

Business Service Manager
Version 6.2.0

Installation Guide



Note

Before using this information and the product it supports, read the information in [Appendix A, “Notices,” on page 247.](#)

Edition notice

This edition applies to IBM® Tivoli Business Service Manager Version 6 Release 2.0 and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 2008, 2020.**

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

Contents

Chapter 1. About this publication.....	1
Audience.....	1
Publications.....	1
TBSM library.....	1
Prerequisite publications.....	1
Related publications.....	2
Accessing terminology online.....	2
Accessing publications online.....	2
Ordering publications	3
Accessibility.....	3
Tivoli technical training.....	3
Support information.....	3
Conventions used in this publication	3
Typeface conventions	3
Chapter 2. Introduction to IBM Tivoli Business Service Manager.....	5
Chapter 3. What's new in TBSM Version 6.2.0	7
Chapter 4. Technical overview of TBSM.....	11
TBSM architecture.....	11
TBSM components.....	11
Integrated applications.....	13
Operating system variables and paths.....	15
Java support.....	16
Chapter 5. Planning.....	17
Prerequisite scanner.....	17
System requirements.....	17
Installing prerequisite software.....	18
TBSM installation user.....	18
Install checklist.....	19
Failover considerations.....	19
IBM Tivoli Netcool Impact Considerations.....	19
IBM Tivoli Netcool OMNIBus Considerations.....	20
Netcool/OMNIBus triggers.....	21
Installation with external LDAP user registry considerations.....	22
User registry considerations.....	23
Dashboard server LDAP configuration.....	24
Data server LDAP configuration	24
Netcool/OMNIBus user registry.....	24
Use Netcool/OMNIBus to access LDAP server.....	25
Multiple user registry warnings.....	25
TBSM users, user groups, and roles.....	25
DB2 Installation.....	27
DB2 Database Setup.....	28
Database Disk Space Requirements.....	28
DB2 Administration and Maintenance.....	29
Database Schema Descriptions.....	29
Other Important Considerations.....	30

Chapter 6. Installing TBSM.....	31
Installation types.....	31
Installation order.....	31
DB2 schema configuration.....	31
Advanced DB2 configuration.....	34
Creating database schema after installation.....	38
Installing TBSM interactively.....	39
Performing installations.....	39
TBSM Discovery Library Toolkit configuration.....	40
Tivoli Event Integration Facility Probe Installation.....	43
Installing the Data Server component.....	44
Installing the Dashboard server component.....	55
Installing TBSM in silent mode.....	61
Dashboard Server response file.....	62
Data Server response file.....	63
DBConfig response file.....	65
Installing TBSM using the console.....	67
Uninstalling TBSM.....	68
Uninstalling TBSM Data Server.....	68
Uninstalling TBSM Dashboard Server.....	70
Uninstalling DbConfig.....	72
Reinstalling TBSM.....	73
Reinstalling TBSM after a failed installation.....	73
Chapter 7. Restoring system from a backup.....	75
Chapter 8. Installing and configuring the TBSM Agent.....	77
Installing the TBSM agent.....	77
Configuring the TBSM agent.....	78
Enabling historical reporting.....	79
Enabling Agent Management Services for TBSM.....	80
Setting up TBSM agents as Windows services.....	80
TBSM Agent installation reference.....	81
Workspaces reference.....	81
Attribute groups and attributes for Business Service Management Agent.....	83
Disk capacity planning for historical data.....	94
Predefined situations.....	95
Predefined take action commands.....	97
Predefined policies.....	97
Chapter 9. National Language Support.....	99
Installing the language pack for TBSM Agent.....	99
Uninstalling the language pack for TBSM Common Agent.....	100
Chapter 10. Migrating to TBSM 6.2.0.....	101
Chapter 11. Configuring TBSM: post installation.....	105
Installing the Historical Reports.....	105
Specifying the schema name.....	107
Configuring data source failover.....	108
Configuring IBM Tivoli Business Service Manager for failover.....	108
Overview of the failover process.....	108
DB2 high availability configuration.....	110
Configuring the ObjectServer communication information for failover.....	113
Configuring Data servers for failover.....	114

Replicating the alerts.service_deps table required for OMNIBus failover.....	116
Starting the servers in a failover environment.....	117
Verifying that the servers failover successfully.....	118
Load balancing.....	119
Exporting data from a stand-alone server.....	121
Setting up a load balancing cluster.....	122
Joining a node to a load balancing cluster.....	123
Enabling server-to-server trust.....	124
Verifying a load balancing implementation.....	126
Preparing the HTTP server for load balancing.....	127
Setting clone IDs for nodes.....	128
Generating the plugin-cfg.xml file.....	129
Configuring SSL from each node to the IBM HTTP Server.....	132
Importing data to a cluster.....	133
Removing a node.....	134
Removing a load balancing cluster.....	135
Monitoring a load balancing cluster.....	135
Configuring secure connections.....	136
Secure connections overview.....	136
Certificate management for TBSM servers.....	138
Running the secure server command.....	140
Configure secure communications to Netcool/OMNIBus.....	143
Securing connections to the Discovery Library Toolkit.....	146
Verifying secure connections.....	146
Enabling TLSv1.2 for the Dashboard server.....	147
Replacing the default SSL certificates with certificates signed by a certificate authority.....	148
Enabling the Policy Editor from the TBSM Rules page.....	153
Modifying the tivoli_eif.props file.....	153
Ensure service templates were created.....	153
Chapter 12. Troubleshooting installation issues.....	155
Common installation issues.....	155
Database configuration utility issues.....	156
Database configuration installer fails.....	156
TBSM database install fails when user id contains a hyphen.....	156
Russian Windows TBSM and DB2 install fails.....	157
Cannot create database.....	157
Database configuration error messages.....	157
Windows issues.....	158
RAD shell issue connecting to the Data server.....	158
Dashboard server cannot connect to the Data server on Linux.....	159
Clearing the Java plug-in cache on Windows.....	159
Clearing the Java plug-in cache on UNIX.....	159
Default TBSM groups not created during the installation process.....	159
Netcool/OMNIBus install fails on SuSE with ssh access.....	160
TBSM server fails after Netcool/Impact install/migration.....	161
Separating the Data server and Dashboard server with a firewall.....	161
Chapter 13. Reference.....	163
Log files that TBSM uses.....	163
Log files generated by the installation of Discovery Library Toolkit.....	163
Sample response file.....	163
Dashboard Application Service Hub overview.....	164
Accepting the security certificate.....	164
Logging in.....	164
Port assignments.....	165
Viewing the application server profile.....	166

Configuring.....	166
Administering.....	207
Appendix A. Notices.....	247
Trademarks.....	248
Index.....	249

Chapter 1. About this publication

This guide contains information how to operate, maintain, and configure the product.

Audience

This publication is for administrators and system programmers who need to use, install, maintain, or configure TBSM.

Publications

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

TBSM library

The following documents are available in the TBSM library:

- *Installation Guide*, GI11-8054-10
Provides information about installing the product.
- *Quick Start*, GI11-8055-04
Provides overview information about TBSM.
- *Exploring IBM Tivoli Business Service Manager*, GI11-8056-10
Provides an overview of the product features.
- *Administrator's Guide*, SC23-6040-10
Provides information about managing and configuring TBSM.
- *Service Configuration Guide*, SC23-6041-10
Provides information on how to use the features of the product console.
- *Customization Guide*, SC23-6042-10
Provides information on how to customize select features of the product.
- *Troubleshooting Guide*, GI11-8057-10
Provides information about resolving common problems with the product.
- *Release Notes*,
Provides latest information about the product discovered late in the test cycle that cannot be incorporated into the other publications.

Prerequisite publications

To use the information in this publication effectively, you must have some prerequisite knowledge, which you can obtain from the publications listed here.

These publications are available on the Tivoli Netcool/OMNIbus Knowledge Center:

https://www.ibm.com/support/knowledgecenter/SSSHTQ_8.1.0/com.ibm.netcool_OMNIbus.doc_8.1.0/omnibus/wip/common/reference/omn_ref_PDFbooks.html

- IBM Tivoli Netcool/OMNIbus *User Guide*

Provides an overview of Netcool/OMNIBus components, as well as a description of the operator tasks related to event management using the desktop tools. TBSM uses Netcool/OMNIBus as its event manager.

- IBM Tivoli Netcool/OMNIBUS *Administration Guide*

Provides information about how to perform administrative tasks using the Netcool/OMNIBus Administrator GUI, command line tools, and process control. It also contains descriptions and examples of ObjectServer SQL syntax and automations.

- IBM Tivoli Netcool/OMNIBUS *Probe and Gateway Guide*

Provides information contains introductory and reference information about probes and gateways, including probe rules file syntax and gateway commands. For more information about specific probes and gateways, refer to the documentation available for each probe and gateway.

- IBM Tivoli Netcool/OMNIBUS *Probe for Tivoli EIF*

Provides reference information about the optional Probe for Tivoli EIF that is included with TBSM.

Related publications

The following documents also provide useful information and are included in the TBSM Information Center.

These publications are available on the IBM Tivoli Business Service Manager Knowledge Center:

<https://www.ibm.com/support/knowledgecenter/SSSPFK>

- IBM Tivoli Netcool/Impact *Administration Guide*

Provides information about installing, configuring and running Netcool/Impact and its related software components. TBSM uses Netcool/Impact policies to parse events and other data.

- IBM Tivoli Netcool/Impact *User Interface Guide*

Provides information about using the Netcool/Impact user interface.

- IBM Tivoli Netcool/Impact *Policy Reference Guide*

Provides reference information about the Netcool/Impact Policy Language (IPL). It contains complete information about policy language syntax, data types, operators and functions.

