console.
   f. Select **Centrally managed**.
   g. Click **OK**.
3. Return to **Secure administration, applications, and infrastructure** > **Federated repositories** and add an entry to the base realm:
   a. Click **Add Base entry to Realm**.
   b. Enter the distinguished name (DN) of a base entry that uniquely identifies this set of entries in the realm. This base entry must uniquely identify the external repository in the realm.
   c. Click **OK**.

   If multiple repositories are included in the realm, use the DN field to define an additional distinguished name that uniquely identifies this set of entries within the realm. For example, repositories LDAP1 and LDAP2 might both use *o=ibm, c=us* as the base entry in the repository. So *o=ibm, c=us* is used for LDAP1 and *o=ibm2, c=us* for LDAP2. The specified DN in this field maps to the LDAP DN of the base entry within the repository (for example, *o=ibm, c=us b*). The base entry indicates the starting point for searches in this LDAP directory server (such as *o=ibm, c=us c*).

   Note: When you use the federated repositories functionality, all the configured repositories, which you specify as part of the federated repository configuration, become active. The user ID, and the distinguished name (DN) for an LDAP repository, must be unique in multiple user repositories configured under the same federated repository configuration.

For example, there might be three different repositories that are configured for the federated repositories configuration: Repository A, Repository B, and Repository C. When user1 logs in, the federated repository adapter searches each of the repositories for all the occurrences of that user. If multiple instances of that user are found in the combined repositories, an error message displays.

For more information about federated repositories see *Federated Repositories* in *Network Deployment, Distributed platforms and Windows Version 6.1* in the WebSphere Application Server documentation information center http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/welcome.html.

4. Restart the server to enable the configuration.
5. To verify that the federated repository is properly configured, complete the following steps:
   a. In the navigation pane, click **Users and Groups** > **Manage Users**.
   b. From the **Search by** list, select **User ID**.
   c. Click **Search** to search Users in federated repository. This list displays users from both LDAP and the local file registry.
6. If you want to create a user in LDAP, click **Users and Groups** > **Manage Users**, and then click **Create** and continue as for the previous step: Enter user ID, first name, last name, e-mail, and password.
7. For the changes to take effect, save, stop, and restart the embedded version of the WebSphere Application Server.

What to do next

In order for Netcool/Impact to start using the LDAP server for authentication and authorization, you need to configure it in the embedded version of WebSphere Application Server so that the roles exposed in the application can be mapped to groups and users stored in LDAP.

Single sign-on

The single sign-on (SSO) capability in Tivoli products means that you can log on to one Tivoli application and then launch to other Tivoli Web-based or Web-enabled applications without having to re-enter your user credentials.

The repository for the user IDs can be the Tivoli Netcool/OMNIbus ObjectServer or a Lightweight Directory Access Protocol (LDAP) registry. A user logs on to one of the participating applications, at which time their credentials are authenticated at a central repository. With the credentials authenticated to a central location, the user can then launch from one application to another to view related data or perform actions. Single sign-on can be achieved between application servers deployed to Tivoli Integrated Portal servers on multiple machines.

Single sign-on capabilities require that the participating products use Lightweight Third Party Authentication (LTPA) as the authentication mechanism. When SSO is enabled, a cookie is created containing the LTPA token and inserted into the HTTP response. When the user accesses other Web resources (portlets) in any other application server process in the same Domain Name Service (DNS) domain, the cookie is sent with the request. The LTPA token is then extracted from the cookie and validated. If the request is between different cells of application servers, you must share the LTPA keys and the user registry between the cells for SSO to work. The realm names on each system in the SSO domain are case sensitive and must

match exactly. See Managing LTPA keys from multiple WebSphere Application Server cells on the WebSphere Application Server Information Center.

Prerequisites to configuring single sign-on

Configuring single sign-on (SSO) is a prerequisite to integrating products that are deployed on multiple servers. Before you configure single sign-on support and a federated repository, check the following requirements:

Before you begin

- All instances must point to the central user registry, such as a Lightweight Directory Access Protocol server (LDAP).
- The Tivoli Integrated Portal installations must be in the same domain. For example, **xxx.yyy.company.com** and **zzz.yyy.company.com**
- Configure either the ObjectServer or the LDAP server.
  - Configure the ObjectServer during installation of Netcool/Impact, by using the advanced installation option on the launchpad.
  - Configure a federated repository, for example LDAP, post installation. See "Configuring LDAP or Active Directory" on page 56. During the configuration, leave the default value for REALM to ensure that both LDAP instances operate in the same realm.
  - Restart the servers and check that the user accounts are stored in the LDAP or the ObjectServer.

Verifying that user accounts are stored in the LDAP or Netcool/ImpactObjectServer

Restart the servers, verify that the user accounts are stored in the LDAP or the Netcool/Impact ObjectServer before you configure single sign-on.

Procedure

1. Log in to the Tivoli Integrated Portal.
2. Select **Users and Groups** > **Manage Users**.
3. Click **Search** to view the list of user accounts. The values in the **Unique Names** column are displayed in the following format:
   - For the ObjectServer: `uid=<name>,o=netcoolObjectServerRepository`
   - For the LDAP server: `cn=TIP Administrator,ou=Administrator,ou=TIP,DC=tivoli`
   - For the local file system: `uid=tipadmin,o=defaultWIMFileBasedRealm`

   Restriction: Tivoli Monitoring single sign-on (SSO) support is only available with Tivoli Monitoring Version 6.2 Fix Pack 1 or higher.

Configuring LTPA and exchange keys for single sign-on

You can use the Lightweight Third Party Authentication (LTPA) mechanism to generate a trusted keystore file. Exchanging keys between Tivoli Integrated Portal servers establishes a trusted relationship between them.

About this task

How to configure LTPA and exchange keys to enable single sign-on between two remote Tivoli Integrated Portal installations. In this setup, the installations are called *TIP-1* and *TIP-2*. *TIP-2* is the instance where the operator view files are stored.

Procedure

1. Log in to the Tivoli Integrated Portal.
2. In the navigation pane, click **Settings** > **Websphere Administrative Console** and click **Launch Websphere administrative console**.
3. In the WebSphere Application Server administrative console navigation pane, click **Security** > **Global security**.
4. In the **Authentication** area, expand **Web security** and click **Single sign-on**.
5. Click the **Enabled** option if SSO is disabled.
6. In the **General Properties** section, complete the following steps:
   a. If all of the requests are expected to use HTTPS click **Requires SSL**.
   b. Enter the fully qualified domain names in the Domain name field where SSO is effective. If the domain name is not fully qualified, the Tivoli Integrated Portal Server does not set a domain name value for the **LtpaToken** cookie. Single sign-on is valid only for the server that created the cookie. For SSO to work across Tivoli applications, their application servers must be installed in same domain and use the same domain name.
   c. Optional: To support SSO connections in WebSphere Application Server version 5.1.1 or later to interoperate with previous versions of the application server enable the **Interoperability Mode** option.
   d. Optional: To add information during the login at a specific Tivoli Enterprise Portal Server to propagate to other application server instances enable the **Web inbound security attribute propagation** option.
7. Click **Apply**. Click **OK** and return to the **Global Security** page.

Configuring the LTPA parameters to export the key

In the **Global Security** page, configure the LTPA Parameters to export the key.

Procedure

1. In the **Authentication** section, select the **LTPA** radio button if it is not already selected.
2. To configure the LTPA parameters, click the **LTPA** link. In the **LTPA timeout** section, the **LTPA timeout for forwarded credentials** field shows how much time is available after logging in to the Tivoli Integrated Portal to forward credentials to another application. The default time interval is `120` minutes. When the time interval expires, you must log in to the Tivoli Integrated Portal again to be able to share credentials.
   a. In the **Password** field, type a password for key encryption or decryption.
   b. In the **Confirm Password** field, type the password again.
   c. In the **Fully qualified key file name**, specify the location on the file system to which you want export and store the encrypted key. Give the file a name that indicates which key is which. Identifying keys is important when you are using two instances of the Tivoli Integrated Portal. For example, the file name for the first instance of the Tivoli Integrated Portal *TIP-1* is `C:\myKeys\TIP-1-key.txt`. Click the **Export keys** button. The key is generated key and exported to the file listed in the file path.
   d. A confirmation message similar to the following example is displayed:

      ```
      The keys were successfully exported to the file C:\myKeys\TIP-1-
      key.txt" to be continued ...
      ```
3. Repeat the steps in the second Tivoli Integrated Portal instance, **TIP-2**. Export the key, then specify the location where you want to import and store the key from. Import the key from the first Tivoli Integrated Portal instance **TIP-1**.

4. Repeat the steps for "Configuring LTPA and exchange keys for single sign-on" on page 59 and steps 1 on page 60 and 2 on page 60 in this section for the second Tivoli Integrated Portal **TIP-2**.

Specifying locations to import and exchange keys

After exporting the LTPA keys, specify the locations to import and store the keys and exchange the keys between the Tivoli Integrated Portal instances.

Procedure

1. In the second Tivoli Integrated Portal instance, change the value for the **Fully qualified key file name**. Specify a location on the file system **TIP-2** to which you want to import and store the key from the other Tivoli Integrated Portal instance **TIP-1**.
2. Manually copy the exported key file in the **TIP-2** file system from there to the file location in **TIP-1**.
3. In the **Password** field, type a password for key encryption or decryption. In the **Confirm Password** field, type the password again.
4. Click the **Import keys** button. A confirmation message is displayed and you are asked to acknowledge the changes.
5. Click **Save** to implement the changes to the master configuration `security.xml` file in `<TIP-HOME>\tipv2\profiles\TIPProfile\config\cells\TIPCell\security.xml` The imported key file is displayed shown in the specified location.
6. Repeat the steps in this section for the other Tivoli Integrated Portal installation, **TIP-1**.
7. Restart both Tivoli Integrated Portal instances **TIP-1** and **TIP-2** to implement the changes.

Results

You have exchanged the file keys and created passwords to encrypt and decrypt the key files. The authentication sharing works in both directions, from **TIP-1** to **TIP-2** and **TIP-2** to **TIP-1**.

Verifying that Single Sign-on is working

Log in to the first instance of the Tivoli Integrated Portal in browser and open an operator view to check single sign-on is working.

Procedure

1. Log in to the installation called **TIP-1**.
2. On the same browser, open a new tab, and point it to the following URL:

   http://<TIP-2HOSTNAME>:16310/opview/displays/NCICLUSTER-AddRWin.html

   You see the displayed operator view file from a security protected web area of **TIP-2** server.
3. If you are using a Mozilla Firefox browser, you can check the cookies, to see whether cookies to support SSO were created. See the browser help for information about how to view cookies. For the domain name, **LtpaToken** and **LtpaToken2** are created to handle the single sign-on.

Chapter 5. Server clustering

In a server cluster, you install a group of related Impact Servers and configure them to operate as a single server instance.

You use this feature to add failover and load-balancing functions to a Netcool/Impact installation.

Server clustering overview

You can install many instances of the Netcool/Impact components and configure them to provide failover capability.

Each component can be configured for resiliency. For example the Name Server can be configured as a redundant pair. This way you avoid a scenario when the Name Server acts as a single point of failure in a deployment.

The Impact Server can be clustered for both resiliency and scaling processing. An Impact Server cluster consists of the primary server, which is responsible for gathering events from various event sources and making it available to all servers in the cluster (itself and all secondary servers). Secondary servers do not connect directly to the event sources using the reader and listener services.

GUI Servers cannot form a cluster, but you can install several GUI Server so that you can always access the Netcool/Impact UI even if one of the GUI Servers goes down.

You start setting up a cluster, by installing separate instances of Netcool/Impact components on each target system. All servers in a cluster must use the same cluster name. Each server in a cluster must have its unique name. You specify the cluster name, and the instance name during the installation.

You configure cluster members in the same way you configure single servers. Most changes made to the server configuration using the GUI are propagated automatically to other cluster members. Manual changes made to the server properties files are not automatically propagated. Configuration for certain services like event processor, self-monitoring and command-line manager are not replicated to the secondary servers. You need to manually configure these services in the secondary server using the command-line service.

You can update the Name Server list after the installation by running `nci_configuration_utility`. For more information, see “Post installation utility” on page 30.

The Impact Server cluster components

An Impact Server cluster consists of the primary server, and secondary servers.

Primary server

The primary server is the cluster member that is responsible for gathering events from various event sources using the reader or listener services and for making events available to the secondary servers for processing, as well as processing events on its own.

The primary server also runs any configured instances of services like OMNIbusEventReader, OMNIbusEventListener, JMSMessageListener, Database Event Reader, and DatabaseEventListener. The first cluster member that registers itself in the primary nameserver becomes the primary server.

Secondary servers

Secondary servers are cluster members that are responsible for retrieving events from the primary server, and for running policies in response to the incoming events. Secondary servers function solely as event processing services in the cluster.

Most services are disabled in the secondary servers. The following services are not disabled:

- Event processor
- Policy logger
- Self-monitoring service
- Command line manager

By default, each secondary server synchronizes its services, data types, data sources, policies, and configuration settings with the primary server before becoming active. Certain configuration like the number of event processing threads, are not synchronized.

The Name Server cluster components

A Name Server cluster consists of a primary Name Server and one or more secondary name servers.

The primary Name Server is the component responsible for communicating in real time with other deployment components like Impact Server and the GUI Server. It stores information about the primary server of the cluster and where it is located in the network.

The secondary name servers are responsible for providing failover functionality for the primary Name Server. They do not communicate with the Netcool applications in real time, except in the case that the primary Name Server fails and a secondary Name Server assumes the role of the primary.

You can update the Name Server list after the installation by running `nci_configuration_utility`. For more information, see “Post installation utility” on page 30.

Setting up a cluster

Use this procedure to set up a Netcool/Impact cluster.

About this task

This procedure uses an example of a typical installation of two Impact Servers, and two GUI servers. Two name servers are configured as a redundant pair. Because the Name Server is a subcomponent of the Impact Server, each Name Server will have the same hostname, or IP address as its corresponding Impact Server. For this configuration two separate systems will be used:

- System 1, with the hostname `primary.com`, will contain the first Impact Server-GUI Server pair. It will also contain the primary Name Server.

- System 2, with the hostname `secondary.com`, will contain the second Impact Server-GUI Server pair. It will also contain the secondary Name Server.

GUI servers cannot form a cluster, but it is good practice to install several GUI servers so that you can always access the Netcool/Impact UI even if one GUI Server goes down.

Restriction: If you are configuring the Tivoli Integrated Portal profile for a cluster of servers, you must use the same user name and password for all the servers in the cluster.

Procedure

1. On the first system install the first Impact Server-GUI Server pair.
   a. In the installation type screen, select the advanced installation type.

      For more information, see "Selecting the type of installation" on page 15.
   b. In the nameserver configuration screen, enter hostname:port pairs of both your name servers, with the primary Name Server at the top of the list.

      For example:

      ```
      primary.com:9080
      secondary.com:9080
      ```

      Port 9080 is the default port that is used to communicate with the application server, but you can use a different port for that purpose. For more information, see "Configuring Impact Server WebSphere profile ports" on page 17. For more information about the nameserver configuration screen, see "Configuring the Name Server" on page 18.
   c. In the Impact Server instance configuration screen, choose the name of the cluster group to which you want to assign the server, and give the Impact Server a unique name within that cluster.

      For example, `NCI1`, and `NCICLUSTER`. For more information, see "Configuring the Impact Server instance" on page 19.

   From the cluster perspective, you can go with the default values for the remaining installation screens, and modify them after the installation, if necessary.

   If you are installing your cluster components in silent mode, you select your installation options by editing the silent installation response file. For more information, see "Silent mode" on page 20.
2. Stop the primary Impact Server after the installation is completed.

   This is to prevent the secondary server from finding the primary server and starting to replicate, during the installation. For more information about stopping the Impact Server, see "Stopping and starting the Impact Server" on page 42.
3. On the second system install the second Impact Server-GUI Server pair.
   a. In the installation type screen, select the advanced installation type.
   b. In the nameserver configuration screen, enter hostname:port pairs of both name servers, and make sure the list, and the order of servers is exactly the same as during the installation of the first Impact Server-Name Server pair.

      You can reuse the `$IMPACT_HOME/etc/nameserver.props` file that was created during the installation of the first Impact Server-Name Server pair.
   c. In the Impact Server instance configuration screen, assign the server to the same cluster group, as the server that you installed first, for example `NCICLUSTER`. and give the server a unique name within that cluster, for example `NCI2`.

For more information, see “Configuring the Impact Server instance” on page 19.

From the cluster perspective, you can go with the default values for the remaining installation screens, and modify them after the installation, if necessary.

4. Stop the secondary Impact Server after the installation is completed.

   This is to prevent the secondary from assuming the role of the primary server.

5. Start the primary Impact Server.

   For more information about starting the Impact Server, see “Stopping and starting the Impact Server” on page 42.

6. Start the secondary Impact Server.

   Important: Make sure that the primary server is up and running, otherwise the secondary server may join the cluster earlier and assume the primary role.

What to do next

- Check the status of the server cluster in the configuration documenter. For more information, see the *Configuration documenter* chapter in the *User Interface Guide*.
- Check the cluster status in the Name Server browser frontend. For more information about checking the cluster status, see “Checking the cluster status in the browser” on page 71.
- Perform runtime analysis of server logs. For more information about runtime analysis, see “Runtime analysis of the server logs” on page 71.
- At any time, if you need to reconfigure the cluster, run the `$IMPACT_HOME/install/nci_configuration_utility` on each of the cluster components. This tool will let you set up a Name Server cluster, or reconfigure the existing one.

Sharing nameservers with TBSM

Although a configuration where a TBSM data server is in the same cluster as an Impact Server is not supported you can configure them to share nameservers.

This shared nameserver configuration is necessary to enable some integrations between TBSM and a remote Impact Server, for example, event enrichment and PassToTBSM.

The TBSM data server only exists in the TBSMCLUSTER, and the only other TBSM data server that can exist in that cluster is the TBSM backup server. You can configure TBSM to share a nameserver cluster, but you cannot add Impact Servers to the TBSMCLUSTER.

In a distributed configuration both the TBSM data server and the Impact Server are running nameservers independently. However, because the TBSM data server is basically an Impact Server with TBSM customizations, the nameserver in it is the same as the Impact Server is running so they both can connect to each other's nameservers. In this configuration, either the TBSM data server nameserver, or the Impact Server nameserver can be the primary, with the other secondary. For this configuration to work, all servers, including the TIP server, must have exactly the same nameservers list configured.

Configuring shared name servers with Tivoli Business Service Manager

You use the nci_configuration_utility to configure Tivoli Business Service Manager and Netcool/Impact with a list of shared name servers.

About this task

For this sample procedure, we assume that the following three servers are installed, each on a dedicated machine, with the following configuration:

- TBSM Data Server, tbsmdataserver.com, with the primary Name Server.
- Impact Server, impactserver.com, with the secondary Name Server.
- Tivoli Integrated Portal Server with the Netcool/Impact UI, and the TBSM UI on it.

We run the nci_configuration_utility on these three servers. In your environment, you must run the utility on all Impact Servers, TBSM data servers, and Tivoli Integrated Portal servers that you have installed and that you want to share name servers. For information about the nci_configuration_utility, see "Post installation utility" on page 30.

Procedure

1. Run the nci_configuration_utility on the machine where the TBSM Data Server is running.

   You can find the utility in the $TBSM_HOME/install directory.

   a. Type the Impact Server host name, and the port name, in the **Format is -> Host_Name:Port_Number** field, and click the **+** icon to add it to the list. The following list of name servers as an example:

      ```
      tbsmdataserver.com:9080
      impactserver.com:9080
      ```

   b. Click **Next**.

   c. In the **Cycle eWAS server** screen, make sure that the **Select to stop and restart the eWAS server** option is selected and click **Next** to restart the TBSM server.