- IBM Tivoli Netcool/Impact *Solutions Guide*

Provides information about implementing Netcool/Impact in your environment.

- IBM Tivoli Netcool/Impact *DSA Reference Guide*

Provides reference information about Netcool/Impact data source adaptors (DSA).

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

The format of the publications is PDF, HTML, or both.

IBM posts publications for this and all other Tivoli products, as they become available and whenever they are updated, to the Tivoli Knowledge Center at <https://www.ibm.com/support/knowledgecenter/SSSPFK>

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

Ordering publications

According to e-Business strategy, IBM Publications Center no longer supports ordering publications. The publications are made available in electronic format to be viewed or downloaded free of charge.

For documentation related to TBSM, go to <https://www.ibm.com/support/knowledgecenter/en/SSSPFK>.

Accessibility

This guide contains information how to operate, maintain, and configure the product.

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully. In this release, the TBSM console does not meet all accessibility requirements.

Tivoli technical training

For Tivoli technical training information, refer to the IBM developerWorks Website at <https://www.ibm.com/developerworks>.

Support information

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

Online

Access the IBM Software Support site at <https://www.ibm.com/support/home/>.

IBM Support Assistant

The IBM Support Assistant is a free local software serviceability workbench that helps you resolve questions and problems with IBM software products. The Support Assistant provides quick access to support-related information and serviceability tools for problem determination. To install the Support Assistant software, go to <https://www-01.ibm.com/software/support/isa/>.

Troubleshooting Guide

For more information about resolving problems, see the problem determination information for this product.

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

Chapter 2. Introduction to IBM Tivoli Business Service Manager

This information can help you understand IBM Tivoli Business Service Manager (TBSM), including its business value and key technologies.

TBSM delivers the real-time information that you need in order to respond to alerts effectively and in line with business requirements, and optionally to meet service-level agreements (SLAs).

The TBSM tools enable you to build a service model that you integrate with IBM Tivoli Netcool/OMNIBus alerts or optionally with data from an SQL data source. TBSM includes optional components that let you access data from other IBM Tivoli applications such as IBM Tivoli Monitoring, and IBM Tivoli Application Dependency Discovery Manager. TBSM processes the external data based on the service model data you created in the TBSM database and returns a new or an updated TBSM service event to Netcool/OMNIBus.

The TBSM console provides a graphical user interface (GUI) that allows you to logically link services and business requirements within the service model. The service model provides an operator with a view of how, second by second, an enterprise is performing at any given moment in time or how the enterprise has performed over a given time period.

Chapter 3. What's new in TBSM Version 6.2.0

This documentation applies to TBSM version 6.2.0 and all fix packs, unless indicated otherwise. TBSM Version 6.2.0 contains support for Jazz for Service Management as the new Dashboard for TBSM and Netcool/Impact 7.1.0.

Jazz for Service Management

TBSM has been updated to use the IBM Dashboard Application Services Hub as its new UI. In order to use these new features, you also need to have Jazz® for Service Management and the IBM Dashboard Application Services Hub installed as part of your environment.

For more information see *Administration Guide > TBSM and IBM Dashboard Application Services Hub*.

IBM Dashboard Application Services Hub

The IBM Dashboard Application Services Hub provides user interface and dashboard services in Jazz for Service management. This new self-service dashboard capability enables you to combine a variety of visual widgets such as gauges, tables, charts, lists or topology views into custom dashboards using a guided work flow. These dashboards can also include management data from sources such as:

- Service status and metrics from TBSM
- Third-party data from Netcool/Impact
- Performance metrics from IBM Tivoli Monitoring

Mobile Support: The self-service dashboard enable you to view business dashboards on mobile devices including tablets and phones. This enables access to both information technology and business data anytime / anywhere and gives you the ability to support your customers more effectively.

New functionality in TBSM 6.2

The following functionality is new in TBSM 6.2:

- TBSM widgets and DASH sidget can communicate with each other without needing to create a separate connection.
- Since TBSM Portlet is installed on JazzSM, all the CURI TBSM datasets are available locally without the need for a remote connection.
- The dashboard provides the following types of widgets that can be used by custom pages to build responsive and interactive pages:
 - Status gauge
 - Area chart
 - Bar chart
 - Line chart
 - Topology

Features modified in, or removed, from TBSM

The following features have been either modified or removed in TBSM 6.2:

- Tivoli Integrated Portal (TIP) and its components (Graphs, Custom Canvas, Service viewer Applet code, Birt charts) have been removed.
- TBSM portlets and pages throughout DASH have been ported to DASH.
- TBSM has been upgraded to support the latest version of Java (JDK 1.8).
- Packaging method changed from Install Anywhere to Installation Manager.

- Obsolete features and functionality (Custom Canvas, JViews, and Birt charts) have been removed.
- Prerequisite stack (for example Netcool and Netcool Impact) moved from embedded to stand alone.
- Impact runs in the Liberty server and DASH server runs in WAS 8.5 Service Navigation.
- Custom canvases have been removed
- Birt charts have been removed but COGNOS reports are available as is.

Service Editor

The following changes have been made to the Service Editor:

- Tabs have been removed and the Edit view is now the default in the Server Editor portal. The View tab has been removed as it was based on Applet and now you can use DASH Topology widgets to create a similar topological view of services.
- Clicking on the service now opens the corresponding Service view in the Service Editor
- The Invalidate button and the button related to ESDA have been hidden by default and can be enabled if you wish by enabling the flag defined in property file. To enable it on the screen, go to JazzSM directory where TBSM Deployable Artifact has been installed: `./opt/IBM/JazzSM/profile/installedApps/JazzSMNode01Cell/isc.ear/sla.war/etc/RAD_sla.props` and change the key `'impact.esda.enable=false'` value to true and restart the JazzSM Server.
- The Policy Editor communication protocol has been changed from SOAP to Rest API.
- SSO between JazzSM and Impact is now a prerequisite for the TBSM Policy editor
- Run functionality in the Policy Editor has been disabled.

Service Availability

The following changes have been made to Service Availability:

- Events/Node graphs have been implemented using Rave based on the policy.
- Service Viewer has been removed because it was Applet based.

Change to TBSM Admin pages, to be within JazzSM

The following changes have been made to the TBSM admin pages to be within JazzSM:

- TBSM admin pages have been put under the catalog listing of Business Service Management.
- You can create custom pages using TBSM widgets by choosing from the Business Service Management catalog.

Changes to the use of TBSM widgets

The following changes have been made to the use of TBSM widgets:

- The pre-existing TBSM widget remains functional as it was in the earlier version.
- TBSM widgets can communicate with DASH widgets. Clicking on any row on the Service Tree can render respective topology.

What's new in Netcool/Impact

TBSM Data Server integrates with Netcool/Impact version 7 release 1.0.13. The new version includes these enhancements:

New visualization: The new visualization include Operator View customization enhancements and UI Services provided by Jazz for Service Management. These will enable clients to link their own data accessed through Impact's proven data access methods with visual widgets such as gauges, tables, or lists to create dashboards.

Linked data integration: Netcool/Impact can also use the Jazz for Service Management registry services that follow the Open Services for Lifecycle Collaboration (OSLC) standards.

Service Level Objective (SLO) Reporting: Enables you to establish and report on service level objectives based on their own measures (for example, incidents, tickets, and availability).

Consumability: Continued improvements to enhance the user experience, including MWM cluster replication and e-mail reader enhancements.

Enhanced Web Services Integrations and Wizards: Enhances and simplifies access to web services data sources.

Chapter 4. Technical overview of TBSM

This section contains topics about the product architecture and the main software components.

TBSM architecture

This section describes the basic architecture of the IBM Tivoli Business Service Manager (TBSM).

TBSM architecture shows the basic architecture for TBSM. The TBSM Data server analyzes IBM Netcool/OMNIbus ObjectServer events or SQL data for matches against the incoming-status rules you configured for your service models. If the matching data changes the service status, the status of the TBSM service model changes accordingly. When a services status changes, TBSM sends corresponding service events back to the ObjectServer.

You can also use data from an external database or an ObjectServer to drive custom views and charts. The Discovery Library Toolkit lets you create TBSM service objects using data from Discovery Library Adaptor (DLA) books or from the IBM Tivoli Application Dependency Discovery Manager.

The TBSM users and group permissions are managed by the Dashboard Application Service Hub, which can authenticate users internally, or use data from an external source such as an ObjectServer or LDAP server.

TBSM components

This topic describes the components included on the product DVD.

The applications IBM Tivoli Netcool/Omnibus, Netcool/Omnibus WebGUI, IBM Tivoli Netcool/Impact and JazzSM/Dashboard Application Service Hub are not included on the TBSM product DVD. These applications must be installed as pre-requisite products on a host that is accessible by the TBSM server.

TBSM has the following components included in the Installer package or DVD:

- TBSM Dashboard server
- TBSM Data server
- TBSM DBConfig
- Discovery Library Toolkit

Note: In TBSM 6.2, Netcool/OMNIbus, Netcool/OMNIbus WebGUI and Netcool/Impact products are not included in the installer. These products are prerequisites and must be installed separately before installing TBSM.

Tivoli Netcool/OMNIbus

TBSM monitors the Tivoli Netcool/OMNIbus ObjectServer for incoming events. The ObjectServer collects events from probes, monitors, and other applications such as IBM Tivoli Monitoring. You use TBSM to create service models that respond to the data received in the incoming events. For example, the incoming event data can change the status of a service or start the tracking of a potential SLA violation. In short, if you can set up a probe or other application to forward data to the TBSM ObjectServer, you can use that data to build and monitor your service models. The TBSM installation package includes Netcool/OMNIbus. If you want to use the Discovery Library toolkit, or the IBM Tivoli Event Integration Facility (EIF) probe you need version 7.1 or higher.

Note that Tivoli Netcool/OMNIbus V8.1.0.x should be installed as a pre-requisite before installing TBSM.

For more information see: [IBM Tivoli Netcool/OMNIbus documentation](#)

Netcool/OMNIBus Web GUI

The Web GUI is the browser console for Netcool/OMNIBus and TBSM uses Web GUI components to display events related to service models. The Active Event List (AEL) and Service Details portlet in TBSM are Web GUI components, and are required to be installed as part of the pre-requisite products before installing TBSM on the server where JazzSM is installed.

For more information see: **IBM Tivoli Netcool/OMNIBus Considerations** in the *IBM Tivoli Business Service Manager Installation Guide* .

TBSM Dashboard server

The TBSM Dashboard server manages the TBSM console display. You can have multiple dashboard servers for a single data server. The dashboard server enhances the scalability, performance, and availability of TBSM.

The TBSM Dashboard server communicates with the TBSM Data server to support the creation and visualization of service models through connected TBSM consoles. As console users view portions of the service model, the dashboard server will acquire and maintain status of services from the data server.

TBSM Data server

The TBSM Data server monitors the ObjectServer and external databases for data that affect the status of the services you configured in the TBSM console or with the RAD shell command line tool. The server calculates the status of these services by applying rules to the external data. Your service models and the rules are stored in the TBSM database.

TBSM DB2 database

The TBSM DB2® database stores all the information on the service models you created in the TBSM console. This data includes rules that determine how your service model changes in relation to data in external data sources. This database also includes tables for the metrics and markers used in the Time Window Analyzer and demo data. A Metric History database, which has a default name of TBSMHIST, is also included to store the historical metric data,

Discovery Library Toolkit

The Discovery Library Toolkit enables TBSM to discovery resources and to automatically build service models from Discovery Library data sources. These sources include: IBM Tivoli Application Dependency Discovery Manager , Discovery Library books conforming to the common data model, Discovery Library books containing objects for an alternate namespace, the Discovery Library toolkit API, or auto-pop objects.

Data discovered through the toolkit can be enriched through notifications sent to Impact. This enriched data can then be used in the automatic building of the service model.

Netcool/Impact

Netcool/Impact is the automation, correlation, and integration engine for the IBM Tivoli Netcool suite of software products. You can use Netcool/Impact to automate event management tasks, to correlate event information with other information in your environment, and to integrate Netcool products with a wide variety of third party systems and applications.

TBSM 6.2 does not include the Netcool/Impact as part of the Data Server. You need to install Impact 7.1.0.13 prior to installing TBSM.

As a consequence of this integration, you can now take advantage of these Netcool/Impact capabilities:

- You can use Netcool/Impact services and policies to acquire, enrich, and pass data to TBSM to use for service status determination or visualization.

- TBSM uses the same policy functions and policy language as Netcool/Impact. Javascript is supported as a policy language in addition to IPL (Impact Policy Language).
- Event enrichment is supported as an out-of-box function. Impact policies enrich events before TBSM reads these same events for status determination and propagation.
- The Impact User Interface is installed separately as prerequisite along with Impact Server and SSO configuration between Dashboard Application Service Hub and Impact. This provides a common user interface for administration of both TBSM and Impact policies and services.
- The Data server package includes a name server that enables you to access Netcool/Impact server clusters.

For more information about Netcool/Impact, see the Tivoli Netcool/Impact publications at: [Netcool/Impact Documentation](#).

Integrated applications

This section is an overview of the optional external applications you can integrate with TBSM.

The following applications either forward data to TBSM, or receive data from TBSM:

- Using the IBM Tivoli EIF probe, you can forward data from IBM Tivoli Monitoring version 6 release 1 and above, Tivoli Enterprise Console® version 3 release 9 or later, IBM Tivoli Netview version 3 release 7 or later.
- IBM Tivoli Application Dependency Discovery Manager version 7 release 1.2 or later
- IBM Tivoli Change and Configuration Management Database (CCMDB) version 7 releases 1 and 1.1
- Discovery Library Adapters including those from the following products:
 - IBM Tivoli Monitoring (6.2.3 or higher is recommended)
 - IBM Tivoli Composite Application Manager for SOA
 - IBM Tivoli Composite Application Manager for WebSphere®
 - IBM Tivoli Composite Application Manager for Transaction Tracking
 - IBM Tivoli Network Manager
 - IBM Tivoli NetView® for z/OS®
 - IBM Tivoli Storage Productivity Center version 4, release 1.1

You can launch to or from the following applications from TBSM:

- Tivoli Monitoring 6.2 with fix pack 1 or later
- Tivoli Application Dependency Discovery Manager 7.1 or later
- CCMDB version 7.1 or later
- Netcool/OMNIBus Web GUI component.
- IBM Tivoli Network Manager IP Edition version 3 release 8
- IBM Tivoli Composite Application Manager for Transactions version 7 release 1.0.2
- IBM Tivoli TotalStorage Productivity Center (TPC)

Note: For launch support, the supported product versions may be more restrictive than those specified for data exchange above.

Jazz for Service Management

TBSM uses Dashboard Application Service Hub as its UI component in 6.2 instead of Tivoli Integrated Portal. In order to use these new features, you also need to have Jazz for Service Management and the IBM Dashboard Application Services Hub installed as part of your environment.

Jazz for Service Management employs a new deployment pattern and mechanism that helps you integrate shared components such as your User Interface, Linked Data Registry, Reporting, Security, and Administrative Services. This new mechanism helps you speed up delivery cycles for clients and simplify deployments.

For more information see *Administration Guide > TBSM and IBM Dashboard Application Services Hub*.

IBM Dashboard Application Services Hub

The IBM Dashboard Application Services Hub provides user interface and dashboard services in Jazz for Service management. This new self-service dashboard capability enables you to combine a variety of visual widgets such as gauges, tables, charts, lists or topology views into custom dashboards using a guided work flow. These dashboards can also include management data from sources such as:

- Service status and metrics from TBSM
- Third-party data from Netcool/Impact
- Performance metrics from IBM Tivoli Monitoring

Mobile Support: The self-service dashboard enable you to view business dashboards on mobile devices including tablets and phones. This enables access to both information technology and business data anytime / anywhere and gives you the ability to support your customers more effectively.

Tivoli Event Integration Facility (EIF) probe

You can set up the optional IBM Tivoli Event Integration Facility (EIF) probe to access the event data from applications such as IBM Tivoli Monitoring, Tivoli Enterprise Console, and Tivoli Netview. The probe forwards the event data to the TBSM Netcool/OMNIbus ObjectServer. You can use TBSM to create service models based on the event data from the Event Pump for z/OS, Tivoli Monitoring (and Tivoli Monitoring agents), Tivoli Enterprise Console, and Tivoli NetView.

IBM Tivoli Netcool/Impact

Netcool/Impact is the automation, correlation, and integration engine for the IBM Tivoli Netcool® suite of software products. You can use Netcool/Impact to automate event management tasks, to correlate event information with other information in your environment, and to integrate Netcool products with a wide variety of third party systems and applications.

You can configure Netcool/Impact to forward events to the Netcool/OMNIbus ObjectServer monitored by TBSM and use those events to update your service model. Netcool/Impact is designed for Netcool administrators who want to enhance, customize, and extend the capabilities of the Netcool suite. For more information, see the Netcool/Impact publications.

Change and Configuration Management Database

TBSM can launch into a Change and Configuration Management Database (CCMDB) associated with a Tivoli Application Dependency Discovery Manager. If you create service models with the TBSM Discovery Library integration, these services can contain data about a Tivoli Application Dependency Discovery Manager server. If a service contains data about a Tivoli Application Dependency Discovery Manager server, you can launch the Tivoli Application Dependency Discovery Manager and CCMDB consoles from the TBSM console. Likewise, you can also launch TBSM from the Tivoli Application Dependency Discovery Manager console.

Operating system variables and paths

On both the Data server and the Dashboard server a script is provided that allows you to set environment variables for quick access to the TBSM directory structure. If you do not set the variables, you can substitute directories with full path names when you run commands.

You must run the script that applies to the servers that you installed. If you installed both servers on the same system, you must run both scripts.

The locations of these setup scripts on UNIX systems are as follows:

- `installdirectory/tbsm/bin/setupTBSMData.sh` for the Data server
- `installdirectory/tbsmdash/bin/setupTBSMDash.sh` for the Dashboard server

where *installdirectory* is the directory in which you installed the server. The default directory is `/opt/IBM/tivoli`.

The syntax used to run the UNIX scripts is:

```
. installdirectory/tbsm/bin/setupTBSMData.sh
```

The locations of these setup scripts on Windows systems are as follows:

- `installdirectory\tbsm\bin\setupTBSMData.bat` for the Data server
- `installdirectory\tbsmdash\bin\setupTBSMDash.bat` for the Dashboard server

where *installdirectory* is the directory in which you installed the server. The default directory is `C:\Program Files\IBM\tivoli`.