2. Run the nci_configuration_utility on the machine where the Impact Server is running.

   You can find the utility in the $IMPACT_HOME/install directory.

   a. In the **Choose Install Set** screen, select the **Configure Name Servers** option.

   b. Type the TBSM Data Server host name, and the port name, in the **Format is -> Host_Name:Port_Number** field, and click the **+** icon to add it to the list. The Impact Server Name Server shows at the top of the list.

   c. Using the same Name Server list as an example, select impactserver.com:9080 and use the down arrow to move it to the bottom of the list, for example:

      ```
      tbsmdataserver.com:9080
      impactserver.com:9080
      ```

      Attention: 9080 is the default HTTP port used for non-secure communication. If you have HTTPS enabled, the port value will be different.

   d. Click **Next**.

e. In the **Cycle eWAS server** screen, make sure that the **Select to stop and restart the eWAS server** option is selected and click **Next** to restart the Impact Server.

3. Run the `nci_configuration_utility` on the machine where the Tivoli Integrated Portal server is running.

   Assuming that theTivoli Integrated Portal server was installed as part of the TBSM deployment, you can find the utility in the `$IMPACT_HOME/install` directory.

   a. At the moment, the Tivoli Integrated Portal server points only to the Name Server that is running in the TBSM server. You must add the Impact Server Name Server to the list.

   b. Use the up down arrows to get the list of name servers in order, for example:

      ```
      tbsmdataserver.com:9080
      impactserver.com:9080
      ```

   c. Click **Next**.

   d. In the **Cycle eWAS server** screen, make sure that the **Select to stop and restart the eWAS server** option is selected and click **Next** to restart the Tivoli Integrated Portal server.

Results

After you configure shared Name Servers between TBSM and an Impact Server cluster, the Netcool/Impact UI in Tivoli Integrated Portal treats the TBSM cluster as any other Impact Server cluster.

You can select the **TBSMCLUSTER** in the **Cluster** menu in the Netcool/Impact UI and get the full Netcool/Impact configuration that is inside the TBSM server.

The Impact Server clustering process

The Impact Server clustering has the following phases.

Startup

At startup, each server communicates with the Name Server, and, if no other cluster member is currently registered, it registers itself as the primary server.

Event monitoring

During the event monitoring phase, the primary server queries the ObjectServer at intervals for new and updated events, and (optionally) receives notification from the ObjectServer when an event is deleted.

The primary server places the incoming events in an event queue. The event queue will eventually get picked up by the EventProcessor Service that is running on each server in the cluster. Secondary servers do not query the ObjectServer at any time.

Event processing

During the event processing phase, each secondary server queries the primary server for events to process. Similarly, the primary server requests events from its own event queue.

For each retrieved event, the server runs the corresponding policy, dependent on the filter conditions specified in the event reader or listener.

Failover

The secondary servers ping the primary at intervals during runtime. This assures the secondary servers that the primary is active and functioning.

If the primary server fails, the first secondary server to become aware of the failure contacts the nameserver and registers itself as the new primary. When the original primary server is restarted, it becomes another secondary server. If a secondary server fails, there is no impact on the other servers in the cluster.

Failback

Failback is optional in an Impact server cluster, and takes place only if you manually designate a particular server as the primary server.

Shutdown

If the primary server is manually shut down, the first secondary server to become aware of the failure contacts the nameserver and registers itself as the new primary, as in the failover phase. If a secondary server is shut down, there is no impact on the other servers in the cluster.

Startup in an Impact Server cluster

At startup, each server communicates with the Name Server, and, if no other cluster member is currently registered, it registers itself as the primary server.

If a primary server is already registered, it declares itself a secondary server. By default, each secondary server synchronizes its services, data types, data sources, projects, policies, and configuration settings with the primary server before becoming active.

You can configure the secondary server to wait for the primary for some time. If it does not find the primary in the cluster, after this time, it checks the Name Server if any other server registered itself as primary. If it did not, the secondary server starts itself in primary mode. Otherwise, it registers itself in secondary mode.

Important: In earlier releases, the secondary server did not start, if there were any locked files on the primary. This restriction has been removed in this release, but you can still choose to keep this restriction, by setting the `impact.server.ignorelocksduringreplication` property in the server properties file. For more information, see "The Impact Server properties file" on page 74.

Failback in an Impact Server cluster

Failback is optional in an Impact server cluster, and takes place only if you manually designate a particular server as the primary server.

Failback takes place after failover when an original secondary server is behaving as primary. When the original primary starts again, it first assumes a secondary role, and replicates all the configuration. It then assumes the primary role, if it is a preferred primary. If it is not a preferred primary, it starts as, and remains a secondary server.

Important: When all servers are down, start the server which is the preferred primary first.

The Name Server clustering process overview

Nameserver clustering process has the following phases.

Important: For information about updating the Name Server list by running `nci_configuration_utility` see "Post installation utility" on page 30.

Startup

At startup, each nameserver cluster member reads the cluster member list and its position on that list from its `web.xml` file.

The nameserver stores this information internally and uses it to determine its behavior during runtime.

Runtime

During runtime, the cluster member queries the other cluster members at intervals to determine their status and to synchronize nameserver data between them.

Command replication

When you start and stop Netcool/Impact and other components, they issue registration and un-registration commands to the nameserver cluster members.

Failure and recovery

When a cluster member fails, the other members detect the failure during normal runtime operation.

Shutdown

When you shut down a cluster member, it determines whether to save its nameserver information to disk by reading the `PERSIST_ENABLED` property in the `web.xml` file.

If `PERSIST_ENABLED` is set to true, it saves its nameserver information in the directory specified by the `PERSIST_PATH` property. This information is then read by the cluster member at startup.

Command replication in a Name Server cluster

When you start and stop Netcool/Impact and other components, they issue registration and un-registration commands to the nameserver cluster members.

When a cluster member receives such a command, it performs one of the following actions:

- If the member is the primary nameserver instance, it will execute the command and then publish the command to every other nameserver in the cluster. The other nameservers then execute the command and update their registration information.
- If the member is not the primary nameserver instance, it forwards the command to the primary without executing it. The primary then executes the command as described above.

Failure and recovery in a Name Server cluster

When a cluster member fails, the other members detect the failure during normal runtime operation.

After a failure has been detected, the other members update their clustering information and do not attempt to synchronize data with the failed member. When a cluster member recovers, the other members detect the recovery and notify it that it must resynchronize its nameserver data. Failback for a nameserver cluster is enabled by design, and it cannot be disabled.

Runtime analysis of the server logs

If you want to get more verbose logging to perform runtime analysis, edit the `$IMPACT_PROFILE/properties/log4j.properties` file.

Procedure

Change the level from INFO to DEBUG to get verbose logs:

```
log4j.category.com.micromuse=DEBUG,NETCOOL
```

By default, this property is set to `log4j.category.com.micromuse=INFO,NETCOOL`.

Checking the cluster status in the browser

You can view the status of your Name Server instances in a Web browser.

The address of the page where you can view the currently defined Name Server instances and whether they are running or stopped is `http://`*host*`:`*port*`/nameserver/services`, where *host* is the host name of the system where the Name Server is installed and *port* is its port. For example:

```
http://primary.com:9080/nameserver/services
```

Here is an example, of what you can see when you visit this page:

```
Netcool Nameserver is running.

Current cluster state table at this location:

RPL# | SELF | STATUS | URL
-----+------+--------+-----------------------------------------------
   0 | **** |     UP | http://primary.com:9080/nameserver/services
   1 |      |     UP | http://secondary.com:9080/nameserver/services

LAST 100 COMMANDS RECEIVED:


_____________________________________________________________________________
RECORD:279  RECV:1305729356113  TIME:0
 ACCS=api ACTN=Xgetdesat PROD=Impact SERV=NCICLUSTER DESI=0 SHOWSYS=false
 SUPRESS=true TERSE=false
_____________________________________________________________________________
RECORD:278  RECV:1305729343972  TIME:0
 ID=form0000000 ACCS=ifc ACTN=Xstatserv PROD=Impact SERV=NCICLUSTER
 BIND=secondary.com:1993:NCI2 METH=ROB DESI=-1 SHOWSYS=false SUPRESS=true
 TERSE=false
_____________________________________________________________________________
RECORD:277  RECV:1305729333722  TIME:0
 ID=form0000000 ACCS=ifc ACTN=Xstatbind PROD=Impact SERV=NCICLUSTER
 BIND=secondary.com:1993:NCI2 METH=ROB DESI=-1 SHOWSYS=false SUPRESS=true
 TERSE=false
_____________________________________________________________________________
```

You can also view here the last operations performed by the Name Server. By default, the number of operations is 100, but you can change the default number by modifying the `HISTORY_LOOPSIZE` property for the Name Server.

The nameserver.props configuration file

The properties in the $IMPACT_HOME/etc/nameserver.props file determine how the Impact Server connects to the Name Server cluster.

Table 7. Connection properties between the Impact Server and the Name Server cluster

Property	Description
impact.nameserver.count	Number of Name Server instances in the cluster. If you are not running a Name Server cluster, the value of this property is 1.
impact.nameserver.#.host	Host name of the Name Server instance, where # is an index value that identifies the Name Server instance.
impact.nameserver.#.port	The HTTP port that is used by the application server, where # is an index value that identifies the Name Server instance. It is 9080 by default. For HTTPS the default port is 9081.
impact.nameserver.#.location	The URL of the application server on which the Name Server is installed, where # is an index value that identifies the Name Server instance. Default is /nameserver/services.
impact.nameserver.ssl_enabled	Specifies that the server connects to the Name Server using SSL. Value can be true or false. Default is false.
impact.nameserver.netcall_timeout	Number of seconds that the server waits on calls to the Name Server before timing out.

The web.xml configuration file

The Name Server configuration properties are stored in web.xml files.

- $IMPACT_HOME/nameserver/stage/WEB-INF/web.xml
- $TIP_HOME/profiles/ImpactProfile/installedApps/ImpactCell/nameserver.ear/nameserver.war/WEB-INF/web.xml
- $TIP_HOME/profiles/ImpactProfile/config/cells/ImpactCell/applications/nameserver.ear/deployments/nameserver/nameserver.war/WEB-INF/web.xml

The installer automatically sets the minimum required configuration properties for the Name Server, and for most uses you do not need to change them after the installation. This table shows properties of the Name Server in a cluster:

Table 8. Clustering properties of the Name Server

Property	Description
REPLICANT.COUNT	Number of Name Server in the cluster. Default is 1, for a single, non-clustered Name Server.
REPLICANT.#.HOST	Host name of the Name Server instance, where # is an index value that identifies the Name Server instance. You must assign index values to each cluster member in a sequence starting from zero. For example, for the first member in a cluster, this property must be named REPLICANT.0.HOST. For the second member, the property must be named REPLICANT.1.HOST. Use a routable IP or hostname as value of this property, do not use the localhost IP, or name.

Table 8. Clustering properties of the Name Server (continued)

Property	Description
REPLICANT.#.PORT	Port used, where # is an index value that identifies the Name Server instance. This is the HTTP port used by the embedded version of WebSphere Application Server, which has 9080 by default. For HTTPS the default port is 9081.
REPLICANT.#.LOCATION	URL location on servlet container where Name Server is installed, where # is an index value that identifies the Name Server instance. This is the directory you specified when you installed the Name Server. Default is /nameserver/services.
SELFINDEX	An index value that uniquely identifies this Name Server within a cluster. Specify index values as a series, starting with zero. This value should be unique for each Name Server in a cluster.
REPLICANT_PING_INTERVAL	Interval in milliseconds at which the Name Server contacts other cluster members to determine whether they are still running.
REPLICANT_PING_TIMEOUT	Number of milliseconds after sending a query before the Name Server determines that another cluster member is no longer running.
CHECKSUM_IDLE_THRESHOLD	Minimum number of milliseconds that must pass since the last Name Server command was received (excluding ping commands and any read-only commands) before the Name Server can contact other cluster members to determine whether they are still running.
PERSIST_ENABLED	Specifies whether the Name Server must cache its state information to disk at shutdown and read cached information at startup. Can be true or false. Typically, you set this property to true, which is the default.
PERSIST_PATH	Specifies the directory where state information is cached. Default is $IMPACT_HOME/nameserver/nspersist/webapps/nameserver.

Here are other configuration properties for the Name Server:

Table 9. Basic configuration properties for the Name Server

Property	Description
SSL_ENABLED	Specifies whether the Name Server uses the Secure Socket Layer protocol to communicate with Netcool applications. Value can be true or false. Default is false. For more information on configuring the Name Server to use SSL, see "Setting up SSL communication" on page 113.
HISTORY_LOOPSIZE	Number of commands stored in the Name Server command history. Default is 100. For more information, see "Checking the cluster status in the browser" on page 71.
NSSERVICES_NETCALL_TIMEOUT	Number of seconds the application waits after making a call to the Name Server before timing out. Default is 10.
NSSERVICES_LOCK_TIMEOUT	Number of seconds that the Name Server maintains the lock on registry data after a command to write data has been received from a Netcool application. If the Netcool application does not release the lock within the timeout period, the Name Server automatically releases it. Default is 60.

Table 9. Basic configuration properties for the Name Server (continued)

Property	Description
NSSERVICES_CHECK_TIMEOUT	Interval in seconds at which the Name Server checks registry data for expired locks. Default is 31.
ENABLE_GENERAL_LOGGING	Specifies whether logging of Name Server status and event messages is enabled. Value can be true or false. Default is false.
ENABLE_REQUEST_LOGGING	Specifies whether logging of non-ping requests received by the Name Server is enabled. Value can be true or false. Default is false.
ENABLE_PING_LOGGING	Specifies whether logging of ping requests received by the Name Server is enabled. Value can be true or false. Default is false.

The Impact Server properties file

Various cluster configuration related properties of the Impact Server are located in its properties file.

The server properties file is located in $IMPACT_HOME/etc/*servername*_server.props, where *servername* is the name of the server instance. This table explains the server clustering properties:

Table 10. Cluster configuration properties

Name	Description
impact.cluster.name	Name of the server cluster. The default value is NCICLUSTER.
impact.cluster.pinginterval	Interval in milliseconds at which to ping other cluster members. The default value is 6000.
impact.cluster.pingtimeout	Time in milliseconds to wait before retrying ping. The default value is 6000.
impact.cluster.repingcount	Number of times to retry ping if the first fails. The default value is 3.
impact.cluster.resyncbeforestandby	Specifies whether the server synchronizes data with the primary at startup before becoming active. The default value is true.
impact.replication.receiveupdatesfor.orgnodes	Specifies whether to synchronize the data item cache. The default value is true.
impact.replication.receiveupdatesfor.hibernations	Specifies whether to synchronize hibernations. The default value is true.
impact.replication.receiveupdatesfor.servicestates	Specifies whether to synchronize service states. The default value is true.

Table 10. Cluster configuration properties (continued)

Name	Description
impact.replication.receiveupdatesfor.types	Specifies whether to synchronize data types. The default value is true.
impact.replication.receiveupdatesfor.policies	Specifies whether to synchronize policies. The default value is true.
impact.replication.statusinterval	Interval at which servers synchronize data. The default value is 0.
impact.server.ignorelocksduringreplication	To stop the secondary from starting, if there are any files locked on the primary, set this property to false. From now, you will get a list of the locked files, that you must unlock for the secondary to start.
impact.server.preferredprimary	Use this property to configure failback. In a server that you want to be a preferred primary in the cluster, set this property to true. Disabled by default. Use this property in only one server in the cluster. Set the property before starting the server.

Note: If you add customized properties to the *servername*_server.props file, they will not get exported if you use the nci_export tool, and imported using the nci_import tool. Therefore you must replicate these properties manually to other *servername*_server.props files.

Configuring the secondary server to wait for the primary server

You can configure the secondary server to wait for the primary server at startup for some time before it assumes the role of the primary server.

Procedure

1. Open the $IMPACT_HOME/etc/<*servername*>_server.props file of the secondary server.
   a. Required: To enable the feature use this property:
      impact.server.waitforprimaryonstartup=true
   b. Optional: Use this property to change the default time the secondary server waits for the primary server:
      impact.server.waitforprimary.duration=<*time in miliseconds*>
      If you do not use this property or provide no value for it, by default, the secondary server waits for 240000 miliseconds (4 minutes).
2. Restart the server to apply the changes.

Configuring the event processor service in a cluster

In a clustered environment, changes made to the event processor, do not automatically propagate out from the primary Impact Server to the other servers in the cluster.

You can change the event processor service properties through the GUI or by using the Command Line Interface. The GUI connects only to the primary server in a cluster, so to edit the Event Processor configuration in a secondary server you must use the Command Line Interface.

You must modify the event processor service settings on each Impact Server in cluster individually. If you change any configuration using the GUI or the Command Line Interface, there is no need to restart the server, you must restart only the Event Processor service. Do not make any configuration changes when the service is running and processing events. First stop the service, modify it, then restart the Event Processor service so that it picks up the new configuration. The same process applies when modifying the service through CLI for a secondary server. The Command Line reference provides the commands to stop and start the Event Processor and to make configuration changes.

When you configure the maximum number of threads for the event processor service, for optimal performance the number of processing threads should be greater than, or equal to the size of the connection pool of the SQL data sources used in the policies being triggered. For information about viewing existing thread and connection pool information, see the information in the *Administration Guide* in the section *Command-Line tools, Event Processor commands*. See the **`Select PoolConfig from Service where Name='EventProcessor';`**

Important: In a clustered environment, the event processor configuration is not replicated between servers. You must run the **`Select PoolConfig from Service where Name='EventProcessor';`** command on the primary and the secondary servers.

Use the same considerations when configuring the maximum threads on a secondary server. The secondary server uses its own connection pool which is independent of the size of the connection pool in primary server. For example, a DB2 data source has connection pool size of 30. The DB2 data source is replicated between primary and secondary servers. There could potentially be 30+30 = 60 connections made by Impact primary and secondary servers to the DB2 database. For optimal performance, the maximum number of threads with this setup of connection pool = 30 (should be at least 30 in each server of the cluster). The event processor configuration is not replicated between servers, so it must be set up manually in the secondary using CLI.

Chapter 6. Version control

Version control enable you to save policies, data sources, data types, operator views, and configuration properties as revisions in a source control archive.

Version control is enabled by default in Netcool/Impact, you cannot disable it.

Version control overview

The default version control system, that the version control manager uses is Impact Subversion (SVN), which is Subversion customized for use with Netcool/Impact.

It is installed automatically with the default settings. If you want to use another supported version control, you must install and configure it before installing Netcool/Impact and specify your selection during the installation.

Apart from Impact Subversion, Netcool/Impact supports External SVN, CVS, RCS, and ClearCase as version control systems.

Version control process

The main phases of the version control process are element creation, check out, check in, element deletion, and element renaming.

Element creation

Element creation occurs when you create an item using the GUI or CLI.

Check out

Check out occurs when you open an existing file. At check out, the most recent revision is checked out and locked in the version control system. When an element is locked, the locked icon displays in the GUI. Only the user who has currently checked out that element can modify it.

Check in

Check in occurs when you save an item. At check in, the corresponding file is checked in as a new revision.

Element deletion

Element deletion occurs when you delete an item using the GUI or CLI. At element deletion, the entire element or archive is removed from the version control system.

Element renaming

Element renaming occurs when you rename a policy, data source, or data type using the GUI or CLI. You cannot rename an operator view once it has been saved. When you rename an element, a new element is created using the new name in the version control system. The old element with the previous name is not deleted.

Configuring version control

You can change the version control configuration at any time after you have installed Netcool/Impact.

Use the command line service on the secondary server to update the configuration in the version control properties file, $IMPACT_HOME/etc/*server*_versioncontrol.props. The *server* part, is the name of the instance of the Impact Server.

The following table shows the version control configuration properties.

Table 11. Version Control Configuration Properties

Property	Description
impact.versioncontrol.class	Class name for the version control system to use.
impact.versioncontrol.debug	Reserved for internal use.
impact.versioncontrol.path	Path to the version control executable (CVS and RCS only)
impact.versioncontrol.repository	Location of the version control archive (CVS only)

Version control script

You use the provided version control script, $IMPACT_HOME/bin/nci_version_control, to perform housekeeping and maintenance tasks on data stored in the version control system that you cannot otherwise access using the GUI or the command-line interface.