The `setupTBSMDash` script sets the following variables:

```
TBSM_HOME=/opt/IBM/tivoli/tbsmdash
JAZZ_HOME=/opt/IBM/JazzSM
TBSM_DASHBOARD_SERVER_HOME=
/opt/IBM/JazzSM/profile/installedApps/JazzSMNode01Cell/isc.ear/sla.war
DASHBOARD_PROFILE=JazzSMProfile
JAVA_HOME=/opt/IBM/tivoli/tbsmdash/_jvm/jre
```

The `setupTBSMData` script sets the following variables:

```
TBSM_DATA_SERVER_HOME=/opt/IBM/tivoli/impact/wlp/usr/servers/TBSM/apps/TBSM.ear/
TBSM_HOME=/opt/IBM/tivoli/tbsm
TBSM_LIBS=/opt/IBM/tivoli/impact/lib3p
HOSTNAME=<hostname of the installed server>
HTTPSPORT=<https port of the Impact GUI server>
HTTPPORT=<http port of the Impact GUI server>
```

Variables used in TBSM Publications

For many of the commands and paths specified in this publication, both the UNIX and Windows equivalents are provided. However, in instances where only the UNIX convention has been specified, follow these directions for Windows systems.

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.

Java support

This topic describes the Java™ runtime Environment (JRE) plug-in versions that are required for the IBM Tivoli Business Service Manager user interface in a web browser.

Supported Java runtime versions: The most up-to-date information about supported hardware, software, browsers and operating systems is provided by the IBM Software Product Compatibility Reports at:

<https://www.ibm.com/software/reports/compatibility/clarity/prereqsForProduct.html>

1. In the **Full or partial product name:** field, type **Business Service** and click the search button.
2. From the **Search Results**, select **Tivoli Business Service Manager**.
3. From the **Version** field, select **6.2.0**.
4. From **Mandatory capabilities:**, select **Java**.
5. Click **Submit**.

Note: The Java Runtime Environment that is being used should be updated to the most recent fix level.

Important: These web browser settings are required:

- JavaScript is enabled in the browser.
- Set your browser to allow pop-up windows. If you block pop-up windows, you will disable features of TBSM that require pop-up windows.
- Set your browser to accept third-party cookies.

Chapter 5. Planning

This section details preinstallation considerations, such as supported hardware, software, and operating systems.

Prerequisite scanner

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.

TBSM 6.2 does not have a prerequisite scanner. You can run the prerequisite scanner for individual prerequisite products, for example, Impact, Omnibus, WebGUI and JazzSM before installing the respective products.

System requirements

Your environment must meet these software requirements.

Software Product Compatibility Reports

The most up-to-date information about supported hardware, software, browsers and operating systems is provided by the IBM [Software Product Compatibility Reports](#) at:

<http://pic.dhe.ibm.com/infocenter/prodguid/v1r0/clarity/index.html>

For more information about running the Software Product Compatibility Reports, see the Overview and Planning section of the [TBSM Wiki](#).

Preparing operating systems for installation

TBSM is built on the WebSphere Application Server and you need to review information on how to prepare system before you install TBSM.

For information on preparing your operating system, see:

http://pic.dhe.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=%2Fcom.ibm.websphere.installation.base.doc%2Finfo%2Faes%2Fae%2Ftins_prepare.html

Tuning your operating system

TBSM is built on the WebSphere Application Server and you need to review information on how to tune your system before you install TBSM.

For information on tuning your operating system, see:

https://www.ibm.com/support/knowledgecenter/SSAW57_9.0.0/as_ditamaps/was900_welcome_ndmp.html

Installer temp file space requirement

The TBSM requires a minimum of 500 MB of temporary space to run on all operating systems. Otherwise, the installer will fail.

Note: You must install one of the 2016 versions of Windows Server because other versions are not supported.

Installing prerequisite software

About this task

TBSM requires IBM Netcool/Impact, IBM Netcool/OMNIBus , OMNIBus WebGUI, JazzSM and IBM DB2 products as prerequisites. You install the TBSM databases on the DB2 instance and specify the host, port, and user information during the TBSM installation.

Software Product Compatibility Reports:

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

<https://www.ibm.com/software/reports/compatibility/clarify/index.html>

For more information about running the Software Product Compatibility Reports, see the Overview and Planning section of the [TBSM Wiki](#).

<https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/Tivoli%20Business%20Service%20Manager1>

The order in which the software is installed is important. Install the software in the following order:

1. DB2 Database Configuration utility
2. TBSM Data Server / Discovery Library Toolkit
3. TBSM Dashboard Server

Note: If you are installing TBSM in a failover environment, see the failover information in the Configuring: post installation section of this guide.

TBSM installation user

About this task

The *installation user* is the person who installs products in the TBSM program suite. The installation user must install every TBSM product as the same system user.

Note: The user installing TBSM should be the user who installed the prerequisite software.

On Windows platforms, the TBSM installation user can be any user on the system that is a member of the Administrators group.

Restriction: Windows You cannot install TBSM, DB2, or configure the DB2 databases for TBSM when you are logged in with a user id containing non-English characters. DB2 does not recognize a user id with non-English characters, such as Russian. For example, the Windows Administrator userid on a Russian operating system has Russian characters in the name. Typically, you do not have these problems in UNIX because the user names have English characters.

To install DB2, rename the user id and Administrators group to have English characters only. Log in with the renamed Administrator user id. Install DB2 while logged in with the renamed Administrator id. You also need to run the TBSM Database Configuration utility with the same renamed Administrator user id.

You must use the port used for DB2.

However, to install TBSM, you must log in with a user id with English characters that is a member of the Administrators group. The user id cannot be the renamed non-English Administrator id. You must log in with a user id that was originally created with English characters. One example is to use the `db2admin` user id that is normally created during a DB2 installation.

TBSM database user name restriction: The TBSM database does not recognize a user id containing the following dash/hyphen character: "-".

Install checklist

This section provides a preinstall checklist to use as a guide if you plan to configure failover after you install.

Use the following preinstall checklist as a guide if you plan to configure failover after you install.

- Configure directory permissions so that the non-root user ID you are installing with has write permissions to the installation directory.
- Verify that your DNS configuration files, etc/hosts files, or both, on all servers can communicate with each other using host names.
- Verify that a reference to the current host's name uses the actual IP address and not the loopback address 127.0.0.1. On many systems, you can ping *<yourhostname>* and examine the IP address displayed as the command runs.
- Review any operating system and firewall settings to prevent blocking communication among servers.
- The same Dashboard Application Service Hub administrative userid (for example tbsmadmin) and password must be used for all TBSM Data and Dashboard servers in a failover configuration.

Failover considerations

There are a few additional considerations before performing the advanced installation.

- The primary and backup server must be running the same version and release of TBSM and the same operating system.
- Use the same non-root user ID to install TBSM on the primary and backup server to ensure that the user ID and password are the same for both machines. On Windows, choose the same user ID and password when prompted for the account.
- The primary and backup TBSM installation must use the same authentication method, either LDAP or Tivoli Netcool OMNIbus. The file-based authentication method is not supported for failover configurations.
- If you change any of the default port values during installation, use the same values for both installations.
- If you use the Netcool/OMNIbus ObjectServer for authentication and the ObjectServer is set up for failover, you need replicate the security information in the ObjectServer bi-directional gateways as described in the *Configuring failover* section of this guide.

For additional information about failover configuration see the *Configuring: Post Installation > Configuring failover* in this guide.

Failover and ObjectServer authentication considerations

If you intend to use IBM Tivoli Netcool OMNIbus ObjectServer authentication and you intend to use the Tivoli Netcool OMNIbus ObjectServer you are installing with TBSM as the authentication source, specify this during installation.

The TBSM failover configuration procedures and scripts do not modify the authentication configuration of TBSM nor do they arrange for replication of authentication information between the primary and backup ObjectServers. If you are using ObjectServer authentication, consult the OMNIbus documentation for information about setting up replication of such information using the bidirectional gateway.

IBM Tivoli Netcool Impact Considerations

In TBSM 6.2, Netcool/Impact 7.1.0.13 is prerequisite software and is not included in the TBSM installer.

The Netcool/Impact server that is required by TBSM must have server name TBSM for primary, or TBSM_B for secondary and cluster name TBSMCLUSTER.

IBM Tivoli Netcool OMNIBus Considerations

TBSM supports Tivoli® Netcool® OMNIBus version 8.1.0.5 and later. TBSM users are recommended to upgrade their TBSM's OMNIBus to its latest fix pack to continue receiving software support for problems pertaining the ObjectServer.

It is recommended for users to also upgrade the WebGUI component to the same version as the TBSM OMNIBus to maintain product synchrony.

Note: If you are installing Tivoli Netcool OMNIBus on Red Hat Enterprise Linux® AS, ES, or WS 5, you must ensure that the following files are available on your system before the installation:

- compat-libstdc++-33-3.2.3-61.i386.rpm
- libXp-1.0.0-8.i386.rpm
- openmotif22-2.2.3-18.i386.rpm
- libXmu-1.0.2-5.i386.rpm
- libXpm-3.5.5-3.i386.rpm
- compat-libstdc++-296-2.96-138.i386.rpm

These files should be available on the installation CDs for the operating system.

Adding the TBSM schema to the Object Server

Before installing TBSM Data server, **you need to add the TBSM schema and have the ObjectServer running before you install TBSM .**

You must know the host name and port number because you will be asked to provide them when you are installing TBSM .

Before you install TBSM, you have to modify the Netcool OMNIBus ObjectServer schema. The command and schema files are located in the TBSM install image in the directory:

```
\omnibus\schema_files
```

To update an existing Netcool OMNIBus ObjectServer schema for TBSM:

1. Copy the entire of the \omnibus\schema_files directory to the host where Netcool/OMNIBus is installed.
2. Open a command or shell window.
3. Change to the directory where you copied TBSM\omnibus\schema_files.

Note: Before running the `./import_schema_ksh.sh` and `./import_schema.sh` scripts on UNIX in the next step, you must ensure that the permissions set for them include execute. If they do not have execute permission set, run the following commands to add it now:

```
chmod +x import_schema_ksh.sh
chmod +x import_schema.sh
```

4. Run the `import_schema` command with the schema file **tbsm_db_update.sql** as a parameter.

On Windows, run the following command:

```
.\import_schema.bat %NCHOME% tbsm_db_update.sql
RAD ObjectServerName user password
```

On UNIX, run the following command:

```
./import_schema.sh $NCHOME tbsm_db_update.sql
RAD ObjectServerName user password
```

On AIX, run the following command:

```
./import_schema_ksh.sh $NCHOME tbsm_db_update.sql  
RAD ObjectName user password
```

Where:

- *NCHOME* is the value of the installation directory for Netcool/OMNIBus
- *tbsm_db_update.sql* is the name and location (if needed) of the schema file to read
- *RAD* is the schema validation string
- *ObjectName* is the name of your ObjectServer
- *user* is the user name for the ObjectServer
- *password* is the value of the ObjectServer password

In this example, *\$NCHOME* is `/opt/ibm/netcool`, *ObjectName* is `NCOMS`, *user* is `root`, and *password* is `mypass`, resulting in the following line:

```
./import_schema.sh /opt/ibm/netcool tbsm_db_update.sql RAD NCOMS root mypass
```

Note:

You may receive error messages similar to the following when running `tbsm_db_update.sql` command:

```
ERROR=Object exists on line 83 of statement  
'-----',  
  at or near 'BSM_Identity' (0 rows affected) (0 rows affected)  
(0 rows affected)  
ERROR=Object not found on line 15 of statement  
'-----'  
...', at or near 'service_deps'  
(0 rows affected)  
(0 rows affected)  
(0 rows affected)  
(0 rows affected)  
(0 rows affected)  
(0 rows affected)  
(0 rows affected)
```

These messages can be ignored. The ObjectServer will return an error if a user tries to add a column that already exists, which explains the error on `BSM_Identity`. It will also return an error if a user drops a table that doesn't exist, which explains the second error.

5. Run the **import_schema** command again, but specify **ClearServiceDeps.auto** as the schema file parameter as follows:

On Windows, run the following command:

```
.\import_schema.bat %NCHOME% ClearServiceDeps.auto  
RAD ObjectName user password
```

On UNIX, run the following command:

```
./import_schema.sh $NCHOME ClearServiceDeps.auto  
RAD ObjectName user password
```

On AIX, run the following command:

```
./import_schema_ksh.sh $NCHOME ClearServiceDeps.auto  
RAD ObjectName user password
```

Netcool/OMNIBus triggers

This topic describes the Netcool/OMNIBus triggers installed with TBSM.

Purpose

The triggers help manage event processing for Tivoli Netcool OMNIBus and TBSM.

Trigger descriptions

TBSM includes the triggers:

rad_update_fields_on_dedup

When TBSM updates events, this trigger specifies the deduplication field settings.

itm_deduplication

Deduplication processing for alert.status for IBM® Tivoli® Monitoring alerts only.

itm_event_clear

Tivoli Monitoring Event Problem/Resolution.

tec_deduplication

Deduplication processing for alerts.status for IBM Tivoli Enterprise Console® alerts only.

updatetecstatus

Update TECStatus field with status changes to Netcool/OMNIbus.

synchronizetec

Synchronize Tivoli Enterprise Console server with status/severity changes to Netcool/OMNIbus.

deletetec

Synchronize Tivoli Enterprise Console server with event deletions in Netcool/OMNIbus.

ClearServiceDeps

Removes all entries in the service_deps table that correspond to events that have been deleted in Netcool/OMNIbus. This automation is in the TBSM installation image and only used when you import the TBSM schema into an existing ObjectServer by hand.

Installation with external LDAP user registry considerations

TBSM Data Server

If you are using a Lightweight Directory Access Protocol (LDAP) product, you must select the file-based repository option during the installation of the Impact TBSM server and then manually configure the LDAP repository. For details, see the **Configuring LDAP** topic of the *Netcool/Impact Administration Guide*.

During the Data Server installation you will not be given the option to change to the LDAP user repository. If you want to use an LDAP user repository, you must configure it after the installation has completed.

To configure TBSM to authenticate to an LDAP repository, select **File Registry** as your user repository during installation. You can then configure TBSM to use an LDAP authentication source after the installation has completed.

When you install TBSM, the users `tbsmadmin` and `tbsmuser` and the groups, `tbsmAdmins`, `tbsmReadOnly`, `tbsmUsers`, and `tbsmViewAllServicesUsers`, are created in the target user repository. After installation, you can configure TBSM to use an external user repository.

TBSM DASH Server

If you want to create new users and groups from the user management interfaces in Dashboard Application Service Hub, you must configure the server with the target locations in which the new users and groups reside in the default repository. Although a federated repository in Dashboard Application Service Hub can read user and group information and can authenticate users from multiple registries, you can specify only one repository as the target for user and group creation. See [Configuring the default repository for the Dashboard server](#)

If you want to use an LDAP repository, you should configure the LDAP user repository in Dashboard Application Service Hub manually and then install TBSM Dashboard server.

Two users, `tbsmadmin` and `tbsmuser`, are created by TBSM Dashboard Server when it is installed. The default password for these users is the same as the password of the `impactadmin` user which was created as part of Netcool/Impact installation. Use these users or `tbsmadmin` to log in to TBSM for the first time. If you are installing the TBSM Data Server before installing the TBSM Dashboard Server, you need to create

the tbsmadmin user with the same password as the impactadmin user password in the external User repository or the ObjectServer, whichever is configured as the TBSM user repository.

When you install TBSM, the users tbsmadmin and tbsmuser and the groups tbsmAdmins, tbsmReadOnly, tbsmUsers, and tbsmViewAllServicesUsers are created in the target user repository. After installation, you can configure TBSM to use an external user repository. However, if you select a repository after installation that contains the same user ID or groups as those already in the default repository, you must remove any duplicate groups from the default repository before you configure the external repository. If you do not, you might not be able to log into TBSM using the default administrative user ID.

User registry considerations

The configuration options you have when you set up IBM Tivoli Business Service Manager for an external user registry.

If you are using a Lightweight Directory Access Protocol (LDAP) product, you must select the file-based repository option during the installation of the Impact server and then manually configure the LDAP repository by referring the Impact Administration guide. Also, you must configure JazzSM with LDAP user repository manually before installing TBSM Dashboard Server.

Users and user groups in external repositories

When you use an external repository for TBSM, all the users and user groups must be unique across the logical, federated repository in WebSphere. WebSphere provides the concept of a "Federated Repository" which is a single logical view of potentially multiple physical repositories that could all be connected to at the same time - LDAP, OMNIBus, and file registry.

You cannot have the same user or group name in both an external repository and the internal file-based repository. Otherwise, the user or members of the duplicated user groups cannot access TBSM. WebSphere cannot determine the correct login if the same user ID is defined in more than one user repository. For example, you cannot have a user ID tbsmadmin in both the file registry and also in your Netcool/OMNIBus repository.

When a user signs in to TBSM for the first time, the system looks up the user and the users group assignments in the external repository.

You assign user roles to an individual user or to a user group from the Dashboard Application Service Hub console or command-line tools. The user roles control the access privileges for each user and user group.

Manually configuring TBSM for external user repositories

For detailed information on configuring external user repositories, see the TBSM Administrator's guide here:

Administrator's Guide > Configuring TBSM > Manually configuring TBSM for external user repositories

Servers, failover, and external authentication

When you configure an external user registry, you need to configure the registry for each server in your configuration. For example, if you manually configure an LDAP server in a failover environment, you need to configure each of these servers to use the LDAP server:

1. Primary Dashboard server
2. Backup Dashboard server
3. Primary Data server
4. Backup Data server

Secure Sign-On and LDAP with other applications

If you configure Single Sign On between TBSM, and IBM Tivoli Change and Configuration Management Database (CCMDB) and Tivoli Monitoring, the, wasadmin user **must not** be an LDAP user. If the wasadmin user is in LDAP, you cannot configure the WebSphere Application Server for TBSM.

Uninstalling TBSM and the user registry

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.

Managing Netcool/OMNIbus events

If you are using another user registry such as LDAP, you do not need to set up the Netcool/OMNIbus ObjectServer as an external user registry to enable users to manage events. You cannot have the same user in two user registries, but you can add users to the ObjectServer without configuring it as an external user registry.

If you want to enable a user to manage ObjectServer events from Dashboard Application Service Hub, you must create a matching user in the ObjectServer that is configured for TBSM. The user names must match exactly in both the external user registry and in the ObjectServer. The ObjectServer user needs to have privileges equivalent to the default Normal user group in Netcool/OMNIbus.

For example, if you have a user named **jdoe1** in your LDAP user registry and **jdoe1** needs to manage ObjectServer events, you also need to have a user named **jdoe1** in your ObjectServer.

Use the Netcool Suite Administrator tool to create users for your ObjectServer. For more information on creating ObjectServer users, see *Using Netcool/OMNIbus Administrator to configure ObjectServers* in the information center for your version of Netcool/OMNIbus at:

<http://www.ibm.com/developerworks/wikis/display/tivolidoccentral/OMNIbus>

Dashboard server LDAP configuration

If you want to use an LDAP repository, you should configure the LDAP user repository in Dashboard Application Service Hub manually and then install TBSM Dashboard server.

Data server LDAP configuration

You need to install Netcool/Impact as a Filebased user repository and then manually configure the LDAP user repository using the available scripts before installing the TBSM Data Server.

Netcool/OMNIbus user registry

If you want to use Netcool/OMNIbusObjectServer as your external user registry, you can install it from the IBM Tivoli Business Service Manager installer. If you are familiar with the ObjectServer as a user registry, you may want to use this option.