The primary application of this script is storing in the version control the changes to the copy of the server properties file, the nameserver properties file, the JRExec server properties file, and other server-level configuration files. You can also use it to check in and check out copies of policies files, data source files, data type files, and service properties files. The best practice, however, is to avoid using the script directly for this level of management. For example, in a server cluster changes made on a per-server basis to these files are not propagated to other members of a server cluster. Changes are also not propagated to other parts of the system where meta information is stored. You should use the GUI or command line to manage policies, data sources, data types, and services.

You use the version control script to check out files, check in files, add files, uncheck out files, create a checkpoint, and update the sandbox.

Checking out files

Use this version control script syntax to check out a file from the version control system.

Procedure

nci_version_control *server* co *filename username*
where *server* is the name of the Impact Server, *filename* is the name and relative path of the file and *username* is a valid Netcool username.
The file path you specify is relative to the $IMPACT_HOME directory. For example, if you want to check out the property file for the default event reader in a server named NCI, you specify etc/NCI_server.props as the file name and path. For example:

```
nci_version_control NCI co etc/NCI_server.props admin
```

When you run the version control script, it checks out the file and copies it to the $IMPACT_HOME/etc or $IMPACT_HOME/policy directory. You must check the file back in order for any changes to the Netcool/Impact configuration to take effect.

Checking in files

Use this version control script syntax to check in a file to the version control system.

Procedure

`nci_version_control` *`server`* `ci` *`filename comment username`*
where *server* is the name of the Impact Server, *filename* is the name and relative path of the file, *comment* is a string that describes the check in and *username* is a valid Netcool Virtual Member Manager username. For example:

```
nci_version_control NCI ci etc/NCI_server.props "Checking in new version" admin
```

As with other version control commands, the file name and path that you specify must be relative to the $IMPACT_HOME directory.

Adding files

Use this version control script syntax to add a new file to the version control system.

Procedure

`nci_version_control` *`server`* `add` *`filename username`*
where *server* is the name of the Impact Server, *filename* is the name and relative path of the file, and *username* is a valid Netcool username. For example:

```
nci_version_control NCI add policy/MY_POLICY.ipl admin
```

As with other version control commands, the file name and path that you specify must be relative to the $IMPACT_HOME directory.

Unchecking out files

Use this version control script syntax to remove the lock from a file that you have previously checked out without performing a new check in.

Procedure

`nci_version_control` *`server`* `unco` *`filename username`*
where *server* is the name of the Impact Server, *filename* is the name and relative path of the file, and *username* is a valid Netcool Virtual Member Manager username. For example:

```
nci_version_control NCI unco etc/NCI_server admin
```

As with other version control commands, the file name and path that you specify must be relative to the $IMPACT_HOME directory.

Creating a checkpoint

Use this version control script syntax to create a checkpoint.

Procedure

`nci_version_control` *`server`* `checkpoint` *`checkpoint_id`*
where *server* is the name of the Impact Server, and *checkpoint_id* is a string that you want to use to identify the version control checkpoint. For example:

```
nci_version_control NCI checkpoint config_build_01
```

Updating files

Use this version control script syntax to update a file or a policy to a particular version. If you specify a revision number, that version replaces the version of the file you are currently using.

Procedure

Use the following syntax:

`nci_version_control` *`server`* `update` *`revision item`*

Where *server* is the name of the Impact Server, *revision* is the version of the file you want to update to and *item* is the policy or file that you want to update.
For example, to update the policy, *EnrichEvent* to *revision 29* where the server is <*NCI*> and installation home is under `/opt/ibm/netcool`, use the following command:

```
nci_version_control NCI update r29 /opt/ibm/netcool/impact/policy/NCI_EnrichEvent.ipl
```

Chapter 7. Managing the database server

You use a database script perform database server administration tasks.

You run the `$IMPACT_HOME/bin/nci_db` script using the HSQL command line client.

The database server overview

The database server is a specially configured version of an embedded HSQL that has been prepared for use with Netcool/Impact.

The database server is installed automatically, and it runs as a service in the Impact Server. You need to run the service only in the following circumstances:

- You are using the GUI reporting tools.
- You need to use a local SQL database data source.
- The database server is required for another product running on the same system.

The database that is managed by the database server is a HSQL database, `$IMPACT_HOME/db/`*`servername`*`_impact`, where *servername* is the name of the Impact Server. The database contains tables that store reporting information generated by the Impact Server. It is used to store the underlying data used by the GUI reporting tools.

Restriction: Do not use the HSQL database that is shipped with Netcool/Impact for your production installations.

You start and stop the database server using the ImpactDatabase service. The administrative tasks with the database server are performed using the database script.

Setting the database port

By default, the database server uses port 5435.

Procedure

1. You can change the default port by editing the database properties file.
   a. Open the `$IMPACT_HOME/etc/`*`servername`*`_impactdatabase.props`.
      servername is the name of the Impact Server.
   b. Change the port number in the `impact.impactdatabase.port` property.
   c. Save the file, and restart the service to apply the change.
2. You can change the ImpactDatabase service properties in the service editor in the TIP GUI.
   For more information about editing services in GUI, see the *Working with services* chapter in the *User Interface Guide*.

Running the database server

You start and stop the database server in the **Services** page in the Tivoli Integrated PortalGUI.

In the TIP navigation pane, click **System Configuration** > **Even Automation** > **Services**. Then select the **Global** project to view all services. The database server is displayed in the list of services as ImpactDatabase service. For more information about starting and stopping services, see the *Working with services* chapter in the *User Interface Guide*.

Resetting the database

Resetting the database removes all of the historical data stored by the GUI reporting tools.

Procedure

1. Stop the ImpactDatabase service.
2. Copy the files from `$IMPACT_HOME/install/dbcore` to `$IMPACT_HOME/db` directory and rename them based upon the server name.
3. Restart the ImpactDatabase service to re-initialize it to the base setup.

Connecting to the database with the command line client

The command line client is a SQL tool that is distributed with the HSQL database.

Procedure

To connect to the HSQL database with the client, enter the following command at a command prompt:

`$IMPACT_HOME/bin/nci_db connect <Impact_Server_Name> <Port>`

You can use the client to run SQL statements against the database in real time.

Backing up the database

Use this procedure to back up the database.

Procedure

1. Stop the ImpactDatabase service.
2. Back up the files from the `$IMPACT_HOME/db` directory.

Restoring the database

Use this procedure to restore the database from a backup file.

Procedure

1. Stop the ImpactDatabase service.
2. Copy the backup to the `$IMPACT_HOME/db` directory.

Chapter 8. JRExec server

You use the JRexec server to run external commands, scripts, and applications from within a policy.

Overview of the JRExec server

The JRExec server is a runnable server component of Netcool/Impact that you use to run external commands, scripts, and applications from within a policy, on the same system where Netcool/Impact is installed.

The JRExec server is automatically installed when you install Netcool/Impact. On Windows systems, you must also manually add the JRExec server as a Windows service. You run the JRExec server either using the JRExec server script or through the services administration tools, depending on the operating system. The server is configured through a property file.

You use the JRExecAction function to run external commands from within a policy. For more information about the JRExecAction function, see the *Policy Reference Guide*.

Starting the JRExec server

Use this procedure to start the JRExec server.

Procedure

- On UNIX systems you use the JRExec Server startup script, `nci_jrexec`, located in the `$IMPACT_HOME/bin` directory.

  Run this command in the terminal:

  ```
  ./nci_jrexec
  ```

- On Windows systems you start the JRExec server service, in the Services management console.

  Right-click **Netcool JRExec Server** in the Services window that opens, and select **Start**.

Stopping the JRExec server

Use this procedure to stop the the JRExec server.

Procedure

- On Windows systems you stop the JRExec server service, in the Services management console.

  Right-click **Netcool JRExec Server** in the Services window that opens, and select **Stop**.

- On UNIX systems you must manually terminate the process.

  Two processes are associated with the JRExec Server: the nci_jrexec process, and a JAVA process that was started by the nci_jrexec process.

  1. Obtain their IDs using these commands:
     - `ps -eaf | grep nci_jrexec`

       This command returns the PID of the nci_jrexec process.

– `ps -eaf | grep java`

  Apart from the Impact Server, and the GUI Server process ID, this command should return this process:

```
501      16053     1  1 13:58 pts/2
00:00:02 /home/netcool_usr/IBM/tivoli/tipv2/java/bin/java
-Dibm.tivoli.impact.propertiesDir=/home/netcool_usr/IBM/tivoli/impact/etc
 -Dbase.directory=/home/netcool_usr/IBM/tivoli/impact
 -Dnetcool.productname=impact
-classpath /home/netcool_usr/IBM/tivoli/impact/lib/nciJmxClient.jar:
/home/netcool_usr/IBM/tivoli/tipv2/lib/ext/log4j-1.2.15.jar
 com.micromuse.response.client.RemoteJRExecServerImpl
```

  This is only an example and the PID, and the path of the process. They will be different on your system.

2. Kill both processes using this command:

   `kill -9 pid`

   where *pid* is one of the two process IDs associated with the JRExec Server.

The JRExec server configuration properties

The JRExec server properties file, `jrexecserver.props`, is located in the `$IMPACT_HOME/etc` directory.

The file may contain the following properties:

impact.jrexecserver.port

To change the port number used by the JRExec server. Default is 1345. If you change this property, you must also update the value of the `impact.jrexec.port` property in the *NCI*`_server.props` file, where *NCI* is the name of the Impact Server instance.

impact.jrexecserver.logfile

To enable logging for the JRExec server. Set its value to the path and filename of the target JRExec server log file. For example, `impact.jrexecserver.logfile=/opt/IBM/tivoli/impact/logs/jrexecserver.log`.

JRExec server logging

To enable logging for the JRExec server, you must create a properties file and a log file.

1. Create a properties file called `jrexecserver-log4j.properties` in the `$NCHOME/impact/etc` directory.
2. Define the following properties in the properties file:

```
log4j.rootCategory=INFO
log4j.appender.JREXEC=org.apache.log4j.RollingFileAppender
log4j.appender.JREXEC.threshold=DEBUG
log4j.appender.JREXEC.layout=org.apache.log4j.PatternLayout
log4j.appender.JREXEC.layout.ConversionPattern=%d{DATE} %-5p [%c{1}] %m%n
log4j.appender.JREXEC.append=true
log4j.appender.JREXEC.file=<$NCHOME>/impact/log/nci_jrexec.log
log4j.appender.JREXEC.bufferedIO=false
log4j.appender.JREXEC.maxBackupIndex=3
log4j.appender.JREXEC.maxFileSize=10MB
```

   Ensure that you use the full path for the <*$NCHOME*> value.

3. You also must set `DEBUG` as the default priority for all micromuse and IBM loggers in the same file:

```
log4j.category.com.micromuse=DEBUG,JREXEC
log4j.additivity.com.ibm.tivoli=false
log4j.additivity.com.micromuse=false
```

4. Create a log file that is called `nci_jrexec.log` in the `$NCHOME/impact/log` directory.

Chapter 9. Self-monitoring

Self-monitoring is used to monitor such aspects of Netcool/Impact runtime performance, as memory status usage, event queue size, data source status, service status, and cluster status.

Self-monitoring overview

The Netcool/Impact performance information gathered using self-monitoring is reported to the Netcool/OMNIbus ObjectServer as events.

The events that are sent to the ObjectServer as part of self-monitoring are the same as any other Netcool/OMNIbus events. These events can be viewed by Netcool operators in an event list and managed according to the normal event handling procedures in your environment. You can also configure the event reader to launch custom policies that are designed to respond to these events.

The self-monitoring feature is provided by the `SelfMonitoring` service. The service is created automatically when you installNetcool/Impact. You cannot create new instances of this service. To set up the self-monitoring feature, you configure the `SelfMonitoring` service using the GUI or command line interface (CLI). For more information setting up and running self-monitoring, see "Running self-monitoring using the command line interface" on page 89 and "Setting up self-monitoring using the GUI."

If you are running Impact Servers in a cluster, self-monitoring operates on a per-server basis. For more information about self-monitoring in a server cluster, see "Self-monitoring in a server cluster."

Self-monitoring in a server cluster

You configure self-monitoring on each server instance separately, and each server runs the self-monitoring feature independent of other cluster members.

Service configuration information is not propagated between members of the cluster which allows you to customize self-monitoring for each system where you are running the cluster.

For single-server configurations of Netcool/Impact, using the GUI to set up self-monitoring is recommended. For a clustered server configuration, set up self-monitoring for the secondary members using the `$IMPACT_HOME/etc/`*`servername`*`_selfmonitoring.props` properties file, or CLI since the GUI only configures the primary server.

If you do not want to monitor each server in a cluster you can enable or disable the monitoring elements on a server in a cluster. For more information, see "Cluster Status Monitoring" on page 105

Setting up self-monitoring using the GUI

Use this procedure to set up self-monitoring using the GUI.

Procedure

1. Log in to the Tivoli Integrated Portal as a privileged user, for example `tipadmin`.
2. From the Tivoli Integrated Portal navigation panel on the left, select **System Configuration** > **Event Automation** > **Services**.
3. In the **Services** tab, from the **Project** menu, select the **Global** project.
4. Right-click the **SelfMonitoring** service in the panel and select **Edit** to open the service for editing. From the **ObjectServer Data Source** menu, select **defaultobjectserver**.
5. To start SelfMonitoring you must also enable at least one of the following options:
   - **Memory Status** Select to send status events regarding queue conditions of the event readers that are currently running. Use the interval selector to change the interval at which SelfMonitoring sends status events to the Object Server, which by default is 60 seconds.
   - **Queue Status** Select to enable the service to send events about the status of the event readers and listeners currently running. Use the interval selector to change the interval at which SelfMonitoring sends queue status events to the Object Server, which by default is 60 seconds. Enable or disable the **Deduplication** check box. The deduplication option specifies if event queue status events sent to the ObjectServer are deduplicated or not. If you change the deduplication option while the service is running, you must stop and restart it before the change takes effect.
   - **Cluster Status** Select to start sending events about the status of the cluster. Events are sent when an Impact Server is started and joins the cluster. An Impact Server is stopped and removed from the cluster; or the primary server is down and a secondary server becomes the new primary.
   - **Data Source Status** Select to start sending the status events when certain conditions occur with a data source. For example, SelfMonitoring sends a status message when a user tests a connection to a data source or when a connection cannot be established.
   - **Service Status** Select to start sending service status events to the Object Server.
6. You can optionally configure SelfMonitoring to start automatically when the Impact Server starts by selecting the **Startup: Automatically when server starts** check box.
7. To write self-monitoring events to the log file, select the **Service Log: Write to file** check box. The data is logged to the following locations:
   - On Windows: `$IMPACT_HOME/logs/<impact_server>_selfmonitoring.log`
   - On Linux: `/opt/IBM/tivoli/impact/logs/NCI_selfmonitoring.log`

   If Netcool/Impact is installed as part of the Business Service Manager suite the location of the log file is
   - On Windows `$TBSM_HOME/logs/<tbsm_server>_selfmonitoring.log`
   - On Linux `/opt/IBM/tivoli/tbsm/logs/TBSM_selfmonitoring.log`
8. Save the changes, then start or restart SelfMonitoring.
   - If the service is stopped, right-click SelfMonitoring and select Start.
   - If the service is running, right-click SelfMonitoring and select Stop, then right-click again and Start.

   You can also configure self monitoring by using the Command Line Interface.

Running self-monitoring using the GUI

Follow this procedure to run the self-monitoring service using the GUI.

Before you begin

If you are running the Netcool/Impact deployment in a clustered configuration, you must use the command line interface to start and stop non-primary instances of this service in the cluster. For information on starting and stopping the service using the command line interface, see "Running self-monitoring using the command line interface."

Procedure

1. Open the GUI in a Web browser.
2. In the **Service Status** panel, click:
   a. The **Start** button next to the `SelfMonitoring` service to start the service.
   b. The **Stop** button next to the `SelfMonitoring` service to stop the service.

Running self-monitoring using the command line interface

Use this procedure to run self-monitoring using the command line interface.

Procedure

- To start the self-monitoring service, enter the following command at the CLI prompt:

  ```
  UPDATE Service SET Running = true WHERE Name = 'SelfMonitoring';
  ```

- To stop the self-monitoring service, enter the following command at the CLI prompt:

  ```
  UPDATE Service SET Running = false WHERE Name = 'SelfMonitoring';
  ```

Setting the ObjectServer Data Source for monitoring events

Use this procedure to set the ObjectServer data source for monitoring events.

Procedure

- To check the current ObjectServer DataSource where monitoring events are sent use this command:

  ```
  Select DataSource from Service where Name = 'SelfMonitoring';
  ```

- To change the ObjectServer DataSource where monitoring events are sent use this command:

  ```
  Update Service set DataSource='ObjDS' where Name='SelfMonitoring';
  ```

  where 'ObjDS' is the name of the ObjectServer Data Source.

  Remember: Restart the Self Monitoring service after changing the DataSource.

Memory status monitoring

Memory status monitoring is the process by which the self-monitoring service checks the available Java and system memory and sends events to the Netcool/OMNIbus ObjectServer regarding the memory status at intervals.

Netcool/Impact monitors available memory in both the Java heap and in the system as a whole. The memory status is significant because Netcool/Impact will fail and report an out-of-memory error if the maximum available memory is exceeded.

Disabling buffer and cache size reporting in free memory calculations on Linux

In the installations on Linux systems, the self monitoring service also includes the buffer and cache size in its free memory reporting.

Procedure

1. To disable the buffer and cache size reporting, set the following property in the `$IMPACT_HOME/etc/`*`servername`*`_selfmonitoring.props` file:

   ```
   impact.selfmonitoring.includebuffer=false
   ```

2. Restart the server for the change to take effect.

   After the restart, the free memory reporting will now include only the system memory values.

Configuring available system memory for self monitoring

You can include free swap memory and free system memory when calculating the total available memory for self monitoring.

About this task

On UNIX platforms, the default behavior is to not include the swap memory when calculating the total free memory. Complete the following steps if you want to include swap memory:

Procedure

1. Set the following property in the `$IMPACT_HOME/etc/<`*`ServerName`*`>_selfmonitoring.props` props file:

   ```
   impact.selfmonitoring.includeswap=true
   ```

2. Restart the Impact Server to implement the changes to include free swap and free system memory.

Setting the memory for the Java Virtual Machine on the Impact profile

When you start an instance of the JVM, you can specify a minimum and maximum heap size by using the `-Xms` and `-Xmx` flags. The `-Xms` flag specifies the minimum size of the memory heap and `-Xmx` specifies the maximum size.

About this task

For Netcool/Impact, the default value of `Xms` is 256 MB and the default value of `Xmx` is 1200 MB for 32 bit and 64 bit installations. It is better to increase the heap size to more than the default setting of 1200MB as it can improve performance. In a 64 bit installation, as a guideline increase the heap setting to 80% of the free memory on your system.

Consider the following factors when you are increasing the heap size in either a 32 bit or 64 bit installation:

- Event flow volume

- The number of SQL data types and the amount of data in them
- Internal data types and the size of them
- The number of event readers and event listeners
- The number of hibernations

You can use the self monitoring service to monitor the heap usage, For more information setting up and running self-monitoring, see "Running self-monitoring using the command line interface" on page 89 and "Setting up self-monitoring using the GUI" on page 87.

The following procedure shows how to set the maximum heap size. You may also want to do the same for **TIPProfile** especially if you plan to view large amounts of data in the browser. For example, when viewing data items for an SQL data type which contains large rows of information. To change the memory settings for the **TIPProfile**, see "Setting Java Virtual Machine memory for TIPProfile" on page 92.