You can configure ObjectServer as your external user registry from the TBSM installer. The installer prompts you for the ObjectServer information.

If you selected the file-based repository option during the installation, you can manually configure TBSM Data and Dashboard servers to use the ObjectServer repository.

Use Netcool/OMNIBus to access LDAP server

You can configure the Netcool/OMNIBusObjectServer to use an LDAP server as external user registry. If you are familiar with using LDAP as an external repository for an ObjectServer, you may want to use this option.

In this configuration, you configure your TBSM data and dashboard servers to use the ObjectServer as your user registry.

In Netcool/OMNIBus, you configure the Pluggable Authentication Module (PAM) for the LDAP server you want to access. When the ObjectServer is configured for LDAP, TBSM has access to LDAP data through the connection with the ObjectServer. The Websphere Application Server component of TBSM sees only a single authentication source.

Multiple user registry warnings

It is possible to use multiple user registries for a given TBSM server, but this method can cause authentication errors if you have the same user or user group in multiple user registries.

You can use more than one LDAP server or a combination of LDAP servers and a Netcool/OMNIBus ObjectServer. If you have multiple user registries, each user registry must have a unique set of users and user groups. The set of users and groups has to be unique across all repositories you configure for TBSM. That is, you cannot have the same user in two different user registries.

For example, if the user **jd**oe exists in two separate LDAP user registries (or in LDAP and an ObjectServer), the user **jd**oe cannot log in to TBSM or any other application installed in your Dashboard Application Service Hub instance.

Similarly, if you have a user group called **managers** in two separate user registries, users from in this group cannot authenticate to TBSM. Each registry's user group needs to have a unique set of users and these users need to be in the same registry as the user group.

Tivoli Monitoring and Change and Configuration Management Database user warning

If you set up a single user registry for TBSM, Tivoli Monitoring, and the Tivoli Change and Configuration Management Database (CCMDB), do not put special users such as `tbsmadmin` or `wasadmin` in the LDAP or Netcool/OMNIBus user registries. These users need to be authenticated within the application and the applications do not function if you put these users in an external user registry.

For example, if you have the `wasadmin` user in both CCMDB file-based repository and in LDAP, the Websphere Application Server cannot authenticate the `wasadmin` user.

TBSM users, user groups, and roles

IBM Tivoli Business Service Manager includes predefined users and user groups. Each of these groups assigns certain roles to members of the group.

Users and user groups

Two users, `tbsmadmin` and `tbsmuser1`, are created by TBSM Dashboard Server when it is installed. The default password for these users is the same as the password of the `impactadmin` user which was created as part of Netcool/Impact installation. Use these users or `tbsmadmin` to log in to TBSM for the first time. If you are installing the TBSM Data Server before installing the TBSM Dashboard Server, you need to create the `tbsmadmin` user with the same password as the `impactadmin` user password in the external User repository or the ObjectServer, whichever is configured as the TBSM user repository.

Important: The default login `tbsmadmin` is not defined in external repositories, such as LDAP or Netcool/OMNIBus, even though this default login is created during installation when an external repository is specified.

Users with a blank password cannot log in to the TBSM Dashboard server. The default password for the OMNIbus ObjectServer root user is null; therefore, if you want to log in as root and you have ObjectServer authority, you must specify a non-null password for the root user in the ObjectServer. If you need to change the ObjectServer password, use the procedure *Configuring TBSM > Changing the TBSM configuration > Changing the Netcool/OMNIbus ObjectServer password or user ID*.

The TBSM users and groups are created in the repository that you select during the installation for an advanced installation or in the Netcool/OMNIbus repository during a simple installation. For more information, see the *TBSM Installation Guide*.

You can assign users to the following predefined groups to define their level of access and authority in TBSM:

tbsmAdmins

Use this group for administrators. The roles assigned to this user group enable the group members to view and modify all TBSM objects in the graphical user interface (GUI).

tbsmUsers

Use this group for users who need to view all templates that are defined in the model.

tbsmViewAllServicesUsers

Use this group for users who only need to view all services that are defined in the model.

tbsmReadOnly

Use this group for users who you want to have only read-only access. By default, roles are assigned to this group that provide view-only capabilities. Users assigned to this group are restricted to the **Service Availability** page. These users cannot access administrative tasks.

By default, the tbsmadmin is assigned to the tbsmAdmins group and also to the WebGUI Netcool_OMNIbus_Admin group. The tbsmuser is assigned to the tbsmUsers group and also to the WebGUI Netcool_OMNIbus_Users group. If you perform an advanced installation and select the file registry as the file repository, the tbsmadmin user is also added to the tbsmAdmins, WebGUI Netcool_OMNIbus_Admin, and Netcool_OMNIbus_Users groups.

You can also manage user and group permissions for each service or service template in the TBSM GUI. For more information, see the *TBSM Service Configuration Guide*.

For information about changing the default service and template privileges for users and groups, see *Modifying the default service and template privileges* in the *Administrator's Guide*.

User roles

You can assign any of the following roles to users or groups. These roles specify the authority that users or groups have to view, modify, or administer TBSM settings.

<i>Table 1. TBSM user roles</i>	
Role	Authority assigned to user or group
tbsmAdminUser	Access to both the Service Availability and Service Administration pages in TBSM
tbsmSLAChartViewVisible	Assigned automatically by TBSM to the necessary users and groups. This role does not display in the list roles for users and groups. Do not assign this role manually.
tbsmViewRawEvents	View ObjectServer event lists. Note: This role is no longer used.
tbsmTemplateAdmin	Add, edit, delete, or view templates.
tbsmServiceAdmin	Add, edit, delete, or view services.
tbsmCreateTemplate	Add, edit, or view templates.

<i>Table 1. TBSM user roles (continued)</i>	
Role	Authority assigned to user or group
tbsmEditTemplate	Edit or view templates.
tbsmViewTemplate	View templates.
tbsmCreateService	Add or view services.
tbsmEditService	Edit or view services.
tbsmViewService	View services.
tbsmDataSourceAdmin	Add, edit, delete, or view data sources.
tbsmCreateDataSource	Add, edit, or view data sources.
tbsmEditDataSource	Edit or view data sources.
tbsmViewDataSource	View data sources.
tbsmDataFetcherAdmin	Add, edit, delete, or view data fetchers.
tbsmCreateDataFetcher	Add, edit, or view data fetchers.
tbsmEditDataFetcher	Edit or view data fetchers.
tbsmViewDataFetcher	View data fetchers.
tbsmChartAdmin	Add, edit, delete, or view charts.
tbsmCreateChart	Add, edit, or view charts.
tbsmEditChart	Edit or view charts.
tbsmViewChart	View charts.
tbsmViewDefinitionAdmin	Edit or delete view definitions. Note: The default view definitions are read-only and cannot be edited or deleted.
tbsmReadOnlyUser	Access to the Service Availability page only. This role is assigned by default to the tbsmReadOnly group; users assigned to that group automatically have this role.

DB2 Installation

TBSM requires IBM® DB2® 11.1 or higher. The latest maintenance packs should be applied. The TBSM databases may be installed on an existing DB2 instance or a new one can be created for the sole purpose of TBSM's use.

If you choose to create a failover TBSM solution, you can place the TBSM database into a high availability DB2 setup. For more information see the Failover setup section.

The TBSM database will need to be housed on a properly sized system. See the hardware requirements in this guide about TBSM machine requirements.

Disk space needs are primarily a function of the number of service instances known to the TBSM Data Server and to the resource, relationship, and attribute information stored in the TBSM Service Component Registry.

You should consider an install of the TBSM schema into a test environment for purposes of familiarizing yourself with the details of the TBSM database setup. In most environments this will not be necessary.

Software Product Compatibility Reports:

The most up-to-date information about supported hardware, software, browsers and operating systems is provided by the IBM [Software Product Compatibility Reports](https://www.ibm.com/software/reports/compatibility/clarify/index.html) at:

<https://www.ibm.com/software/reports/compatibility/clarify/index.html>

For more information about running the Software Product Compatibility Reports, see the Overview and Planning section of the [TBSM Wiki](#).

<https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/Tivoli%20Business%20Service%20Manager1>

DB2 Database Setup

TBSM provides an installation image that simplifies the setup and configuration of the TBSM database within the installed DB2 instance. When executing this part of the TBSM installation, you must be running as a DB2 user id with the full authority required for creating databases and all related artifacts such as buffer pools, table spaces, tables, and indexes. The installation application will allow you to immediately setup the database using the interface or produce setup scripts to disk that can be reviewed and executed at a later time. The setup will include the creation of all necessary database objects including databases, table spaces, schemas, buffer pools, tables and indexes.

At this point, you must determine the user that the runtime TBSM Data Server will use to connect to the DB2 database. Although it may be the same userid used to create the database setup, the primary requirement is to allow the runtime TBSM Data Server administrative access to the TBSM database. TBSM Service Component Repository is an intensive user of the database and it requires the ability to insert, update, drop and create indexes, and alter all TBSM database objects as well as run administrative commands through its SQL connection. For example, the SCR will frequently update table statistics by executing the following command -

```
CALL SYSPROC.ADMIN_CMD('RUNSTATS ON TABLE TBSMSCR.cdm_classIds AND INDEXES ALL');
```

This userid is required during the installation of the TBSM Data Server and Service Component Repository.

Database Disk Space Requirements

During the database installation procedure you select a *small*, *medium*, or *large* setting that best describes the environment – in terms of number of resources - that Tivoli Business Service Manager will manage. Follow these available disk space guidelines for TBSM data, depending on your choice.