Procedure

1. To set the maximum heap size to different value, you must run the `wsadmin.sh` script in the `$INSTALL_HOME/tipv2/profiles/ImpactProfile/bin` directory. For Windows use `wasadmin.bat`
   - If the Impact Server is not running, use the following command:

     ```
     ./wsadmin.sh -conntype NONE
     ```
2. When prompted type the user name and password. The default user name is `tipadmin` and the password is set up by the user at installation time.
3. You can check the current minimum and maximum heap size by using the following command:

   ```
   $AdminTask showJVMProperties {-serverName server1 -nodeName ImpactNode}
   ```

   The returned information is similar to the following example:

   ```
   {classpath {}} {bootClasspath {}}
   {verboseModeClass false} {verboseModeGarbageCollection false}
   {verboseModeJNI false} {initialHeapSize 256}
   {maximumHeapSize 1200} {runHProf false}
   {hprofArguments {}}
   {debugMode false}
   {debugArgs -agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=7784}
   {genericJvmArguments {}}
   {executableJarFileName {}}
   {disableJIT false} {osName {}}
   {internalClassAccessMode ALLOW}
   ```
4. To change the heap size to 1500 MB for example, at a command prompt, type the following command:

   ```
   $AdminTask setJVMMaxHeapSize {-serverName server1 -nodeName
   ImpactNode -maximumHeapSize 1500}
   ```

   The command prints `true` after you run it.
5. Save the configuration changes, and exit the administration console.

   ```
   $AdminConfig save
   exit
   ```
6. Restart the Impact Server.

Results

When you enable the memory monitoring feature in the self monitoring service, Netcool/Impact checks the current heap size at intervals and compares it to the maximum size. The severity of the memory monitoring event is calculated based on the heap usage and the available system memory. For more information see, "Memory status severity" on page 93.

Setting Java Virtual Machine memory for TIPProfile

You can increase the amount of memory available to the Tivoli Integrated Portal.

About this task

To increase (or decrease) the amount of memory available to the Java Virtual Machine (JVM), carry out the following steps:

Procedure

1. Manually stop the application server.
2. Change to the *tip_home_dir*/profiles/TIPProfile/bin directory.
3. Use the **wsadmin** command to increase the heap size for the JVM, as follows:
   ```
   wsadmin.sh -lang jython -conntype NONE
   ```
4. At the wsadmin> prompt, issue the following commands, where *xxx* is the new heap size value, in megabytes.
   ```
   jvm=AdminConfig.list("JavaVirtualMachine")

   AdminConfig.modify(jvm, '[[initialHeapSize xxx]]')
   AdminConfig.modify(jvm, '[[maximumHeapSize xxx]]')
   AdminConfig.save()

   exit
   ```
5. Restart the Tivoli Integrated Portal Server. The changes take effect when the Tivoli Integrated Portal Server is restarted.

   Attention: If you attempt to start the Tivoli Integrated Portal Server with a maximum heap size that is too large, error messages that are similar to the following are generated in the *tip_home_dir*/profiles/TIPProfile/logs/server1/native_stderr.log file:
   ```
   JVMJ9GC019E -Xms too large for -Xmx
   JVMJ9VM015W Initialization error for library j9gc23(2): Failed to initialize
   Could not create the Java virtual machine.
   ```

System memory status

In addition to Java heap memory, the memory monitoring feature also monitors available system memory as a whole.

At maximum, Netcool/Impact uses the amount of Java heap memory specified in the XMX variable as well as an additional 100 to 120 megabytes of system memory. In order to effectively report on the memory status for the system, the total amount of system memory available at a given time compared to the amount of memory currently in use by the application must also be taken into consideration .

When you enable the self-monitoring feature, Netcool/Impact checks the current available system memory at intervals and compares it to the maximum memory that it requires. The maximum memory is determined by adding 150 megabytes to the maximum amount allocated to the Java heap. For example, if the XMX variable

is set to `1000M`, the maximum allocated memory is determined to be 1150 megabytes. Then the severity of the system memory status is calculated using the rules in "Memory status severity."

Combined memory status

After calculating the severity of the JVM and system memory status, the event is sent to the ObjectServer.

This event contains the fields described in "Memory event fields." The total severity of the event is the highest severity between the JVM and system memory status.

Memory status severity

The following criteria are used to determine the severity of the JVM memory status.

Table 12. Severity of the JVM memory status

Severity	Criteria
1	Maximum heap limit is greater than twice the current help size.
2	Maximum heap limit is between 1.6 and 2 times the current heap size.
3	Maximum heap limit is between 1.4 and 1.6 times the current heap size.
4	Maximum heap limit is between 1.2 and 1.4 times the current heap size.
5	Maximum heap limit is less than 1.2 times the current heap size.

The following criteria are used by Netcool/Impact to determine the severity of the system memory status.

Table 13. Severity of the system memory status

Severity	Criteria
1	Available system memory is greater than twice the maximum required memory.
2	Available system memory is between 2 and 1.8 times the maximum required memory.
3	Available system memory is between 1.8 and 1.65 times the maximum required memory.
4	Available system memory is between 1.65 and 1.5 times the maximum required memory.
5	Available system memory is less than 1.5 the maximum required memory.

Note: The total severity of any memory event sent to the ObjectServer is the highest severity between the JVM and the system memory status.

Memory event fields

This topic provides a description for the fields in memory events sent to the ObjectServer.

Table 14. Fields in memory events sent to the ObjectServer

Field	Description
Identifier	Impact Memory Status for *server*, or Impact Memory Status for *server* at <*timestamp*> if deduplication is disabled, where *server* is of the *servername:clustername* format. For example: NCI1:NCLUSTER
Node	Host name of the system where the Impact Server is running.
NodeAlias	IP address of the system where the Impact Server is running.
Severity	Severity calculated according to the rules in "Memory status severity" on page 93.
Summary	Detailed information about the memory status, including information about JVM heap usage and system memory usage.
Class	10500
Type	13
AlertGroup	MemoryStatus
FirstOccurrence	Timestamp for the first occurrence of this event.
LastOccurrence	Timestamp for the most recent occurrence of this event.
Manager	Impact SelfMonitoring@<*clustername*>. For example: Impact SelfMonitoring@NCICLUSTER.
Agent	Impact 6.1 SelfMonitoring

Checking if memory status monitoring is enabled

To check to see if memory status monitoring is enabled, enter the following command at the CLI prompt.

Procedure

```
SELECT IsMemoryStatusEnabled FROM Service WHERE Name = 'SelfMonitoring';
```

The CLI returns a value of `true` if memory status monitoring is enabled. Otherwise, the CLI returns a value of `false`.

Enabling memory status monitoring

To enable memory status monitoring, enter the following command at the CLI prompt.

Procedure

```
UPDATE Service SET EnableMemoryStatus = true where Name='SelfMonitoring';
```

Disabling memory status monitoring

To disable memory status monitoring, enter the following command at the CLI prompt.

Procedure

```
UPDATE Service SET EnableMemoryStatus = false where Name='SelfMonitoring';
```

Viewing current memory status

To check to see if memory status monitoring is enabled, enter the following command at the CLI prompt.

Procedure

```
SELECT MemoryStatus FROM Service WHERE Name='SelfMonitoring';
```

Viewing memory status history

To check to see if memory status monitoring is enabled, enter the following command at the CLI prompt.

Procedure

```
SELECT MemoryStatusHistory FROM Service WHERE Name = 'SelfMonitoring';
```

Viewing the total JVM heap size

To view the total JVM heap size, enter the following command at the CLI prompt.

Procedure

```
SELECT TotalVMHeapSize FROM Service WHERE Name = 'SelfMonitoring';
```

Viewing the maximum JVM heap size

To view the maximum JVM heap size, enter the following command at the CLI prompt.

Procedure

```
SELECT MaxVMHeapSize FROM Service WHERE Name = 'SelfMonitoring';
```

Viewing the free system memory

To view the free system memory, enter the following command at the CLI prompt.

Procedure

```
SELECT FreeSystemMemory FROM Service WHERE Name = 'SelfMonitoring';
```

Checking if memory status events are deduplicated

To check to see if memory status events are being deduplication, enter the following command at the CLI prompt.

Procedure

```
SELECT MemoryDeduplication FROM Service WHERE Name = 'SelfMonitoring';
```

The CLI returns a value of `true` if memory status event deduplication is enabled. Otherwise, the CLI returns a value of `false`.

Disabling memory status event deduplication

To disable memory status event deduplication, enter the following command at the CLI prompt.

Procedure

```
UPDATE Service SET MemoryDeduplication = false WHERE Name=
'SelfMonitoring';
```

Viewing memory status event intervals

To view the intervals at which the memory status is checked, enter the following command at the CLI prompt.

Procedure

```
SELECT MemoryInterval FROM Service WHERE Name = 'SelfMonitoring';
```

Changing memory status event intervals

To change the interval at which the Impact Server checks the memory status, enter the following command at the CLI prompt.

Procedure

```
UPDATE Service SET MemoryInterval = interval WHERE Name = 'SelfMonitoring';
```
where *interval* is the number of seconds at which the server checks the queue status,

Queue status monitoring

Queue size monitoring is the process in which the self-monitoring service checks the size of event queues at intervals and sends events to the ObjectServer regarding the event queue status.

For each event, the self-monitoring service calculates the severity by determining the rate at which the queue size is increasing or decreasing since the last interval point.

Queue size monitoring monitors the OMNIbusEventReader, DatabaseEventReader, OMNIbusEventListener, DatabaseEventListener, JMSMessageListener, and EventProcessor services.

The queue status monitoring sends one event per running service it monitors. For example, if OMNIbusEventReader, DatabaseEventReader and EventProcessor are running in the server, then queue monitoring will send 3 events to OMNIbus, one for each service.

The queue monitoring service samples the queue size for the services it is currently monitoring every 5 seconds. When a queue monitoring event is about to be sent, the maximum and minimum values that were found during those sampling periods will be included along with the current queue size. Here is an example of a queue monitoring event:

```
EventProcessor: QueueSize: 100 (Min: 65, Max: 120)
```

For more information on the rules used to determine the severity, see “Queue status severity” on page 97. For more information on the value of fields in the events sent to the ObjectServer see “Queue status event fields” on page 97.

Changing the sampling frequency for the queue monitoring

Use this procedure to change the default sampling frequency for the queue monitoring service.

Procedure

1. Change the value of the `impact.selfmonitoring.samplefrequency` property in the `IMPACT_HOME/etc/servername_selfmonitoring.props` file.

   The sampling frequency is given in seconds. For example:

   ```
   impact.selfmonitoring.samplefrequency=10
   ```

   With this setting, the queue is sampled every 10 seconds instead of the default 5.
2. Restart the server so that it picks up the change.

Queue status severity

The following criteria are used to determine the severity of the event queue status.

Table 15. Severity of the event queue status

Severity	Criteria
1	Number of events in queue is less than or equal to 1.5 times the number of events at previous interval.
2	Number of events in queue is between 1.5 and 2. times the number of events at previous interval.
3	Number of events in queue is between 2 and 3 times the number of events at previous interval.
4	Number of events in queue is between 3 and 5 times the number of events at previous interval.
5	Number of events in queue is more than 5 times the number of events at previous interval.

Queue status event fields

This topic provides a description of the fields in memory events sent to the ObjectServer.

Table 16. Fields in memory events sent to the ObjectServer

Field	Description
`Identifier`	`Identifier Impact Queue Status for server`, or `Impact Queue Status for server at <timestamp>` if deduplication is disabled, where *server* is of the *servername:clustername* format. For example: `NCI1:NCLUSTER`.
`Node`	Host name of the system where the Impact Server is running.
`NodeAlias`	IP address of the system where the Impact Server is running.
`Severity`	Severity calculated according to the rules in "Queue status severity."
`Summary`	Detailed information about the queue status, as described in the table below.
`Class`	`10500`
`Type`	`13`
`AlertGroup`	`QueueStatus`
`FirstOccurrence`	Timestamp for the first occurrence of this event.
`LastOccurrence`	Timestamp for the most recent occurrence of this event.
`Manager`	`Impact SelfMonitoring@<clustername>`. For example: `Impact SelfMonitoring@NCICLUSTER`.

Table 16. Fields in memory events sent to the ObjectServer (continued)

Field	Description
Agent	Impact 6.1 SelfMonitoring

This table shows the queue size information sent as the contents of the Summary field.

Table 17. Queue size information sent as the contents of the Summary field

Value	Description
Service name	Name of the service associated with the queue.
QueueSize	Current® number of events in queue.
DeltaQueue	Change in queue size since the last interval. If this value is greater than zero, the queue size has increased. If the value is less than zero, the queue size has decreased.
QueueIncreaseRate	Rate at which the queue is increasing. This value is generated if DeltaQueue is greater than zero.
QueueDecreaseRate	Rate at which the queue is decreasing. This value is generated if DeltaQueue is less than zero.
Gap	Time difference between the StateChange of the earliest event in queue and the current time. This is only related to OMNIbusEventListener and OMNIbusEventReader.

Checking if queue status monitoring is enabled

To check to see if queue status monitoring is enabled, enter the following command at the CLI prompt.

Procedure

```
SELECT IsQueueStatusEnabled FROM Service WHERE Name = 'SelfMonitoring';
```

The CLI returns a value of true if queue status monitoring is enabled. Otherwise, the CLI returns a value of false.

Enabling queue status monitoring

To enable queue status monitoring, enter the following command at the CLI prompt.

Procedure

```
UPDATE Service SET EnableQueueStatus = true WHERE Name = 'SelfMonitoring';
```

Disabling queue status monitoring

To disable queue status monitoring, enter the following command at the CLI prompt.

Procedure

```
UPDATE Service SET EnableQueueStatus = false WHERE Name = 'SelfMonitoring';
```

Viewing the current queue status

To view the current queue status, enter the following command at the CLI prompt.

Procedure

```
SELECT QueueStatus FROM Service WHERE Name = 'SelfMonitoring';
```

Viewing the queue status history

To view the queue status history, enter the following command at the CLI prompt.

Procedure

```
SELECT QueueStatusHistory FROM Service WHERE Name = 'SelfMonitoring';
```

Checking if queue status event deduplication is enabled

To check to see if queue status event deduplication is enabled, enter the following command at the CLI prompt.

Procedure

```
SELECT QueueDeduplication FROM Service WHERE Name = 'SelfMonitoring';
```

The CLI returns a value of `true` if queue status event deduplication is enabled. Otherwise, the CLI returns a value of `false`.

Enabling queue status event deduplication

To enable queue status event deduplication, enter the following command at the CLI prompt.

Procedure

```
UPDATE Service SET QueueDeduplication = true WHERE Name = 'SelfMonitoring';
```

Disabling queue status event deduplication

To disable queue status event deduplication, enter the following command at the CLI prompt.

Procedure

```
UPDATE Service SET QueueDeduplication = false WHERE Name =
'SelfMonitoring';
```

Viewing queue status event intervals

To view the interval at which the Impact Server checks the queue status, enter the following command at the CLI prompt.

Procedure

```
SELECT QueueInterval FROM Service WHERE Name = 'SelfMonitoring';
```

Changing queue status event intervals

To change the interval at which the Impact Server server checks the queue status, enter the following command at the CLI prompt.

Procedure

`UPDATE Service SET QueueInterval = interval WHERE Name = 'SelfMonitoring';`
where *interval* is the number of seconds at which the server checks the queue status,

Data source status monitoring

Data status monitoring is a process in which the self-monitoring service sends events to the Netcool/OMNIbus ObjectServer that report the status of Netcool/Impact database connections.

Data source monitoring is supported for the following SQL database data sources:

- MySQL DSA
- ObjectServer DSA
- ODBC DSA
- Oracle DSA
- PostgreSQL DSA
- SQL Server DSA
- Sybase DSA
- GenericSQL DSA
- DB2
- MS-SQL Server
- HSQLDD
- Informix

Restriction: DataSource Monitoring events related to Connection Established/Disconnected will only be sent for the ObjectServer data sources different from the Object Server data source that is used by the Self Monitoring service.

Data source status events

The self-monitoring service sends data source status events to the ObjectServer under the following conditions:

Successful test connection to a data source
You can use the GUI to test connections to the primary and backup database servers when you create or configure an SQL database data source. When you successfully test a connection to a database server, the self-monitoring service sends an event to the ObjectServer with a severity of 1.

Failed test connection to a data source
If a successful test connection to a database server cannot be made, the self-monitoring service sends an event to the ObjectServer with a severity of 3.

Failed execution of an SQL query against a data source
SQL queries are sent to data sources under a variety of conditions (for example, when a function like `GetByFilter` or `AddDataItem` is processed in a policy or when you view the contents of a data type in the GUI). If an SQL query fails, another attempt is made to send the query to the database server. If the second attempt fails, the self-monitoring service sends an event to the ObjectServer with a severity of 3.

Termination of existing connections to a data source

When all existing connections to a database server are terminated, the self-monitoring service sends an event to the ObjectServer with a severity of 4. This happens, for example, when a primary database is no longer accessible and is switching to a backup database.

Data source status event fields

The following table shows the fields in data source status events sent to the ObjectServer.

Table 18. Fields in data source status events sent to the ObjectServer

Field	Description
`Identifier`	`Impact:DataSource Status for` *`server`*, where *`server`* is of the following format: *`servername:clustername:typenum:connectionstatus:state`* where: • `servername` is the name of the Impact Server. • `clustername` the cluster to which the server belongs to. • `typenum` is the value of the type field that is used to identify the problem and resolution status. • `connectionstatus` is a numeric value to denote the status that is used by Netcool/Impact for deduplication. • `state`, Primary or Secondary, is not always present. For example, when you click the **Test Connection** button after you edit a datasource. For example, `NCI:NCICLUSTER:2:0:Primary`.
`AlertKey`	*`servername:clustername:datasourcename:status`* For example, `NCI:NCICLUSTER:NCOMS:Primary` if a connection was established to the primary of NCOMS datasource. Note that when you either create a new datasource or test while you are editing a datasource, some information, like `datasourcename`, or `status` will not be known. The AlertKey will be constructed with whatever information is available.
`Node`	Host name of the system where the Impact Server is running.
`NodeAlias`	IP address of the system where the Impact Server is running.
`Severity`	Data source status event severity.
`Summary`	Detailed information about the data source status.
`Class`	`10500`
`Type`	It is set to indicate problem and resolution events - 1 to indicate a problem event and 2 to indicate a resolution event. Note that this does not cover the Query Failure type of events and for that case the Type would be 0 (information event).
`Agent`	`Impact Netcool/Impact 6.1 SelfMonitoring`
`AlertGroup`	`DatasourceStatus`
`Manager`	It is of the format `Impact SelfMonitoring@clustername`. For example: `Impact SelfMonitoring@NCICLUSTER`.
`FirstOccurrence`	Timestamp for the first occurrence of this event.
`LastOccurrence`	Timestamp for the most recent occurrence of this event.

This table shows the data source status information sent as the contents of the Summary field.

Table 19. Data source status information sent as the contents of the Summary field

Condition	Summary Field
Successful test connection	If you are testing the connection while creating a new data source, the value of the Summary field is Connection established to the Host: *hostname* and Port: *port_number*, where *hostname* is the host name or IP address of the system where the database server is running, *port* is the connection port for the database server. If you are testing the connection while editing an existing data source, the value of the summary field is Connection established to the *primary* | *backup* server for the DataSource: *datasource_name* on Host: *hostname* Port: *port_number*.
Failed test connection	If you are testing the connection while creating a new data source, the value of the Summary field is Could not connect to the Host: *hostname* and Port: *port_number*, where *hostname* is the host name or IP address of the system where the database server is running, *port* is the connection port for the database server. If you are testing the connection while editing an existing data source, the value of the summary field is Could not connect to the *primary* | *backup* server for the DataSource: *datasource_name* on Host: *hostname* Port: *port_number*.
Failed execution of an SQL query	Failed to execute query on the *primary* | *backup* server for the DataSource: *datasource_name* on Host: *hostname* Port: *port_num*. Invalidating all connections.
Termination of existing connections	Invalidating all connections to the *primary* | *backup* server for the data source *datasource_name* on host *hostname* port *port_num*.

Checking if data source status monitoring is enabled

Use this procedure to check to see if data source status monitoring is enabled.

Procedure

Enter the following command at the CLI prompt:

```
SELECT IsDataSourceStatusEnabled FROM Service WHERE Name = 'SelfMonitoring';
```

The CLI returns a value of `true` if data source status monitoring is enabled. Otherwise, the CLI returns a value of `false`.

Enabling data source status monitoring

Use this procedure to enable data source status monitoring.

Procedure

Enter the following command at the CLI prompt:

```
UPDATE Service SET EnableDataSourceStatus = true WHERE Name = 'SelfMonitoring';
```

Remember: Restart the service to apply the change.

Disabling data source status monitoring

Use this procedure to disable data source status monitoring.

Procedure

Enter the following command at the CLI prompt:

```
UPDATE Service SET EnableDataSourceStatus = false WHERE Name = 'SelfMonitoring';
```

Remember: Restart the service to apply the change.

Viewing the current data source status

Use this procedure to view the current data source status:

Procedure

Enter the following command at the CLI prompt:

```
SELECT DataSourceStatus FROM Service WHERE Name = 'SelfMonitoring';
```

Viewing the data source status history

Use this procedure to view the data source status history.

Procedure

Enter the following command at the CLI prompt:

```
SELECT DataSourceStatusHistory FROM Service WHERE Name = 'SelfMonitoring';
```

Deduplication of data source monitoring events

When deduplication is enabled, and identifier based on the datasource status is constructed.

The identifier contains the following information:

- Server name and cluster name
- Data source name
- Integer values representing the type of event and the connection status of the data source
- A string indicating if the connection is to the primary or backup database. This information is not available if you are creating a new datasource or doing a test connection of a existing datasource.

This way, all the events to a particular type of a “situation” are deduplicated into a single event. For example, the events for a server NCI belonging to a cluster NCICLUSTER for a datasource NCIDS to the Primary database are deduplicated into a single event. By default, the events are deduplicated. You can disable deduplication by setting a property in the self monitoring property file, or by issuing a command from the command line prompt.

Disabling deduplication of data source monitoring events

Use this procedure to disable deduplication of data source monitoring events using the command line prompt.

Procedure

1. To check if deduplication is enabled Issue the following command:
   ```
   select DataSourceDeduplication from Service where Name='SelfMonitoring';
   ```
2. Use this command to disable deduplication:
   ```
   update Service set DataSourceDeduplication=false where Name='SelfMonitoring';
   ```

   Attention: You can later enable deduplication later again, by issuing this command, with `DataSourceDeduplication=true` instead.

Pairing problem and resolution events together

Data source monitoring can flag the events so that they are treated as a problem and resolution event.

When Netcool/Impact fails to establish a connection to a data source, it sends a connection failure event to OMNIbus which is flagged as problem event. When a successful connection is established to the same data source, a connection established event is sent to OMNIbus. This connection is flagged as a resolution event. It sets the necessary information in the Node, AlertKey, AlertGroup, and Manager fields. Then the generic_clear trigger in OMNIbus identifies the problem and resolution event as a pair and eventually resolves the events.

In OMNIbus, the generic_clear trigger looks for resolution events where Type = 2 (Resolution event) and Severity > 0. If there is a match, it will change the Severity = 0 after 120 seconds. Then the delete_clear trigger clears the event.

You can disable event paring using the command line, or by setting the `datasourceclearpair` property in the `$IMPACT_HOME/etc/<servername>_selfmonitoring.props` file. Using the command line is the preferred method. The `datasourceclearpair` property is enabled by default.

Attention: If you set the `datasourceclearpair` property to false, you can no longer correlate the good and bad events.

Disabling pairing events

Use this procedure to enable or disable pairing events using the command-line prompt.

Procedure

1. To check if paring is enabled, enter the following command at the command-line prompt:
   ```
   SELECT DataSourceClearPair from Service WHERE Name='SelfMonitoring';
   ```
2. To disable paring events, enter the following command at the command-line prompt:
   ```
   UPDATE Service SET DataSourceClearPair=false WHERE Name='SelfMonitoring';
   ```

   Note: You can later enable paring events again using the same command, with `DataSourceClearPair=true` instead.
3. You must restart the **Self Monitoring** service to implement any changes to paring events.

Verbose mode in data source monitoring

Verbose mode in data source monitoring events, means that there will be a higher volume of events that are sent by Netcool/Impact as compared with similar information for non verbose mode.

For example, let us assume a datasource is down, and you click the **Test Connection** button. If verbose mode is not enabled, the request is sent only once, and any subsequent **TestConnection** that has the same information is not sent. By contrast, if verbose mode is turned on, an event is sent for each test connection.

By default verbose mode is disabled. To check if it is disabled or enabled issue this command:

```
select DataSourceVerbose from Service where Name='SelfMonitoring';
```

To enable verbose mode type the following command at the command-line prompt:

```
update Service set DataSourceVerbose=true where Name='SelfMonitoring';
```

Attention: You can use this command to disable verbose mode, if needed, by typing `DataSourceVerbose=false` instead of `DataSourceVerbose=true`.

Cluster Status Monitoring

Cluster status monitoring is a process in which the self-monitoring service sends events to the Netcool/OMNIbus ObjectServer that report the status of members in an Impact Server cluster.

Starting self-monitoring on a secondary cluster member

The self-monitoring service must be running on primary and secondary cluster members in order for this feature to report all possible status data to the ObjectServer. You can start the self-monitoring service on secondary cluster members using the CLI.

Procedure

1. Connect to the command line service running on the secondary server using a telnet application. For example, enter the following command at a command prompt:

   ```
   telnet hostname 2000
   ```

   where *hostname* is the name of the system where the secondary server is running.
2. Log in to the CLI using a valid username and password. The default administration username is `tipadmin`, and the password is `tippass`.
3. Use this command to enable cluster monitoring:

   ```
   UPDATE Service SET EnableClusterStatus=true WHERE Name='SelfMonitoring';
   ```
4. Enter the following command at the command prompt to start the self-monitoring service:

   ```
   UPDATE Service SET Running=true where Name='SelfMonitoring';
   ```

Cluster status events

The self-monitoring service sends cluster status events to the ObjectServer under the following conditions.

A secondary cluster member becomes unavailable
If the primary cluster member detects that a secondary server has become unavailable, the self-monitoring service running in the primary server sends an event to the ObjectServer with a severity of 3.

A secondary cluster member assumes the primary role
If a secondary cluster member assumes the primary role, its self-monitoring service sends an event to the ObjectServer with a severity of 1.

A secondary cluster member detects that the primary has become unavailable
If a secondary cluster member detects that the primary server has become unavailable, its self-monitoring service sends an event to the ObjectServer with a severity of 5.

A secondary cluster member accepts a new primary server
If a secondary cluster member accepts another member of the cluster as the new primary server, its self-monitoring service sends an event to the ObjectServer with a severity of 1.

A server is going down
For example, if it is a single server setup and self monitoring (with cluster monitoring) is running, just before the server is shutdown, it will send a event that it is about to shutdown. The message will be of the following format:

`server_name on host_name is about to shutdown`

If the server is the primary server in the cluster, the message would be:

`Primary server server_name on host_name is about to shutdown`

Cluster status event fields

This topic provides description of the fields in cluster status events sent to the ObjectServer.

Table 20. Fields in cluster status events sent to the ObjectServer

Field	Description
`Identifier`	`Impact Cluster Status for server`, where *server* is of the *servername:clustername* format. For example: `NCI1:NCLUSTER`.
`Node`	Host name of the system where the Impact Server is running.
`NodeAlias`	IP address of the system where the Impact Server is running.
`Severity`	Severity as described in "Cluster status events" on page 105.
`Summary`	Detailed information about the cluster status, as described in Table 21 on page 107.
`Class`	`10500`
`Type`	`13`
`AlertGroup`	`ClusterStatus`
`FirstOccurrence`	Timestamp for the first occurrence of this event.
`LastOccurrence`	Timestamp for the most recent occurrence of this event.
`Manager`	`Impact SelfMonitoring@<clustername>`. For example: `Impact SelfMonitoring@NCICLUSTER`.
`Agent`	`Impact 6.1 SelfMonitoring`

This table shows the cluster status information sent as the contents of the Summary field.

Table 21. Cluster status information sent as the contents of the Summary field

Condition	Summary Field
Secondary cluster member unavailable	Secondary Server *server_name* on *hostname* has gone down, where *server_name* is the name of the Impact Server instance, and *hostname* is the host information about the server.
Secondary cluster member assumes the primary role	*server_name* is the current Primary Server on *hostname*, where *server_name* is the name of the server instance, and *hostname* is the host information about the server.
Secondary cluster member detects that the primary has become unavailable	Secondary Server *secondary_name* reporting that Primary Server *primary_name* on *hostname* has gone down, where *secondary_name* is the name of the secondary server instance, *primary_name* is the name of the primary server instance, and *hostname* is the host information about the server.
Secondary cluster member accepts a new primary	Secondary Server *secondary_name* acknowledging *primary_name* on *hostname* as the new Primary Server, where *secondary_name* is the name of the secondary server instance, *primary_name* is the name of the primary server instance, and *hostname* is the host information about the server.

Checking if cluster status monitoring is enabled

To check to see if cluster status monitoring is enabled, enter the following command at the CLI prompt.

Procedure

```
SELECT IsClusterStatusEnabled FROM Service WHERE Name = 'SelfMonitoring';
```

The CLI returns a value of `true` if cluster status monitoring is enabled. Otherwise, the CLI returns a value of `false`.

Enabling cluster status monitoring

To enable cluster status monitoring, enter the following command at the CLI prompt.

About this task

- `UPDATE Service SET EnableClusterStatus = true WHERE Name = 'SelfMonitoring';`

Attention: By default, cluster monitoring events are not deduplicated, which can result in a large number of events being created in the ObjectServer table. To prevent that, enable deduplicating for cluster monitoring events. For more information, see "Enabling deduplicating of cluster monitoring events" on page 108.

Disabling cluster status monitoring

To disable cluster status monitoring, enter the following command at the CLI prompt.

Procedure

```
UPDATE Service SET EnableClusterStatus = false WHERE Name =
'SelfMonitoring';
```

Viewing the current cluster status

To view the current cluster status, enter the following command at the CLI prompt.

Procedure

```
SELECT ClusterStatus FROM Service WHERE Name = 'SelfMonitoring';
```

Viewing the cluster status history

To view the cluster status history, enter the following command at the CLI prompt.

Procedure

```
SELECT ClusterStatusHistory FROM Service WHERE Name = 'SelfMonitoring';
```

Enabling deduplicating of cluster monitoring events

Use this procedure to enable deduplicating of cluster monitoring events.

Procedure

1. Set the `impact.selfmonitoring.clusterdeduplication=true` property in the `$IMPACT_HOME/etc/<servername>_selfmonitoring.props` file.

   <servername> is the name of your Impact Server. You must edit this property for each Impact Server in the cluster.
2. Restart the Impact Server to apply the change.

Service status monitoring

Service status monitoring is a process in which the self-monitoring service sends events to the Netcool/OMNIbus ObjectServer that report the status of services in an Impact Server.

Service status events

The self-monitoring service sends Service status events to the ObjectServer under the following conditions.

When a service is started in the server

When a Service is started, either through the GUI or Command Line, Self Monitoring would send a Service Status event indicating the Name of the Service that got started. This event would have Severity = 1. Note that this notification is sent for services started after Self Monitoring (with Service Monitoring enabled) is running. So certain cases like Auto startup of Services during Server initialization will not be reported.

When a service is stopped in the server

When a currently running Service is stopped, either through the GUI or Command Line, Self Monitoring would send a Service status event indicating the name of the Service that was stopped. This event would have a Severity of 4.

Self monitoring service configuration window

Use this information to configure the self monitoring service.

Table 22. Self Monitoring Service window

Window element	Description
ObjectServer Data Source	Select the ObjectServer that you want to use to send events.
Memory Status: Enable	Select to send status events regarding queue conditions of the event readers that are currently running.
Interval	Select or type how often the service must send status events to the ObjectServer.
Deduplication	Deduplication is enabled by default. See the Netcool/OMNIbus library for information about deduplication of events.
Queue Status: Enable	Select to enable the service to send events about the status of the event readers and listeners currently running.
Interval	Select or type (in seconds) how often the service must send queue status events.
Deduplication	Deduplication is enabled by default. See the Netcool/OMNIbus library for information about deduplication of events.
Cluster Status: Enable	Select to enable the service to send events about the status of the cluster to which it belongs. It sends events when: • A Impact Server is started and joins the cluster • A server is stopped and removed from the cluster • A primary server is down and a secondary server becomes the new primary
Data Source Status: Enable	Select to enable the service to send the status when certain conditions occur with a data source. For example, the service sends a status message when a user tests a connection to a data source or when a connection cannot be established. **Remember:** Restart the service to apply the change.
Service Status: Enable	To enable service monitoring, select this check box and start the self-monitoring service. The self-monitoring service sends service status events to the ObjectServer. **Remember:** Restart the service to apply the change.
Startup: Automatically when server starts	Select to automatically start the service when the server starts. You can also start and stop the service from the GUI.
Service log: Write to file	Select to write log information to a file.

Deduplication of service monitoring events

By default, service monitoring events will not get deduplicated.

Procedure

- To turn on deduplication, perform one of the following actions:
  - Set the following property in the `$IMPACT_HOME/etc/`*`servername`*`_selfmonitoring.props` file:

    ```
    impact.selfmonitoring.servicededuplication=true
    ```

    Restart the server so that it picks up the change.

- Issue the following command from the command line service:
  ```
  update Service set ServiceDeduplication=true where Name='SelfMonitoring';
  ```
  Restart the service to apply the change.

- To turn off deduplication (default setting), perform one of the following actions:
  - Set the following property in the $IMPACT_HOME/etc/*servername*_selfmonitoring.props file:
    ```
    impact.selfmonitoring.servicededuplication=false
    ```
    Restart the server so that it picks up the change.
  - Issue the following command from the command line service:
    ```
    update Service set ServiceDeduplication=false where Name='SelfMonitoring';
    ```
    Restart the service to apply the change.
- To check if service deduplication is enabled issue the following command:
  ```
  select ServiceDeduplication from Service where Name='SelfMonitoring';
  ```

Viewing the total time a service ran for service monitoring

You can view the information about the duration for which the service ran for service monitoring.

This is only applicable for the event that is sent when a service is stopped. The summary contains the name of the service, the name of the server, and the time for which the service ran. The summary information has the following format:

servicename Stopped on Impact Server *servername* Duration: *time*

where, *time* will be of the format *hours, minutes, seconds*. For example, 1h 23m 33s, or 23m 33s (if the service ran for less than a hour), or 33s (if the service ran for less than a minute).

If deduplication is disabled, the identifier for the service monitoring event has one of the following format:

Impact Service Status for *Service Name*:*ServerName*:*ClusterName* at date_time

If deduplication is enabled, it will have the following format:

Impact Service Status for *Service Name*:*ServerName*:*ClusterName*

Service status event fields

This topic describes the fields in data source status events sent to the ObjectServer.

Table 23. Fields in service status event sent to the ObjectServer

Field	Description
Identifier	Impact Service Status for *server*, or Impact Service Status for *server* at <*timestamp value*>, where *server* is of the *servername:clustername* format. For example: NCI1:NCLUSTER.
Node	Host name of the system where the Impact Server is running.
NodeAlias	IP address of the system where the Impact Server is running.
Severity	Severity as described in "Service status events" on page 108.
Summary	Name of the Service indicating if it is started or stopped.
Class	10500
Type	13
AlertGroup	ServiceStatus

Table 23. Fields in service status event sent to the ObjectServer (continued)

Field	Description
FirstOccurrence	Timestamp for the first occurrence of this event.
LastOccurrence	Timestamp for the most recent occurrence of this event.
Manager	Impact SelfMonitoring@<*clustername*>. For example: Impact SelfMonitoring@NCICLUSTER.
Agent	Impact 6.1 SelfMonitoring

Checking if service status monitoring is enabled

To check to see if Service status monitoring is enabled, enter the following command at the CLI prompt:

Procedure

```
SELECT IsServiceStatusEnabled FROM Service WHERE Name = 'SelfMonitoring';
```

The CLI returns a value of true if Service status monitoring is enabled. Otherwise, the CLI returns a value of false.

Enabling service status monitoring

To enable Service status monitoring, enter the following command at the CLI prompt:

Procedure

```
UPDATE Service SET EnableServiceStatus = true WHERE Name =
'SelfMonitoring';
```

Remember: Restart the service to apply the change.

Disabling service status monitoring

To disable Service status monitoring, enter the following command at the CLI prompt:

Procedure

```
UPDATE Service SET EnableServiceStatus = false WHERE Name =
'SelfMonitoring';
```

Remember: Restart the service to apply the change.

Viewing the current service status

To view the current Service status, enter the following command at the CLI prompt:

Procedure

```
SELECT ServiceStatus FROM Service WHERE Name = 'SelfMonitoring';
```

Viewing the service status history

To view the Service status history, enter the following command at the CLI prompt:

Procedure

```
SELECT ServiceStatusHistory FROM Service WHERE Name = 'SelfMonitoring';
```

Chapter 10. Secure communication

In Netcool/Impact there are several interfaces that are available over nonsecure TCP channels.

These interfaces are utilized by the GUI and Impact servers to acquire data, update configuration, keep primary and secondary servers in sync, etc. Most of the securing of Impact can be accomplished via the setting of a single property within the server.props file. Establishing connections between components is one step towards a more secure environment. However, to provide the most secure environment possible, interfaces still available over non-secure ports should be disabled.

Setting up SSL communication

Use this procedure to enable SSL communication in Netcool/Impact.

About this task

For this procedure we assume a cluster with two nodes:

- Node 1 - NCI1, with a GUI Server, and Impact Server+Nameserver installed. This will be the primary node.
- Node 2 - NCI2, also with a GUI Server, and Impact Server+Nameserver installed.

Procedure

1. Optional: Generate a server certificate.

   For more information see, “Generating a self signed certificate” on page 118.
2. Optional: Add the certificate to the application server trust store.

   For more information, see “Adding a certificate to the application server trust store” on page 119.
3. Enable SSL on all deployment components on both cluster nodes:
   a. Impact Servers

      For more information, see “Enabling SSL on the Impact Server” on page 114.
   b. GUI Servers

      For more information, see “Enabling SSL on the GUI Server” on page 114
   c. Nameservers

      For more information, see “Enabling SSL on the Name Server” on page 115
4. Exchange certificates between all the cluster components.

   For secure communications to occur between servers, the security certificates of the servers must be exchanged so that a secure connection can be established. The trust store of each server must contain the signer's certificate of the other. Certificates can be imported into the server trust store using the WebSphere security UI or by using the wsadmin command. In this cluster configuration you will exchange certificates between the following server pairs:
   - from the secondary Impact Server+Nameserver to the primary GUI Server
   - from the primary Impact Server+Nameserver to the secondary GUI Server

- from the secondary Impact Server+Nameserver to the primary Impact Server+Nameserver
- from the primary Impact Server+Nameserver to the secondary Impact Server+Nameserver

For more information, see "Exchanging certificates between the GUI Server and the Impact Server" on page 115, and "Exchanging certificates between Impact Servers" on page 116.

5. Disable the non-secure ports on all nodes in the cluster.

   For more information, see "Disabling nonsecure interfaces" on page 117.

6. Enable SSL for the command line interface.

   For more information, see "Enabling SSL for the command line interface" on page 117.

7. Restart the cluster components.

Enabling SSL on the Impact Server

Use this procedure to enable SSL for the Impact Server.

Procedure

1. Edit the `$IMPACT_HOME/etc/nameserver.props` file:

   ```
   impact.nameserver.ssl_enabled=true
   impact.nameserver.cluster_member.port=port
   ```

   where *cluster_member* is the number of a given Name Server in a cluster, and *port* is the port number of the HTTPS connector. The default is 9081.

   Important: If you are not sure what the current setting is, check the value of the **WC_defaulthost_secure** property in the `$TIP_HOME/profiles/ImpactProfile/properties/portdef.props` file.

2. Optional: Edit the `$IMPACT_HOME/etc/NCI_server.props` file:

   ```
   impact.server.ssl_enabled=true
   ```

   This property enables SSL for internal connections between Impact Servers.

   Important: This property is only required if the GUI Server is deployed separately from the Impact Server. If you install both servers in the same eWAS, then you do not have to set this property.