- *Small* - 3G of disk space
- *Medium* – 6G of disk space
- *Large* – 10G of disk space

During the life of the database, monitor the database for resource usage, especially when service models are undergoing significant changes.

In addition to data requirements, TBSM requires database log space during operations. The default log settings for TBSM database are described in this table.

Description	Property setting
log file size (4-KB)	(LOGFILSIZ) = 32000
Number of primary log files	(LOGPRIMARY) = 6
Number of secondary log files	(LOGSECOND) = 10

The estimate for the required default log space is determined with the formula: $(6 + 10) * 4\text{-KB} * 32000 = 2\text{-GB log space}$

Some TBSM operations require extra logs, and in these cases the number of secondary logs is increased. For example, the Discovery Library Toolkit can require a secondary log number of up to 100. With these log settings, the log space requirement is 11.5-GB of disk space.

DB2 Administration and Maintenance

One of the advantages of TBSM using DB2 is that database administration and maintenance activities can be consolidated with other Tivoli product's that also use DB2 as its primary datastore. Consolidating database management can create a cost-efficient use of the skills required for common database administrative tasks such as starting and stopping a particular database, monitoring the health of the database, tuning, backing up and restoring, and maintenance of the database.

This guide will not attempt to document DB2 procedures to accomplish database administrative tasks since extensive documentation is available through IBM and is accessed via the internet. However, it is recommended that the TBSM administrator become familiar with basic DB2 concepts and administrative tasks. An overview of DB2 administration concepts is presented at the *IBM DB2 Version 11.1 for Linux, UNIX, and Windows* Information Center web site. You can access a complete set of documentation at https://www.ibm.com/support/knowledgecenter/SSEPGG_11.1.0/com.ibm.db2.luw.welcome.doc/doc/welcome.html . The *Database administration* section discusses essential administration concepts and tasks and is well worth a review.

As is generally recommended with the latter releases of DB2, TBSM recommends the use of DB2's Automatic Maintenance features for many of the maintenance and tuning functions of the relational database. The topics which follow discuss some key points about how TBSM uses the DB2 database.

Database Schema Descriptions

Since the TBSM database setup program, will optionally set up one to four databases, it is best to discuss the purposes of the database through the schemas that are created. For a simple install a single database holds all of the following schemas.

- The primary schemas will be the most demanding since they house the primary data typically associated with TBSM.
- TBSMBASE

This schema consists of the tables that are associated with the TBSM Data Server that contain information that includes service instance data as well as template and rule data. The size and volatility of these tables are most effected by the number of service instances that the TBSM Data Server persists and secondarily the number of templates and rules defined for those instances. Data access patterns are rather simple however in environments where services are frequently updated or fluctuate in number, the larger tables in this schema can quickly become fragmented.

- TBSMSCR

This schema consists of the tables that contain information imported into the TBSM Service Component Registry via TADDM, iDML books, or the SCR API. The demands on these tables are significant due to the nature of the processing of the Service Component Registry that is detailed in the *TBSM Customization Guide*. Data access patterns for the TBSMSCR and TBSMSAxy schemas are advanced and are most sensitive to database tuning.

- TBSMUDF

TBSM user defined functions (UDF) are defined within this schema. TBSM UDFs are an addition to the existing built-in DB2 functions to support TBSM-unique functions. The functions are implemented JAVA UDFs and are by default configured to run in fenced mode. See DB2 documentation for details concerning UDFs.

- TBSMSAxy

A set of schemas starting with 'TBSMSA' and ending with a combination of two characters, the first being the number of 1 or 2 and the second being one of the following characters T, B, A, and G, make up the SCR staging tables used for importing data. Multiple schemas provide each SCR import mechanism its own

stage area as indicated by the character value. The numerical value is a grouping mechanism that is used in failover configurations to allow each SCR server to have its own staging tables to minimize the chances of accidental corruption of data during unusual failover scenarios. These tables are very volatile since at any point large amounts of data waiting to be imported could be present. However, once the data in the staging tables has been reconciled with information in the TBSMSCR schema, the data it contains is no longer relevant. At the beginning of each import process, all data in the appropriate staging table is deleted.

- TBSM Configuration Artifact Schema
- TBSMCONFIG

This schema contains the artifact datastore used by the TBSM Import/Export feature described in the *TBSM Administration Guide*. In general, the demands on these tables are minimal since they hold configuration artifacts that are accessed infrequently. Care should be taken to not accidentally overlay an older version of the artifacts it contains during a database restore process.

- TBSM Schemas related to Metric History Collection and the Time Window Analyzer
- TBSMHISTORY

This schema houses metrics collected for the Time Window Analyzer portlet and are referred to as short-term history metrics. See the *TBSM's Administrator's Guide -> TBSM Metric Collection* for details on the feature. Specific database size considerations can be found in the *Important metric data store issues* section. Under heavy load, the tables in this schema will have large number of inserts, queries to support the Time Window Analyzer views, and deletes as a result of a pruning process.

- TWAMARKER

This schema also houses marker data that is related to the Time Window Analyzer portlet. See the *TBSM's Administrator's Guide -> TBSM Marker Repository Service* for details on the feature. Specific database size considerations can be found in the *Important issues concerning the marker data store* section. Access patterns are similar to the TWAHISTORY schema but with significantly less volume.

- Other Schemas
- EVENTRULES

- This schema contains rules collected to support the Tivoli Impact EIC feature. This schema should have minimal maintenance issues considering the relatively small amount of data it contains.

- TBSMDEMO

- This schema contains tables that are referenced in the TBSM documentation for product education purposes.

Other Important Considerations

- Since TBSM relies on its DB2 database, the TBSM Administrator should have access to the database via a database query tool such as the DB2 Control Center or other available database query tools. TBSM does not provide any type of generic query mechanism for administrative or support purposes.
- Though previous releases of TBSM provided a command line tool for resetting the TBSM database back to its state at installation, a similar task is accomplished through a best practice procedure. Once you have completed an initial install, take a backup of the TBSM databases for restore purposes when needed.
- The Discovery Library Toolkit will frequently update statistics on the TBSMSCR schema tables. However, it is recommended that all schema table statistics are periodically updated. See the DB2 RUNSTATS command.
- It is recommended that tables, especially in the TBSMBASE and TBSMSCR schemas, are monitored for table fragmentation or a plan is put in place to periodically reorganize them using the DB2 REORG function.

Chapter 6. Installing TBSM

About this task

The TBSM 6.2.0 Base release package comes with IBM Installation Manager version 1.8.x. If installing TBSM 6.2.0 on an environment with IBM Installation Manager 1.9.x, please note that the TBSM 6.2.0 Base release installation should be done using *only* the **Console** or **Silent** installers. Do not use the GUI installer scripts.

This section details a number of installation topics and scenarios.

Installation types

TBSM 6.2 supports only Advanced installation, this allows for the configuration of multiple servers.

You are provided with two separate installers, one for the TBSM Data Server and the other for the TBSM Dashboard Server which can be installed in any sequence in different servers, provided the prerequisites are already installed on those servers.

Installation order

The order in which you install the TBSM components is significant.

Always install the software features in the following order:

1. Set up your database schema with the TBSM Database Configuration utility. You need an instance of DB2 installed before you can run the Database Configuration utility.
2. Make sure that Netcool/Impact, Netcool/OMNIbus, JazzSM with DASH and WebGUI are installed before installing TBSM.
3. IBM Tivoli Business Service Manager Data server and Discovery Library Toolkit and IBM Tivoli Business Service Manager Dashboard server(s).

Note: TBSM Data and Dashboard server can be installed in any sequence, provided the pre-requisites are met. Each installer must be installed separately, one at a time.

DB2 schema configuration overview

The IBM Tivoli Business Service Manager database configuration utility creates the files needed to configure the Data Server, Metric Marker, and Metric History databases for TBSM. Optionally, the installer can create the schema in DB2 for the TBSM databases.

Prerequisites

You need to run the database configuration utility on the DB2 host where you want to install the TBSM database. The TBSM schema contains several user-defined functions (UDF's), and the jar file containing these functions must reside on the DB2 host. You need to know the ports for the DB2 instance.

Software Product Compatibility Reports:

The most up-to-date information about supported hardware, software, browsers and operating systems is provided by the IBM [Software Product Compatibility Reports](#) at:

<https://www.ibm.com/software/reports/compatibility/clarity/index.html>

For more information about running the Software Product Compatibility Reports, see the Overview and Planning section of the [TBSM Wiki](#).

<https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/Tivoli%20Business%20Service%20Manager1>

For more information on installing and using DB2, see the information center listed here for the version you are using:


https://www.ibm.com/support/knowledgecenter/SSEPGG_10.1.0/com.ibm.db2.luw.kc.doc/welcome.html

If you want to create the TBSM database schema during the installation, you must be logged on to the DB2 host as a user who has permissions to create database tables (SYSADM or SYSCTRL). Optionally, the installer can create the configuration files, and the `tbsm_db` script can be run to create the tables after the installation.

Windows command environment:

On Windows, the installer must be run from a command window that has been opened by the DB2 **db2cwadmin** script. In Windows 2016, if you log in as a user other than the Windows Administrator, you must use the DB2 command window option with the **Administrator** authority. Adding the user to the **Windows Administrators** group is not sufficient due to Windows 2016 User Access Control security functions. From the Windows start panel, click:

- **Start->All Programs->IBM DB2->Command Line Tools->Command Window -Administrator**
- Or, open a command window and enter the command: **db2cwadmin**.