Enabling SSL on the GUI Server

You need to perform this procedure only if the GUI Server is deployed separately from the Impact Server. If both servers are installed in the same eWAS, then you only need to edit this file once, for the Impact Server.

Procedure

Edit the `$IMPACT_HOME/etc/nameserver.props` file:

```
impact.nameserver.ssl_enabled=true
impact.nameserver.cluster_member.port=port
```

where *cluster_member* is the number of a given Name Server in a cluster, and *port* is the port number of the HTTPS connector. The default is 9081.

Important: If you are not sure what the current setting is, check the value of the **WC_defaulthost_secure** property in the `$TIP_HOME/profiles/ImpactProfile/properties/portdef.props` file.

Enabling SSL on the Name Server

Use this procedure to enable SSL for the Name Server.

About this task

Edit these web.xml files:

- $IMPACT_HOME/nameserver/stage/WEB-INF/web.xml
- $TIP_HOME/profiles/ImpactProfile/installedApps/ImpactCell/nameserver.ear/nameserver.war/WEB-INF/web.xml
- $TIP_HOME/profiles/ImpactProfile/config/cells/ImpactCell/applications/nameserver.ear/deployments/nameserver/nameserver.war/WEB-INF/web.xml

Procedure

1. Open each of web.xml files in a text editor.
2. Find the context-param element that contains the SSL_ENABLED property and set its value to true.

   ```
   <context-param>
       <param-name>SSL_ENABLED</param-name>
       <param-value>true</param-value>
   </context-param>
   ```

3. Find the context-param element that contains the REPLICANT.*cluster_member*.PORT property and update it with the SSL port number. Repeat that for each replicant port element.

   ```
   <context-param>
       <param-name>REPLICANT.0.PORT</param-name>
       <param-value>9081</param-value>
   </context-param>
   ```

   where *cluster_member* is the number of a given Name Server in a cluster, and *port* is the port number of the HTTPS connector. The default is 9081.

   Important: If you are not sure what the current setting is, check the value of the **WC_defaulthost_secure** property in the $TIP_HOME/profiles/ImpactProfile/properties/portdef.props file.

4. Save the updated web.xml files.

Exchanging certificates between the GUI Server and the Impact Server

Use this procedure to exchange certificates between the GUI Server and the Impact Server.

Procedure

1. Log on to TIP at this URL:

   https://*server_address*:16311/ibm/console

   Important: Port 16311 is the default secure port. To make sure what port you are using, check the value of the **WC_defaulthost_secure** property in the $TIP_HOME/profiles/TIPProfile/properties/portdef.props file.

   Log on using the wsadmin username and password, that you provided during the installation, in the eWAS configuration screen.

2. Navigate to the eWAS administrative console page by selecting **Settings** > **WebSphere Administrative Console** > **Launch WebSphere administrative console**.

3. Navigate to the NodeDefaultTrustStore page by selecting **Security** > **SSL certificate and key management** > **Key stores and certificates** > **NodeDefaultTrustStore**.
4. Select **Signer certificates** from the **Additional properties** menu.
5. Click the **Retrieve from port** button to retrieve a certificate from a remote server.

   Provide the required information:

   Host The hostname or IP address of the server from which you want to retrieve the certificate.

   Port The secure port of the server from which you want to retrieve the certificate, by default 9081.

   Alias The name under which you want the retrieved certificate to be displayed in the list of signer certificates, for example “trusted”.
6. Click **Retrieve signer information**.

   If the operation is successful you will see the details of the retrieved certificate in the same screen. Click OK to confirm and to return to the list of signer certificates. You should now see the retrieved certificate in the list. Click **Save** to save the changes.
7. Log out of the console.

Exchanging certificates between Impact Servers

Use this procedure to exchange certificates between Impact Servers.

Procedure

1. Log on to the eWAS administrative console page at this URL:

   `https://server_address:9086/ibm/console`

   9086 is the default port. To make sure what port you are using, look into the `AboutThisProfile.txt` file in the `$IMPACT_PROFILE/logs` directory.
2. Enter the wsadmin username and password.

   You provided these during the installation, in the eWAS configuration screen.
3. Navigate to the NodeDefaultTrustStore page by selecting **Security** > **SSL certificate and key management** > **Key stores and certificates** > **NodeDefaultTrustStore**.
4. Select **Signer certificates** from the **Additional properties** menu.
5. Click the **Retrieve from port** button to retrieve a certificate from a remote server.

   Provide the required information:

   Host The hostname or IP address of the server from which you want to retrieve the certificate

   Port The secure port of the server from which you want to retrieve the certificate, by default 9081.

   Alias The name under which you want the retrieved certificate to be displayed in the list of signer certificates, for example “trusted”.
6. Click **Retrieve signer information**.

   If the operation is successful you will see the details of the retrieved certificate in the same screen. Click OK to confirm and to return to the list of signer certificates. You should now see the retrieved certificate in the list. Click **Save** to save the changes.

7. Log out of the console.

Disabling nonsecure interfaces

To provide the most secure environment you must disable interfaces that are still available over non-secure ports.

Procedure

To disable these interfaces from running on non-secure ports, edit the `server.xml` file by looking for the WCInboundDefault chain element and set the value of the enable attribute to "false".
For example, in `$WAS_HOME/profiles/<profileName>/config/cells/<cellName>/nodes/<nodeName>/servers/<serverName>/server.xml`, to disable HTTP:

1. Locate the transport channel for the endPointName you want to disable. For HTTP, the endPointName is WC_defaulthost.

   ```
   <transportChannels xmi:type="channelservice.channels :TCPInboundChannel"
   xmi:id="TCPInboundChannel_1183077764085" name="TCP_2"
   endPointName="WC_defaulthost"
   maxOpenConnections="20000" inactivityTimeout="60"
   threadPool="ThreadPool_1183077764085"/>
   ```

2. Take note of the id attribute, which in this case is TCPInboundChannel_1183077764085.

   There are two `<chains>` entries that match the TCPInboundChannel_1183077764085 `<transportChannel>`:

   - ```
     <chains xmi:id="Chain_1183077764085"
     name="WCInboundDefault" enable="true"
     transportChannels="TCPInboundChannel_1183077764085
     HTTPInboundChannel_1183077764085
     WebContainerInboundChannel_1183077764085"/>
     ```
   - ```
     <chains xmi:id="Chain_1183077764088"
     name="HttpQueueInboundDefault" enable="true"
     transportChannels="TCPInboundChannel_1183077764085
     HTTPInboundChannel_1183077764085
     HTTPQueueInboundChannel_1183077764084
     WebContainerInboundChannel_1183077764088"/>
     ```

   Change the value of the **`enable`** attribute to "false" for both these entries, and the port will be disabled the next time the server is started.

Enabling SSL for the command line interface

To use SSL from the command line interface, you need to set the SSL_ENABLED environment variable from your shell.

The value of this variable is not important, it simply needs to be set to trigger secure communication between the command line interface and the eWAS server.

The method of setting this environment variable depends on what operating system you are running.

Enabling SSL communication with the ObjectServer

You can set up secure communication between Netcool/Impact and the ObjectServer.

Procedure

1. Optional: Create a certificate, add it to the ObjectServer trust store, and configure the ObjectServer for SSL communication. This step is not required if the ObjectServer is already configured to run in SSL mode.
2. Import the ObjectServer certificate into the Impact Server trust store.

   For more information, see "Exchanging certificates between the GUI Server and the Impact Server" on page 115, and "Exchanging certificates between Impact Servers" on page 116.
3. Enable SSL in the ObjectServer data source by checking the **SSL Mode: Enable** checkbox in the ObjectServer data source editor. For more information, see the *ObjectServer data source configuration window* topic in the *User Interface Guide*.

Generating a self signed certificate

You use the ikeyman tool to generate your server certificate. The tool is located in the $TIP_HOME/bin directory.

Procedure

1. Run ikeyman from the command line. The ikeyman editor window opens.
2. Click the new button to open a new key window.
   a. In the **File Name** field, type the name for your new key, for example, `mykey.jks`.
   b. In the **Location** field, point to the directory where you want your new key to be saved, for example, `/tmp`.
   c. Click OK to save the key. A window opens asking you for a password for your new key. Make sure that you use a password of adequate strength. After you confirm the password for your new key, you return to the main ikeyman menu.
3. From the **Key database content** menu, select **Personal Certificates**.
4. Click the **New Self-Signed** button to open a new self signed certificate window.
   a. Provide all the required information for your new certificate, and any optional information that you want to provide. The mandatory fields are:
      - Key label – type in the name for your key, for example, "impact".
      - Version – you can leave the default value, for example, "X509 V3".
      - Key size – use an adequate key size, in bits, for example 1024 bits.
      - Common Name – your name or the name of your server, for example, "ibm.emea.*servername*.com".
      - Validity Period – the time after which the certificate expires, by default 365 days.
   b. Click OK to save your new certificate and return to the main ikeyman menu. Your new certificate displays in the list of personal certificates. For example, it displays as "impact", if that is what you typed in the **Key label** field.
5. Select your certificate and click the "Extract Certificate" button. In the window that opens add the following information:
   - Data type – the type of certificate encoding, "Base64-encoded ASCII data" is the preferred choice.
   - Certificate file name – the physical name of your certificate file, for example, `mycertificate.arm`.

- Location – a directory to which you want to export your new certificate, for example `/tmp`.

Click OK to export the certificate. You will get a confirmation that the export was successful and will return to the main ikeyman menu.

6. Make sure that your certificate was saved to disk and close the ikeyman editor.

Adding a certificate to the application server trust store

If you created a server certificate, you must add it to the WebSphere application server (eWAS) trust store.

Procedure

1. Copy your signed certificate to the directory that holds your application server trust store:
   - GUI Server - `$TIP_HOME/tipv2/profiles/TIPProfile/etc`
   - Impact Server - `$TIP_HOME/profiles/ImpactProfile/etc`
2. Log on to the application server administration console.
   - GUI Server - log on to the console using the TIP interface. For more information, see "Exchanging certificates between the GUI Server and the Impact Server" on page 115.
   - Impact Server - Log on to the console at this URL:

     `https://server_address:9086/ibm/console`

     For more information, see "Exchanging certificates between Impact Servers" on page 116.
3. In the navigation panel, select **Security** > **SSL certificate and key management**. The SSL certificate and key management page is displayed.
4. Under **Related Items**, click **Key stores and certificates**, and then select **NodeDefaultTrustStore**. The **General Properties** page for NodeDefaultTrustStore is displayed.
5. Under **Additional Properties**, click **Signer certificates**. A table that includes a list of signer certificates is displayed.
6. In the table, click the **Add** button to add your signed certificate. Provide the following information:
   a. In the **Alias** field, type "trusted".
   b. In the **File name** field, type the name of your certificate that you copied in step 1 of this section, for example, `mycertificate.arm`.
   c. From the **Data type** list, select "Base64–encoded ASCII data".
   d. Click OK to add the new certificate to the list.
7. Log out of the administration console, and restart the server, to which you added the certificate.

Configuring SSL to compile a WSDL file

Before you can use Secure Socket Layer (SSL) communication to compile a Web Services Description Language (WSDL) file, you must complete the following steps.

Procedure

1. To create a trust store and a key store, use the key management utility (iKeyman) tool that is in the `<install_dir>/bin` directory.

For example, if you installed Netcool/Impact on a UNIX operating system, the directory is `/opt/IBM/tivoli/impact/tipv2/bin/ikeyman.sh`

To create a truststore and a keystore:

a. Create the following folders:
   - `C:\SSLTrustStore`
   - `C:\SSLKeyStore`

b. Double click the `ikeyman.bat/sh` file to run the key management utility (iKeyman) tool.

c. To create the truststore:
   - Click **Key Database File** > **New**.
   - Select the key database type. For example, **PKCS12** and **JKS** are possible values.
   - Enter a name for the file.
   - Enter `C:\SSLTrustStore` in the **Location** field.
   - When prompted, enter the password.

d. To create the keystore:
   - Click **Key Database File** > **New**.
   - Select the key database type. For example, **PKCS12** and **JKS** are possible values.
   - Enter a name for the file.
   - Enter `C:\SSLKeyStore` in the **Location** field.
   - When prompted, enter the password.

2. Import the SSL certificate from the Web Services Server to the truststore that you created in the previous step.
3. Copy the `samplewsdl2java.ssl.props` file to the directory where you want to store the file.
4. Configure the truststore properties as follows:

   a. To add the truststore to the properties file, add the following property:

      `javax.net.ssl.trustStore=<truststorekeystoredirectory>/wsdltruststore.p12`

      where *<truststorekeystoredirectory>* is the directory path for the folder where you saved the truststore and keystore files.

      For example, if you save the truststore and keystore files to the `/opt/IBM/tivoli/impact` directory, add the following property:

      `javax.net.ssl.trustStore=/opt/IBM/tivoli/impact/wsdltruststore.p12`

   b. To add the password, change the following property:

      `truststore.password=<password>`

      The first time that you enable SSL for a WSDL file, you must set the password. If you change the password, you must change this property. When the WSDL tool runs, the property is removed. Netcool/Impact stores an encrypted version of the password for further use.

   c. To specify the truststore type, change the following property:

      `javax.net.ssl.trustStoreType=<truststoretype>`

      `PKCS12` and `JKS` are examples of valid values for this field. For example, add the following property to specify `PKCS12` as the truststore:

      `javax.net.ssl.trustStoreType=PKCS12`

5. Configure the keystore properties as follows:

a. To specify the keystore, add the following property:

   `javax.net.ssl.keyStore=<truststorekeystoredirectory>/wsdltruststore.p12`

   where *<truststorekeystoredirectory>* is the directory path for the folder where you saved the truststore and keystore files.

   For example, if you save the truststore and keystore files to the `/opt/IBM/tivoli/impact` directory, add the following property:

   `javax.net.ssl.keyStore=/opt/IBM/tivoli/impact/wsdltruststore.p12`

b. To specify the password for the keystore, change the following property:

   `keystore.password=<password>`

   The first time that you enable SSL for a WSDL file, you must set the password. If you change the password, you must change this property. When the WSDL tool runs, the property is removed. Netcool/Impact stores an encrypted version of the password for further use.

c. To specify the type of keystore, change the following property:

   `javax.net.ssl.keyStoreType=<keystoretype>`

   `PKCS12` and `JKS` are valid values for this field. For example, add the following property to specify `PKCS12` as the keystore type:

   `javax.net.ssl.keyStoreType=PKCS12`

6. Remove the comment from the following line in the `nci_compilewsdl` file that is in the `impact/bin` directory:

   ```
   #
   #IMPACT_SSL_PROPS=${IMPACT_HOME}/wsdl2java.ssl.props export IMPACT_SSL_PROPS
   #
   ```

7. Copy the `samplewsdl2java.ssl.props` file from the `$IMPACT_HOME/etc` directory to `$IMPACT_HOME` directory.
8. Rename the `samplewsdl2java.ssl.props` file to `wsdl2java.ssl.props` and save.

Results

You can use the Web Services Wizard or the `NCI_Wizard` command to import a WSDL file from an https session.

Chapter 11. Command-line tools

This chapter contains information about the Tivoli Netcool/Impact command-line tools.

nci_crypt

The `nci_crypt` tool encrypts a string using the Netcool/Impact encryption algorithm.

You can use this tool to encrypt passwords passed from the command line with `nci_trigger`. This tool is located in the `$IMPACT_HOME/bin` directory.

To run this tool, use the following syntax:

```
nci_crypt string
```

where *string* is the string you want to encrypt.

Important: Enclose a string containing a space within double quotes ("). Do not use single quotes (') for they will be interpreted incorrectly by the `nci_crypt` tool.

nci_export

The `nci_export` tool exports data source, data type, service, policy, and project information from an instance of the Impact Server to a specified directory.

This tool is in the `$IMPACT_HOME/bin` directory.

Run the script using the following syntax:

```
nci_export arguments options
```

You can only use the `nci_export` tool to export data from a primary server in a cluster. You cannot use it to export data from secondary cluster members. If you do try to export data from a secondary cluster member, Netcool/Impact displays an error and creates an entry in the log file.

Important: Make sure that the instance of the Impact Server that contains the data you are exporting is running before you use `nci_export`.

Arguments

server name
: The name of the server instance whose data you want to export.

directory
: The directory where you want to store the exported data.

--project
: This option is followed by the name of the project whose data you want to export. Use this option to export a single project.

Important: The following restrictions on the `nci_export` usage apply, when Netcool/Impact is running as part of a TBSM installation:

- Do not run it without a "--project" argument, if you then import the data into another TBSM 6.1 data server.
- Do not run it with a "--project TBSM_BASE" argument, if you then import the data into another TBSM 6.1 data server.

Examples

Use this command to export all data from the NCI instance to the `/tmp/NCI_export` directory:

```
./nci_export NCI /tmp/NCI_export
```

Use this command to export the data from the F00 project only on the NCI instance to the `/tmp/NCI_export` directory:

```
./nci_export NCI --project F00 /tmp/NCI_export
```

nci_import

The `nci_import` tool imports data that was previously exported from an instance of the Impact Server.

This data includes all data sources, data types, policies, and services currently defined in the server instance. This tool is located in the `$IMPACT_HOME/bin` directory.

Important: You must run the script from the primary server. If you try to run the script from a secondary server, the script fails.

Note: Make sure that the target instance of the Impact Server is running before you use `nci_import`.

To run this tool, use the following syntax:

```
nci_import server_name directory
```

where *server_name* is the name of the server instance where you want to import the data and *directory* is the location that contains data exported using nci_export.

If Netcool/Impact is running in a cluster setup, it is recommended to shutdown all secondary servers and have only primary server running before running nci_import. After nci_import is completed successfully and there are no locks in the primary, the secondary servers can be started and they will replicate the configuration from the primary.

nci_trigger

The `nci_trigger` tool allows you to start a policy from the command line.

This tool is located in the `$IMPACT_HOME/bin` directory. To run this tool, use the following syntax

```
nci_trigger [-version] | server_name [ user_id | user_id/password |
      -e/user_id/encrypted_password | NULL ] policy_name field value field value ...
```

Note: If no password is defined for the user, you must specify the password as `NULL` when you run the command.

These are the command line arguments for `nci_trigger`:

-version
Causes nci_trigger to print the Netcool/Impact version number, platform, and command syntax to standard output and then exit.

server_name
Instance of the Impact Server where you want the policy to run.

user_id/password
Username and password of a valid user who has access to Netcool/Impact.

-e/*user_id/encrypted_password*
Username and encrypted password of a valid user who has access to Netcool/Impact. You can encrypt passwords using the nci_crypt tool.

policy_name
Name of the policy to run.

field Name of a field in the event container passed to the policy. Optional.

value Value of a field in the event container passed to the policy. Optional.

Note: If you want to run a policy that contains a call to the ReturnEvent function, you must include Identifier and Serial fields in the event container passed to policy.

Runtime parameters

The following example shows how to to pass runtime parameters to a policy, using nci_trigger. In this example, the policy is named POLICY_01

```
Log(EventContainer.Value1);
Log(EventContainer.Value2);
```

To run this policy using nci_trigger, you can enter the following at a command prompt:

```
nci_trigger NCI tipadmin/netcool POLICY_01 Value1 Testing1 Value2 Testing2
```

This prints the following to the policy log:

```
Testing1
Testing2
```

Examples

This example shows how to run a simple policy from the command line that does not process an incoming event. In this example, the policy is named POLICY_01, the user is admin, the password is netcool and the server instance is NCI.

```
nci_trigger NCI tipadmin/netcool POLICY_01
```

This example shows how to run a policy using an encrypted password. In this example, the password was previously encrypted using the nci_crypt tool.

```
nci_trigger NCI_02 -e/admin/7E6C7364EFD7CD69 POLICY_02
```

This example shows how to run a policy and pass event field values to the policy as the contents of an incoming event container.

```
nci_trigger NCI admin/netcool POLICY_03 Node host_01 Summary Node_down
AlertKey host_01Node_down
```

nci_policy

You use the nci_policy script to do various command line tasks with your policies.

For example, you can check the syntax of a policy that you created in an editor other than the integrated policy editor. Like other command line scripts it is located in the $IMPACT_HOME/bin directory.

Syntax

```
nci_policy [SERVER_NAME] [COMMAND] [ARGUMENTS]
```

Parameters

SERVER_NAME
Name of the Impact Server instance.

COMMAND
Unless you use the **--help** or **--h** option, one of the following commands must be present:

- **syntax** - This command is used to check the syntax of a policy. It requires arguments.
- **push** - This command is used to upload or update a policy. It requires arguments.

ARGUMENTS
Arguments to use with the **syntax** or **push** command.

Example

Use this command to check the syntax of the policy.ipl policy on the NCI1 server instance:

```
nci_policy NCI1 syntax /path/to/policy.ipl
```

Use this command to upload the policy.ipl policy to project MyProject on the NCI1 server instance:

```
nci_policy NCI1 push user password /path/to/policy.ipl --project MyProject
```

Replace the *user* and *password* with the values that you use to log on to the server instance.

nci_removeserver

You use the nci_removeserver script to remove an Impact instance from the application server.

The script has the following syntax:

```
nci_removeserver instance
```

where *instance* is the Impact instance name.

This is an example of using the nci_removeserver script on Windows to remove the Impact instance named NCI:

```
nci_removeserver.bat NCI
```

Important: After running the `nci_removeserver` tool you must remove the configuration files used by the instance that you removed.

- If you are using Impact Subversion for version control, run the `nci_svn_remove` script.
- If you are using a different source control system, ask the administrator to remove the related configurations files from the source control system.

You must not run the `nci_svn_remove` script before removing the Impact Server instance from the application server.

Using the command-line manager

The command-line manager service tool allows you to access the Impact Server from the command-line interface to start and stop services as well as configure their parameters.

About this task

To configure the command-line manager service, you specify the port to which you are connecting to. You can also specify whether you want the service to start automatically when the Impact Server starts. Follow this procedure to connect to the command-line manager.

Procedure

1. Start the command-line interface by opening the following address in any telnet application:

   ```
   telnet <hostname> <commandlineport>
   ```

   where *hostname* is the name of the system where the Impact Server is running and *port* is the command-line port. The default port is 2000 (as specified during the installation). To find the port number used by the Command Line Service check its configuration in the Command Line Manager service (it is stored in the configuration file `$IMPACT_HOME/impact/etc/<servername>_commandlinemanager.props`). For more information about configuring the service, see *Command Line Manager service* in the User Interface Guide.
2. When prompted for the user name and password use the same user name and password as you would use to log on in the GUI. The default user name and password are `admin` and `netcool`.

Results

After you have logged in to the command-line service, you can work with services using specialized commands. For information about using JavaScript commands see "JavaScript in the command-line manager" on page 128.

Example

You can use certain commands with all services. For example, to check if a service is running, you can use the following command:

```
Select Running from Service where Name = '<servicename>';
```

So for OMNIbusEventReader, it would be `Select Running from Service where Name = 'OMNIbusEventReader';`.

The following list has more examples of the commands that you can use with all services:

- `Select Standby from Service where Name = '<servicename>';` - use this command to check if the service is in standby mode (applicable to secondary members in a cluster).
- `Select Status from Service where Name = '<servicename>';` - this command returns the status of the service.
- `Select Log from Service where Name = '<servicename>';` - use this command to display the log message for the service.

You can start and stop a service from the command line. Note, that this is not applicable to all services since you cannot stop some services, like, for example, the Command Line Manager and Policy Logger service. Use the following command to stop a service:

```
Update Service set Running = FALSE where Name = '<servicename>';
```

To start a service use this command:

```
Update Service set Running= TRUE where Name = '<servicename>';
```

A service specific command, for example, would be:

```
Update Service set EnableMemoryStatus=true where Name='SelfMonitoring';
```

Issue this command to enable memory monitoring.

What to do next

If you change the port number in the primary server in a cluster, it does not affect the port where the secondary server is running. Change these settings on the secondary server using the GUI.

JavaScript in the command-line manager

Use the following commands to switch from IPL commands to JavaScript commands in the command-line manger.

Issue the `js<enter>` command to switch to JavaScript command mode in the command line.

Issue the `script` command to display the current script buffer, once you are in JavaScript command mode.

Issue the `clear` command to clear out the current script buffer. The **`clear`** command only works in JavaScript command mode.

When the command-line manager is in JavaScript mode, you can run any commands that are listed in the *Policy Reference Guide*.

Issue the `exit` command to close the command line manager and exit JavaScript command mode. The only way to exit JavaScript command mode is to use the exit command to close the command line manager. After you exit the command line manager, you must log in again. When you do, the command line manager is once again in IPL mode.

Event Processor commands

The event processor is the service responsible for managing events coming into Netcool/Impact.

You can use the following commands in the command line interface to work with the event processor service:

Select NumEventsInQueue from Service where Name='EventProcessor';
Gets the number of events in the queue.

Select Events from Service where Name='EventProcessor';
Gets the events waiting to be processed.

Select ProcessingStatusHistory from Service where Name='EventProcessor';
Gets the processing status history.

Update Service set ClearQueue=true where Name='EventProcessor';
Clears the event queue.

Select NumThreads from Service where Name='EventProcessor';
Gets the number of processing threads.

Update Service set NumThreads=<numthreads> where Name='EventProcessor';
Sets the number of processing threads. For example:

```
Update Service set NumThreads=4 where Name='EventProcessor';
```

Select PoolConfig from Service where Name='EventProcessor';
Gets thread and connection pool information for diagnosis and makes recommendations.

Thread Analysis Example:

```
STARTRECORD
PoolConfig
----------------------
---------------
THREAD ANALYSIS
---------------

Max Processing Threads: 15
Min Processing Threads: 2
Ratio of Max/Min Threads: 7.5

Recommendation: The difference between the Max and Min Processing Threads
is too large. Consider increasing the Min Processing Threads to 10.
Please note that you will have to restart the EventProcessor service
for the change to take affect.
```

Connection Pool Analysis Example:

```
PASS:

defaultobjectserver

ACTION REQUIRED

For the following SQL DataSource(s) shown below -
the Connection Pool size is lower than the Maximum Threads in the EventProcessor (15)

DB2DS - Connection Pool size: 5

myflatfile - Connection Pool size: 5

MSSQLDS - Connection Pool size: 5

OracleDS - Connection Pool size: 5

EIC_alertsdb - Connection Pool size: 5
```

```
EventrulesDB - Connection Pool size: 2

SCR_DB - Connection Pool size: 5

ReportsHSQLDB - Connection Pool size: 5


Recommendation: Consider increasing the size of the Connection Pool to be >= 15
for the Data Sources where performance is of concern.
Please make sure your database can handle the potential increase in load
since increasing connection pool size allows Impact to make more connections
 to your database.

ENDRECORD

READY
```

Select MinNumThreads from Service where Name='EventProcessor';
Gets the minimum number of processing threads. This command is specific to release 5.1.

Update Service set MinNumThreads = <minnumthreads> where Name='EventProcessor';
Sets the minimum number of processing threads. For example: `Update Service set MinNumThreads = 10 where Name = 'EventProcessor';` This command is specific to release 5.1.

Update Service set MaxNumThreads =< maxnumthreads> where Name='EventProcessor';
Sets the maximum number of processing threads. For example: `Update Service set MaxNumThreads = 50 where Name = 'EventProcessor';` This command is specific to release 5.1.

Update Service set MaximumThroughput = True where Name = 'EventProcessor';
Sets up the service to achieve maximum throughput. This command is specific to release 5.1.

Update Service set SaveTuningConfig = True where Name = 'EventProcessor';
Saves the tuned configuration for the next restart. This command is specific to release 5.1.

Select NumEventsPerQuery from Service where Name='EventProcessor';
Gets the block size of the events fetched. This command is specific to releases before 5.1.

Update Service set NumEventsPerQuery=<num> where Name='EventProcessor';
Sets the block size of the events fetched. For example: `Update Service set NumEventsPerQuery=20 where Name='EventProcessor';` This command is specific to releases before 5.1.

Select EventFetchRate from Service where Name='EventProcessor';
Gets the interval at which the service asks for events. This command is specific to releases before 5.1.

Update Service set EventFetchRate=<rate-in-milliseconds> where Name='EventProcessor';
Sets the interval at which the service asks for events. For example: `Update Service set EventFetchRate=5000 where Name='EventProcessor';` This command is specific to releases before 5.1.

For more information about the Event Processor, see *Event processor service* in the User Interface Guide. For more information about the command-line interface, see "Using the command-line manager" on page 127.

Email Reader commands

The e-mail reader service polls a specified host for e-mail messages. It reads e-mail from a mailbox at intervals that you define when you create an e-mail reader service.

This service is commonly used in escalation and notification policies to look for responses to e-mail notifications sent out by Netcool/Impact.

If the number of emails waiting to be read from the email reader service is more than 25, the timeout value increases automatically. When the number of emails waiting to be read returns to less than 25. The timeout value is reset to the default value or the value specified in the service property file.

You can use the following commands in the command-line interface to work with the e-mail reader service:

Select Host from Service where Name='DefaultEmailReader';
Gets the host.

Update Service set Host='<host>' where Name='DefaultEmailReader';
Sets the host.

Select UserName from Service where Name='DefaultEmailReader';
Gets the user name.

Update Service set UserName='<new_user_name>' where Name='DefaultEmailReader';
Sets the user name.

Select PollInterval from Service where Name='DefaultEmailReader';
Gets the poll interval.

Update Service set PollInterval=<newpollinterval> where Name='DefaultEmailReader';
Sets the poll interval.

Select Policy from Service where Name='DefaultEmailReader';
Gets the policy name.

Update Service set Policy='<policyname>' where Name='DefaultEmailReader';
Sets the policy to which e-mails are to be sent.

For the sake of examples, it is assumed that the name of the Email Reader instance is DefaultEmailReader. Change the syntax if the name of the instance is different.

Email Sender commands

The e-mail sender service lets you configure the local Netcool/Impact e-mail address information so that you can send e-mail notification to users and to other installations of Netcool/Impact

You can use the following commands in the command line interface to work with the e-mail sender service:

Select SMTPHost from Service where Name='EmailSender';
Get the SMTP host.

Update Service set SMTPHost='<smtphost>' where Name='EmailSender';
Sets the SMTP host.

Select SenderAddress from Service where Name='EmailSender';
Gets the sender address.

Update Service set SenderAddress='senderaddress' where Name='EmailSender';
Sets the sender address.

Policy Activator commands

The policy activator service activates policies at the intervals you specify for each policy selected.

You can use the following commands in the command line interface to work with the policy activator service:

Select Interval from Service where Name='DefaultPolicyActivtor';
Gets the interval at which the policy is activated.

Update Service set Interval=<interval> where Name='DefaultPolicyActivator';
Sets the interval at which the policy is activated.

Select Policy from Service where Name='DefaultPolicyActivator';
Gets the policy which this service triggers.

Update Service set Policy='policyname' where Name='DefautPolicyActivator';
Sets the policy that will be activated.

For the sake of examples, it is assumed that the name of the service instance is DefaultPolicyActivator.

Hibernating Policy Activator commands

The hibernating policy activator is the service that is responsible for waking hibernating policies.

You can use the following commands in the command line interface to work with the hibernating policy activator service:

Select Interval from Service where Name='HibernatingPolicyActivtor';
Gets the interval at which the policy is activated.

Update Service set Interval=<interval> where Name='HibernatingPolicyActivator';
Sets the interval at which the policy is activated.

Select WakeProcessImmediately from Service where Name='HibernatingPolicyActivator';
Gets the status of the WakeProcessImmediately flag.

Update Service set WakeProcessImmediately=true where Name='HibernatingPolicyActivator'
Enables the WakeProcessImmediately flag.

Update Service set WakeProcessImmediately=false where Name='HibernatingPolicyActivator';
Disables the WakeProcessImmediately flag.

Update Service set ClearHibernations=true where Name='HibernatingPolicyActivator';
Clears the hibernations.

Policy Logger commands

The policy logger service is responsible for managing the policy log.

The log is a text stream used to record messages generated during the runtime of a policy. You can use the following commands in the command line interface to work with the policy logger service:

Select ErrorHandlingPolicy from Service where Name='PolicyLogger';
Gets the error handling policy.

Update Service set ErrorHandlingPolicy='<policyname>' where Name='PolicyLogger';
Sets the error handling policy.

Select LogLevel from Service where Name='PolicyLogger';
Gets the LogLevel.

Update Service set LogLevel=<loglevel> where Name='PolicyLogger';
Sets the LogLevel.

Select LogAllSQLStatements from Service where Name='PolicyLogger';
Checks if the service logs all SQL statements.

Update Service set LogAllSQLStatements=true where Name='PolicyLogger';
Enables logging of all SQL statements.

Update Service set LogAllSQLStatements=false where Name='PolicyLogger';
Disables logging of all SQL statements.

Select LogPreModuleParams from Service where Name='PolicyLogger';
Checks if the service logs Pre-Execution Action Module parameters.

Update Service set LogPreModuleParams=true where Name='PolicyLogger';
Enables logging of Pre-Execution Action Module parameters.

Update Service set LogPreModuleParams=false where Name='PolicyLogger';
Disables logging of Pre-Execution Action Module parameters.

Select LogPostModuleParams from Service where Name='PolicyLogger';
Checks if the Service logs Post-Execution Action Module parameters.

Update Service set LogPostModuleParams=true where Name='PolicyLogger';
Enables logging of Post-Execution Action Module parameters.

Update Service set LogPostModuleParams=false where Name='PolicyLogger';
Disables logging of Post-Execution Action Module parameters.

Select LogAllModuleParams from Service where Name='PolicyLogger';
Checks if the service logs all parameters.

Update Service set LogAllModuleParams=true where Name='PolicyLogger';
Enables logging of all parameters.

Update Service set LogAllModuleParams=false where Name='PolicyLogger';
Disables logging of all parameters.

Select AppendThreadName from Service where Name='PolicyLogger';
Checks if the service logs ThreadName to logs.

Update Service set AppendThreadName=true where Name='PolicyLogger';
Enables ThreadName logging.

Update Service set AppendThreadName=false where Name='PolicyLogger';
Disables ThreadName logging.

Select AppendPolicyName from Service where Name='PolicyLogger';
Checks if the service logs policy name to logs.

Update Service set AppendPolicyName=true where Name='PolicyLogger';
Enables policy name logging.

Update Service set AppendPolicyName=false where Name='PolicyLogger';
Disables policy name logging.

Select CollectReports from Service where Name='PolicyLogger';
Checks if the service reporting is enabled.

Update Service set CollectReports=true where Name='PolicyLogger';
Enables collecting of reports.

Update Service set CollectReports=false where Name='PolicyLogger';
Disables collecting of reports.

Select PolicyProfiling from Service where Name='PolicyLogger';
Checks if policy profiling is enabled.

Update Service set PolicyProfiling=true where Name='PolicyLogger';
Enables policy profiling.

Update Service set PolicyProfiling=false where Name='PolicyLogger';
Disables policy profiling.

OMNIbus Event Reader commands

The event reader service uses the host and port information of a specified ObjectServer data source so that it can connect to an ObjectServer to poll for new and updated events and store them in a queue.

The event processor service requests events from the event reader. You can use the following commands in the command line interface to work with the OMNIbus event reader service:

Update Service set ClearQueue=true where Name='OMNIbusEventReader';
Clears the event reader queue.

Select QueueSize from Service where Name='OMNIbusEventReader';
Gets the queue size.

Update Service set ClearBuffer=true where Name='OMNIbusEventReader';
Clears the event buffer.

Select BufferSize from Service where Name='OMNIbusEventReader';
Gets the buffer size.

Update Service set ClearState=true where Name='OMNIbusEventReader';
Clears the reader state.

Select StatusEvents from Service where Name='OMNIbusEventReader';
Checks if the service reads the status events inserted from the Self Monitoring Service.

Update Service set StatusEvents=true where Name='OMNIbusEventReader';
Enables reading of status events.

Update Service set StatusEvents=false where Name='OMNIbusEventReader';
Disables reading of status events.

Select LockEvents from Service where Name='OMNIbusEventReader';
Checks if event locking is enabled.

Update Service set LockEvents=true where Name='OMNIbusEventReader';
Enables event locking.

Update Service set LockEvents=false where Name='OMNIbusEventReader';
Disables event locking.

Select LockingExpression from Service where Name='OMNIbusEventReader';
Gets the event locking expression.

Update Service set LockingExpression='<lockingexpression'> where Name='OMNIbusEventReader';
Sets the event locking expression.

Select GetDeletes from Service where Name='OMNIbusEventReader';
Checks if the service reads deleted events.

Update Service set GetDeletes=true where Name='OMNIbusEventReader';
Enables reading of deleted events.

Update Service set GetDeletes=false where Name='OMNIbusEventReader';
Disables reading of deleted events.

Select PolicyForDeletes from Service where Name='OMNIbusEventReader';
Gets the policy which gets triggered for deleted events.

Update Service set PolicyForDeletes='<policyfordeletes>' where Name='OMNIbusEventReader';
Sets the policy where 'Deletes' are sent.

Select OrderByExpression from Service where Name='OMNIbusEventReader';
Gets the OrderByExpression.

Update Service set OrderByExpression='<orderbyexpression>' where Name='OMNIbusEventReader';
Sets the OrderByExpression.

Select CollectReports from Service where Name='OMNIbusEventReader';
Checks if collecting of reports is enabled.

Select GetUpdates from Service where Name='OMNIbusEventReader';
Checks if the Service is configured to get updates.

Update Service set GetUpdates=true where Name='OMNIbusEventReader';
Enables getting updates.

Update Service set GetUpdates=false where Name='OMNIbusEventReader';
Disables getting updates.

Select DataSource from Service where Name='OMNIbusEventReader';
Gets the objectserver data source from which events are read.

Update Service set DataSource='<datasourcename>' where Name='OMNIbusEventReader';
Sets the data source from which events are read.

Select PollingInterval from Service where Name='OMNIbusEventReader';
Gets the polling interval.

Update Service set PollingInterval='<pollingintervalinmilliseconds>' where Name='OMNIbusEventReader';
Sets the polling interval. For example: `Update Service set PollingInterval=4000 where Name='OMNIbusEventReader';`

Select Fields from Service where Name='OMNIbusEventReader';
Gets the fields used in select statements.

Update Service set Fields='<fields>' where Name='OMNIbusEventReader';
Sets the fields used in select statements.

Select ExecuteAllMatches from Service where Name='OMNIbusEventReader';
Checks if the service sends the event to all the policies that pass the filter mapping evaluation.

Update Service set ExecuteAllMatches=true where Name='OMNIbusEventReader';
Sends the event to all the policies whose filter it matches.

Update Service set ExecuteAllMatches=false where Name='OMNIbusEventReader';
Sends the event to the first policy whose filter it matches.

Select NumFilters from Service where Name='OMNIbusEventReader';
Gets the number of filters.

Update Service set NumFilters=<numfilters> where Name='OMNIbusEventReader';
Sets the number of filters.

Select FilterNum<num> from Service where Name='OMNIbusEventReader';
Gets the information on a specific filter. For example, use the `Select FilterNum1 from Service where Name='OMNIbusEventReader';` command to get the information for the first filter. The command will display the Policy, Restriction Filter and whether it is Active for that filter.

Select PolicyNum<num> from Service where Name='OMNIbusEventReader';
Gets the policy for a particular filter.

Update Service set PolicyNum<num>='<policynum>' where Name='OMNIbusEventReader';
Sets the policy for a particular filter.

Select ActiveNum<num> from Service where Name='OMNIbusEventReader';
Checks if a particular filter is active.

Update Service set ActiveNum<num>=true where Name='OMNIbusEventReader';
Makes a particular filter active.

Update Service set ActiveNum<num>=false where Name='OMNIbusEventReader';
Makes a particular filter inactive.

Select RestrictionNum<num> from Service where Name='OMNIbusEventReader';
Gets the restriction clause for a particular filter.

Update Service set RestrictionNum<num>='<restriction>' where Name='OMNIbusEventReader';
Sets the restriction clause for a particular filter.

For the sake of examples, it is assumed that the name of the Service is OMNIbusEventReader (this Service was called DefaultEventReader before Tivoli Netcool/Impact 5.1).

Database Event Reader commands

The database event reader polls an SQL data source at intervals and retrieves rows from a table.

You can use the following commands in the command line interface to work with the database event reader service:

Update Service set ClearQueue=true where Name='DatabaseEventReader';
Clears the EventReader queue.

Select QueueSize from Service where Name='DatabaseEventReader';
Gets the queue size.

Update Service set ClearBuffer=true where Name='DatabaseEventReader';
Clears the event buffer.

Select BufferSize from Service where Name='DatabaseEventReader';
Gets the buffer size.

Update Service set ClearState=true where Name='DatabaseEventReader';
Clears the reader state.

Select GetUpdates from Service where Name='DatabaseEventReader';
Checks if a service is configured to get updates.

Update Service set GetUpdates=true where Name='DatabaseEventReader';
Enables getting updates.

Update Service set GetUpdates=false where Name='DatabaseEventReader';
Disables getting updates.

Select DataSource from Service where Name='DatabaseEventReader';
Gets the data source from which events are read.

Update Service set DataSource='<datasourcename>' where Name='DatabaseEventReader';
Sets the data source from which events are read.

Select PollingInterval from Service where Name='DatabaseEventReader';
Gets the polling interval.

Update Service set PollingInterval='<pollingintervalinmilliseconds>' where Name='DatabaseEventReader';
Sets the polling interval. For example: `Update Service set PollingInterval=4000 where Name='DatabaseEventReader';`

Select Fields from Service where Name='DatabaseEventReader';
Gets the fields used in select statements.

Update Service set Fields='<fields>' where Name='DatabaseEventReader';
Sets the fields used in select statements.

Select ExecuteAllMatches from Service where Name='DatabaseEventReader';
Checks if the service sends event to all the policies which pass the filter mapping evaluation.

Update Service set ExecuteAllMatches=true where Name='DatabaseEventReader';
Sends the event to all the policies whose filter it matches.

Update Service set ExecuteAllMatches=false where Name='DatabaseEventReader';
Sends the event to the first policy whose filter it matches.

Select NumFilters from Service where Name='DatabaseEventReader';
Gets the number of filters.

Update Service set NumFilters=<numfilters> where Name='DatabaseEventReader';
Sets the number of filters.

Select FilterNum<num> from Service where Name='DatabaseEventReader';
Gets information on a specific filter. For example, use the `Select FilterNum1 from Service where Name='DatabaseEventReader';` command to see the information for the first filter. The command will output the Policy, Restriction Filter and whether it is Active for that filter.

Select PolicyNum<num> from Service where Name='DatabaseEventReader';
Gets the policy for a particular filter.

Update Service set PolicyNum<num>='<policynum>' where Name='DatabaseEventReader';
Sets the policy for a particular filter.

Select ActiveNum<num> from Service where Name='DatabaseEventReader';
Checks if particular filter is active.

Update Service set ActiveNum<num>=true where Name='DatabaseEventReader';
Makes a particular filter active.

Update Service set ActiveNum<num>=false where Name='DatabaseEventReader';
Makes a particular filter inactive.

Select RestrictionNum<num> from Service where Name='DatabaseEventReader';
Gets the restriction clause for a particular filter.

Update Service set RestrictionNum<num>='<restriction>' where Name='DatabaseEventReader';
Sets the restriction clause for a particular filter.

Select DataType from Service where Name='DatabaseEventReader';
Gets the data type from which events are read.

Update Service set DataType='<datatypename>' where Name='DatabaseEventReader';
Sets the data type from which events are read.

Select KeyField from Service where Name='DatabaseEventReader';
Gets the Key field used in this service.

Update Service set KeyField='<ieldname>' where Name='DatabaseEventReader';
Sets the Key field.

Select TimeStampField from Service where Name='DatabaseEventReader';
Gets the TimeStamp field.

Update Service set TimeStampField='<imestampfield>' where Name='DatabaseEventReader';
Sets the TimeStamp field.

For the sake of examples, it is assumed that the name of the Service is DatabaseEventReader (this service was called GenericEventReader before Tivoli Netcool/Impact 5.1).

Database Event Listener commands

The database event listener service monitors an Oracle event source for new, updated, and deleted events.

This service works only with Oracle databases. When the service receives the data, it evaluates the event against filters and policies specified for the service and sends the event to the matching policies. You can use the following commands in the command line interface to work with the database event listener service:

Select QueueSize from Service where Name = DatabaseEventListener;
Gets the current queue size.

Update Service set ClearQueue=TRUE where Name = DatabaseEventListener;
Clears the events in the queue.

JMS Message Listener commands

The JMS message listener service is a service that runs a policy in response to incoming messages sent by external JMS message providers.

The message provider can be any other system or application that is capable of sending JMS messages. This section contains JMS message listener service commands that you can use in the command line interface to work with the JMS message listener service.

For more information about the JMS Message Listener see *JMS Message Listener service* in the *User Interface Guide*. For more information about the command line interface, see "Using the command-line manager" on page 127.

Note: The examples assume the name of the service to be JMSMessageListener.

Getting information about the service

Select Running from Service where Name=JMSMessageListener;
Checks if the service is running.

Select QueueSize from Service where Name=JMSMessageListener;
Checks the queue size.

Select Policy from Service where Name=JMSMessageListener;
The policy triggered.

Select DataSource from Service where Name = 'JMSMessageListener;
Data source used.

Select JNDIFactory from Service where Name=JMSMessageListener
Checks the JNDI Factory.

Select ProviderURL from Service where Name=JMSMessageListener;
Checks provider URL.

Select URLPackage from Service where Name=JMSMessageListener;
Checks URL package.

Select ConnectionFactory from Service where Name=JMSMessageListener;
Checks connection factory.

Select Destination from Service where Name=JMSMessageListener;
Checks the destination.

Select MessageSelector from Service where Name=JMSMessageListener;
Checks message selector.

Select IsDurableSubscription from Service where Name=JMSMessageListener;
Checks if durable subscription is used.

Select ClientID from Service where Name=JMSMessageListener;
Gets the client ID.

Select SubscriptionName from Service where Name=JMSMessageListener;
Checks subscription name.

Select Username from Service where Name=JMSMessageListener;
Checks user name.

Changing service configuration

Update Service set Running=true where Name=JMSMessageListener;
Starts the listener service.

Update Service set Running=false where Name=JMSMessageListener;
Stops the listener service.

Update Service set ClearQueue=true where Name=JMSMessageListener;
Clears the message queue.

Update Service set Policy=NewPolicy where Name=JMSMessageListener;
Changes the policy triggered.

Update Service set DataSource = 'NewDataSource where Name=JMSMessageListener;
Changes the JMS data source used with service.

Update Service set MessageSelector = 'NewMessageSelector where Name=JMSMessageListener;
Changes the messages selector used by the listener.

Update Service set DurableSubscription=true where Name=JMSMessageListener;
Enables durable subscription.

Update Service set DurableSubscription=false where Name=JMSMessageListener;
Disables durable subscription.

Update Service set ClientID = 'NewClientID where Name=JMSMessageListener;
Sets the client ID.

Update Service set SubscriptionName=NewName where Name=JMSMessageListener;
Sets the subscription name.

Using WebServices through the command line

Only a privileged user with the bsmAdministrator role can use WebServices on the command line, after being authenticated and authorized.

This privileged user, usually tipadmin, is created during the installation, therefore no additional post installation steps are required. If needed, however, you can re-configure this default role-to-user association after the installation. For example, if the mapping of users and groups to the bsmAdministrator role changes, after the installation.

You re-configure the role-to-user association using wsadmin commands that modify a few configuration files in the ImpactProfile.

Table 24. Configuration files that are involved in role-to-user association for the command-line access to WebServices

File	Description
`$IMPACT_HOME/etc/server.props`	This file contains the user and password that you provided during the installation. For example, tipadmin:netcool.
`$IMPACT_PROFILE/config/cells/ImpactCell/applications/NCI.ear/deployments/NCI/jaxrpc.war/WEB-INF/web.xml`	This XML file contains security constraints settings inside the <security-constraint> and <security-role> elements. Do not edit this file manually.
`$IMPACT_PROFILE/config/cells/ImpactCell/applications/NCI.ear/deployments/NCI/META-INF/application.xml`	This file defines the role to which a user has to be mapped to access WebServices on the command line. Do not edit this file manually.
`$IMPACT_PROFILE/config/cells/ImpactCell/applications/NCI.ear/deployments/NCI/META-INF/ibm-application-bnd.xmi`	This file creates an association between the user that is defined on the Impact backend server, for example “tipadmin”, and the “bsmAdministator” role. Do not edit this file manually.

Creating groups, and users

If you use the ObjectServer based user registry, you can use the `$IMPACT_PROFILE/bin/jython/createGroupsUsers.py` script to create three typical Netcool/Impact users and groups in TIP.

About this task

The script creates 3 example users, and 3 example groups, and assigns the users to the groups in the following manner:

- impactadmin is assigned to impactAdminGroup
- impactuser is assigned to impactFullAccessGroup
- impactvieweris assigned to impactOpViewGroup

Procedure

The script is used with wsadmin console and has the following syntax:

```
wsadmin -lang jython -profileName ImpactProfile -username "TIP_admin"
-password "TIP_password" -f "createGroupsUsers.py"
 -U "o=netcoolObjectServerRepository"
-G "o=netcoolObjectServerRepository" -n "password"
```

where *TIP_admin* is the TIP administrator user name, *TIP_password* is the TIP administrator password, and *password* is the password for the new users that the script creates. For example, on Windows:

```
wsadmin.bat -lang jython -profileName
ImpactProfile -username "tipadmin" -password "tippass"
-f "C:\Program Files\IBM\tivoli\impact\bin\jython\createGroupsUsers.py"
-U "o=netcoolObjectServerRepository" -G "o=netcoolObjectServerRepository"
 -n "password"
```

Results

The new users and groups changes can be viewed in TIP in the **User and Groups** section.

Mapping groups and users to roles

If you use the Object Server based user registry, you can use the `$IMPACT_PROFILE/bin/jython/mapGroups.py` script to map users, and groups to different Netcool/Impact roles.

About this task

Users and groups must exist in the ImpactProfile before you can map them to the bsmAdministrator role. To keep the mapping for any currently mapped users and groups, you must pass these users and groups along with the new user or group, as arguments to the wsadmin command.

In addition to assigning any new users to the bsmAdministrator role all existing users including tipadmin must be specified on the mapRole command.

Procedure

The script is used with wsadmin console and has the following syntax:

```
wsadmin -lang jython -f mapRole.py -A NCI -r role
-g "<any existing groups>|group1|group2|group3"
-u "<any existing users>user1|user2|user3"
```

Where *group* is a group that you assign, *user* is a user that you assign, and *role* is the role to which you want to assign these groups or users. For example, on Windows:

```
wsadmin -lang jython -f mapRole.py -A NCI -r impactAdminUser
-g "impactAdminGroup" -u "tipadmin"
```

This command maps all the members of the impactAdminGroup group, and the tipadmin user to the bsmAdministrator role (impactAdminUser).

Appendix A. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. These are the major accessibility features you can use with *Netcool/Impact* when accessing it on the *IBM Personal Communications* terminal emulator:

- You can operate all features using the keyboard instead of the mouse.
- You can read text through interaction with assistive technology.
- You can use system settings for font, size, and color for all user interface controls.
- You can magnify what is displayed on your screen.

For more information about viewing PDFs from Adobe, go to the following web site: http://www.adobe.com/enterprise/accessibility/main.html

Appendix B. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to

IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information in softcopy form, the photographs and color illustrations might not be displayed.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered 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.

Adobe, Acrobat, PostScript and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

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

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

Other product and service names might be trademarks of IBM or other companies.

Glossary

This glossary includes terms and definitions for Netcool/Impact.

The following cross-references are used in this glossary:

- See refers you from a term to a preferred synonym, or from an acronym or abbreviation to the defined full form.
- See also refers you to a related or contrasting term.

To view glossaries for other IBM products, go to www.ibm.com/software/globalization/terminology (opens in new window).

A

assignment operator
An operator that sets or resets a value to a variable. See also operator.

B

Boolean operator
A built-in function that specifies a logical operation of AND, OR or NOT when sets of operations are evaluated. The Boolean operators are &&, || and !. See also operator.

C

command execution manager
The service that manages remote command execution through a function in the policies.

command line manager
The service that manages the command-line interface.

Common Object Request Broker Architecture (CORBA)
An architecture and a specification for distributed object-oriented computing that separates client and server programs with a formal interface definition.

comparison operator
A built-in function that is used to compare two values. The comparison operators are ==, !=, <, >, <= and >=. See also operator.

control structure
A statement block in the policy that is executed when the terms of the control condition are satisfied.

CORBA
See Common Object Request Broker Architecture.

D

database (DB)
A collection of interrelated or independent data items that are stored together to serve one or more applications. See also database server.

database event listener
A service that listens for incoming messages from an SQL database data source and then triggers policies based on the incoming message data.

database event reader
An event reader that monitors an SQL database event source for new and modified events and triggers policies based on the event information. See also event reader.

database server
A software program that uses a database manager to provide database services to other software programs or computers. See also database.

data item
A unit of information to be processed.

data model
An abstract representation of the business data and metadata used in an installation. A data model contains data sources, data types, links, and event sources.

data source
A repository of data to which a federated server can connect and then retrieve data by using wrappers. A data source can contain relational databases, XML files, Excel spreadsheets, table-structured files, or other objects. In a federated system, data sources seem to be a single collective database.

data source adapter (DSA)
A component that allows the application to access data stored in an external source.

data type
An element of a data model that represents a set of data stored in a data source, for example, a table or view in a relational database.

DB See database.

DSA See data source adapter.

dynamic link
An element of a data model that represents a dynamic relationship between data items in data types. See also link.

E

email reader
A service that polls a Post Office Protocol (POP) mail server at intervals for incoming email and then triggers policies based on the incoming email data.

email sender
A service that sends email through an Simple Mail Transfer Protocol (SMTP) mail server.

event An occurrence of significance to a task or system. Events can include completion or failure of an operation, a user action, or the change in state of a process.

event processor
The service responsible for managing events through event reader, event

listener and email reader services. The event processor manages the incoming event queue and is responsible for sending queued events to the policy engine for processing.

event reader
A service that monitors an event source for new, updated, and deleted events, and triggers policies based on the event data. See also database event reader, standard event reader.

event source
A data source that stores and manages events.

exception
A condition or event that cannot be handled by a normal process.

F

field A set of one or more adjacent characters comprising a unit of data in an event or data item.

filter A device or program that separates data, signals, or material in accordance with specified criteria. See also LDAP filter, SQL filter.

function
Any instruction or set of related instructions that performs a specific operation. See also user-defined function.

G

generic event listener
A service that listens to an external data source for incoming events and triggers policies based on the event data.

graphical user interface (GUI)
A computer interface that presents a visual metaphor of a real-world scene, often of a desktop, by combining high-resolution graphics, pointing devices, menu bars and other menus, overlapping windows, icons and the object-action relationship. See also graphical user interface server.

graphical user interface server (GUI server)
A component that serves the web-based graphical user interface to web browsers through HTTP. See also graphical user interface.

GUI See graphical user interface.

GUI server
See graphical user interface server.

H

hibernating policy activator
A service that is responsible for waking hibernating policies.

I

instant messaging reader
A service that listens to external instant messaging servers for messages and triggers policies based on the incoming message data.

instant messaging service
A service that sends instant messages to instant messaging clients through a Jabber server.

IPL See Netcool/Impact policy language.

J

Java Database Connectivity (JDBC)
An industry standard for database-independent connectivity between the Java platform and a wide range of databases. The JDBC interface provides a call level interface for SQL-based and XQuery-based database access.

Java Message Service (JMS)
An application programming interface that provides Java language functions for handling messages.

JDBC See Java Database Connectivity.

JMS See Java Message Service.

JMS data source adapter (JMS DSA)
A data source adapter that sends and receives Java Message Service (JMS) messages.

JMS DSA
See JMS data source adapter.

K

key expression
An expression that specifies the value that one or more key fields in a data item must have in order to be retrieved in the IPL.

key field
A field that uniquely identifies a data item in a data type.

L

LDAP See Lightweight Directory Access Protocol.

LDAP data source adapter (LDAP DSA)
A data source adapter that reads directory data managed by an LDAP server. See also Lightweight Directory Access Protocol.

LDAP DSA
See LDAP data source adapter.

LDAP filter
An expression that is used to select data elements located at a point in an LDAP directory tree. See also filter.

Lightweight Directory Access Protocol (LDAP)
An open protocol that uses TCP/IP to provide access to directories that support an X.500 model and that does not incur the resource requirements of the more complex X.500 Directory Access Protocol (DAP). For example, LDAP can be used to locate people, organizations, and other resources in an Internet or intranet directory. See also LDAP data source adapter.

link An element of a data model that defines a relationship between data types and data items. See also dynamic link, static link.

M

mathematic operator
A built-in function that performs a mathematic operation on two values. The mathematic operators are +, -, *, / and %. See also operator.

mediator DSA
A type of data source adaptor that allows data provided by third-party systems, devices, and applications to be accessed.

N

Netcool/Impact policy language (IPL)
A programming language used to write policies.

O

operator
A built-in function that assigns a value to a variable, performs an operation on a value, or specifies how two values are to be compared in a policy. See also assignment operator, Boolean operator, comparison operator, mathematic operator, string operator.

P

policy A set of rules and actions that are required to be performed when certain events or status conditions occur in an environment.

policy activator
A service that runs a specified policy at intervals that the user defines.

policy engine
A feature that automates the tasks that the user specifies in the policy scripting language.

policy logger
The service that writes messages to the policy log.

POP See Post Office Protocol.

Post Office Protocol (POP)
A protocol that is used for exchanging network mail and accessing mailboxes.

precision event listener
A service that listens to the application for incoming messages and triggers policies based on the message data.

S

security manager
A component that is responsible for authenticating user logins.

self-monitoring service
A service that monitors memory and other status conditions and reports them as events.

server A component that is responsible for maintaining the data model, managing services, and running policies.

service
A runnable sub-component that the user controls from within the graphical user interface (GUI).

Simple Mail Transfer Protocol (SMTP)
An Internet application protocol for transferring mail among users of the Internet.

Simple Network Management Protocol (SNMP)
A set of protocols for monitoring systems and devices in complex networks. Information about managed devices is defined and stored in a Management Information Base (MIB). See also SNMP data source adapter.

SMTP See Simple Mail Transfer Protocol.

SNMP
See Simple Network Management Protocol.

SNMP data source adapter (SNMP DSA)
A data source adapter that allows management information stored by SNMP agents to be set and retrieved. It also allows SNMP traps and notifications to be sent to SNMP managers. See also Simple Network Management Protocol.

SNMP DSA
See SNMP data source adapter.

socket DSA
A data source adaptor that allows information to be exchanged with external applications using a socket server as the brokering agent.

SQL database DSA
A data source adaptor that retrieves information from relational databases and other data sources that provide a public interface through Java Database Connectivity (JDBC). SQL database DSAs also add, modify and delete information stored in these data sources.

SQL filter
An expression that is used to select rows in a database table. The syntax for the filter is similar to the contents of an SQL WHERE clause. See also filter.

standard event reader
A service that monitors a database for new, updated, and deleted events and triggers policies based on the event data. See also event reader.

static link
An element of a data model that defines a static relationship between data items in internal data types. See also link.

string concatenation
In REXX, an operation that joins two characters or strings in the order specified, forming one string whose length is equal to the sum of the lengths of the two characters or strings.

string operator
A built-in function that performs an operation on two strings. See also operator.

U

user-defined function
A custom function that can be used to organize code in a policy. See also function.

V

variable
A representation of a changeable value.

W

web services DSA
A data source adapter that exchanges information with external applications that provide a web services application programming interface (API).

X

XML data source adapter
A data source adapter that reads XML data from strings and files, and reads XML data from web servers over HTTP.

Index

Special characters

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

Printed in USA

SC14-7551-00