Restriction:  You cannot install TBSM, DB2, or configure the DB2 databases for TBSM when you are logged in with a user id containing non-English characters. DB2 does not recognize a user id with non-English characters, such as Russian. For example, the Windows Administrator userid on a Russian operating system has Russian characters in the name. Typically, you do not have these problems in UNIX because the user names have English characters.

To install DB2, rename the user id and Administrators group to have English characters only. Log in with the renamed Administrator user id. Install DB2 while logged in with the renamed Administrator id. You also need to run the TBSM Database Configuration utility with the same renamed Administrator user id.

You must use the port used for DB2.

However, to install TBSM, you must log in with a user id with English characters that is a member of the Administrators group. The user id cannot be the renamed non-English Administrator id. You must log in with a user id that was originally created with English characters. One example is to use the `db2admin` user id that is normally created during a DB2 installation.

The installer creates the log file `.../tbsmdb/logs/db2_stdout.log`. This log file contains the output from all of the SQL that was executed. If there are any issues this log file is very helpful.

TBSM database user

The user that TBSM uses to connect to the DB2 database needs access to the TBSM database. Although it can be the same user ID used to create the database setup, the primary requirement is to allow the runtime TBSM administrator access to the TBSM database. TBSM Service Component Registry is an intensive user of the database and it requires the ability to insert, update, and alter all TBSM database objects as well as run administrative commands through its SQL connection.

As a best practice, the TBSM administrator needs access to the DB2 database via database query tool such as the DB2 Control Center or some other DB query tool that supports DB2. The Control Center or other tools are valuable for ongoing TBSM maintenance.

User name restriction: Ensure that the TBSM database user name does not contain a hyphen.

Database configuration installation LogPath consideration

The database configuration utility (for example, when running the `install_gui_dbconf` script using the **Installation Manager Wizard**) can create the TBSM databases for each database **Data Server**, **Metric Marker**, **Metric History** and **Demo**.

In the **Data Server** and **Metric History DB Config** panels, you can specify the new log path location for the databases. However, in the **Metric Marker** and **Demo DB Config** panels, you cannot specify the new log path location for the databases as it is not available.

The sequence of the configuration installation is **Data Server**, **Metric Marker**, **Metric History**, then **Demo**. By default, the **Data Server**, **Metric Marker**, and **Demo** databases all use the same database name TBSM. If they are not changed to use different database names, the setting of the new **LogPath** parameter for the **Data Server** database may subsequently be overwritten by the **Metric Marker** and **Demo** databases configuration during the installation process (using the default value for **LogPath**).

You can change the database log path after installation by setting the `newlogpath` parameter. For details, see [newlogpath - Change the database log path configuration parameter](#)

Creating database schema after installation

The database configuration install utility can create the TBSM schema information or just install the files used to create the schema. If you just install the files, you use the `tbsm_db` script after installation to create the schema. It will create the TBSM databases, indexes, views, and user-defined functions, and it updates specific performance-related DB2 configuration parameters.

Prior to executing `tbsm_db`, you can modify the parameters defined in the property files located in the `tbsmdb/sql` directory. A property file exists for each database (Data Server, Metric Marker, Metric History, and Demo). The values for the property files were set based on the user input during the installation (from the graphical user, console, or response file input). After the schema is created, you can use DB2 commands to update these configuration parameters and others not explicitly updated by the `tbsm_db`

script.

After the TBSM schema is created, you can also use `tbsm_db` script options to drop and recreate the schema.

Database Checker Utility

The TBSM Database Checker Utility verifies that the TBSM databases created with the Database Configuration utility are ready for use by the TBSM Data server.

Windows To run the utility on Windows systems:

1. Change to the directory: `<DBConfig_install> \tbsmdb\bin`

The default directory is:

```
C:\Program Files\IBM\tivoli\tbsmdb\bin
```

2. Enter the command:

```
TBSM_Check_DB.bat
```

UNIX To run the utility on UNIX systems:

1. Change to the directory: `<DBConfig_install> /tbsmdb/bin`

The default directory is:

```
opt/IBM/tivoli/tbsmdb/bin
```

2. Enter the command:

```
TBSM_Check_DB.sh
```

Follow the prompts to enter required user IDs and passwords, and to specify the databases to be checked.

Note: Passwords are not hidden when entered.

Example: Example of the command that is run in Windows.

```
C:\ibm\tivoli\tbsmdb\bin>TBSM_Check_DB.bat
Calling ant script tbsmDatabaseChecker.xml to check the databases
[input] Enter administrative database userid.
This user should have SYSADMIN or SECADMIN authority and will be used
to check authority of data server userid:
Administrator
[input] Enter the password for administrative database userid
Administrator: nothidden
[input] Enter database userid that will be used to connect the database
from the TBSM data server: [Administrator] tbsmdataseveruser
[input] Enter the password for data server database userid
tbsmdataseveruser: [nothidden] dspassword
[input] Enter database(s) to be checked:
[input] S = Service Model
[input] H = Metric History
[input] M = Metric Marker
[input] [ALL]
```

Advanced DB2 configuration

This topic describes the information you need to run Advanced DB2 configuration utility.

To configure the databases:

- Review DB2 configuration settings
- Install the DbConfig utility using the installer for the same

Advanced DB2 configuration settings

Specify the connection information for the database

Purpose

For each database you install, you need to supply the information described in this topic.

You are prompted for this information for the following databases used by TBSM. The information is saved in the following properties files.

```
tbsmdb\sql\tbsm_db.properties
tbsmdb\sql\tbsmudf_db.properties
```

Data server database

This is the primary database for TBSM service configuration. The configuration information for this database is stored in the files:

Metric markers database

The Time Window Analyzer metric marker database. The configuration information for this database is stored in the file:

```
tbsmdb\sql\tbsmmark_db.properties
```

Metric history database

The Time Window Analyzer metric history database. The configuration information for this database is stored in the file:

```
tbsmdb\sql\tbsmhist_db.properties
```


Choices

You can choose from the following options:

Database Name

The name of the database.

By default, this is set to TBSM for all the databases, except for the Metric History database, which has a default name of TBSMHIST.

Database Hostname or IP address

The host name of the system where the DB2 is installed.

By default, this is set to the host name of the local system.

Database Port

The database port number for DB2. The default is 50000.

Database User ID

The database user ID for DB2. This user must have permission to add and drop database tables.

Database password

Database users password. Confirm this in the **Confirm password** field.

Should the installer create the schema for this database

- If you select **Yes**, the installer configures the tables, tablespaces, and views in your DB2 instance.
- If you select **No**, the installer creates the configuration files for the tables, tablespaces, and views, and you install the configuration on your DB2 instance with the **tbsm_db** command.

Database path

The path used to create the database. The value <default> or a null value specifies the default database path specified by the database manager configuration.

If you want to use multiple paths, the first path must contain the database, and the paths must be separated by commas.

Table space configuration

Specify the 16K and 32K table space names for the database. The default names are:

Database	Default table space names
Data server	TBSM16KTS and TBSM32KTS.
Metric History	THM16KTS

Buffer pool configuration

Specify the 16K and 32K (Data server only) buffer pool names and sizes for the database.

Database	Default buffer pool names
Data server	TBSM16KBP and TBSM32KBP.
Metric History	THM16KBP
Demo/Sample	DEM16KBP

Transaction log configuration

Specify the transaction log configuration for the database. This includes the log buffer size, log file, size, number of primary logs, number of secondary logs, and the log file path. The default values will be based on the number of services you will have. You can view the default values for medium size installations (between 5000 and 20000 services) in the response file, dbconfig-installer.properties).

If the transaction log file size is too small, an error is generated:

```
SQL0964C The transaction log for the database is full. SQLSTATE=57011
```

This error is displayed in the TBSM trace log file or the Discovery Library Toolkit log. To update the transaction log size to an appropriate value, open the DB2 command window using the db2cmd command and execute the command:

```
UPDATE DATABASE CONFIGURATION FOR TBSM USING LOGSECOND <size>
```

where <size> is the new transaction log file size that you require.

Example: Log File Path

This example outlines a usage of the Log File Path parameter in the tbsm_db.properties file.

The following line sets **LogPath** property in the tbsm_db.properties file:

```
TBSM.LogPath=/opt/database/TBSM
```

Create the schema with the following command:

```
tbsm_db -s ds -f c -U db_userid -P db_pw
```

where ds is the TBSM data server and component registry database schema.

The command **db2 get db cfg for TBSM** can be used to examine the TBSM database configuration. The log file path information may show the following detail:

```
Changed path to log files (NEWLOGPATH) = /opt/database/TBSM/NODE0000/LOGSTREAM0000/  
Path to log files = /home/db2inst1/db2inst1/NODE0000/SQL00001/LOGSTREAM0000/
```

where:

Changed path to log files is the new log file path specified in the tbsm_db.properties file above.

Path to log files is the default log path.

To further optimize the configuration of the database, please estimate the expected number of service instances that will be managed

The database is configured according the size you specify here.

Size	Number of services	Disk space reserve
Large	More than 25,000	10 GB
Medium	5,000 to 25,000	6 GB
Small	Up to 5,000	3 GB

Running Advanced DB2 schema configuration utility

The Database Configuration Utility enables you to configure the DB2 schema for the Data Server, Metric Marker, and Metric History database in a separate databases.

Before you begin

Before you start the database configuration utility, read the *Advanced DB2 Configuration settings* topic for the settings you specify for the DB2 schema configuration.

About this task

This procedure shows how to run the TBSM Dbconfig installer which is IIM based and you can install in GUI, console or Silent mode.

Procedure

1. Download the platform specific TBSM 6.2 DBConfig installation media (dbconfig_<OS>.zip) and copy it onto the install directory and unzip.
2. Run the installer script for Dbconfig.

Note: You can use GUI mode, Console mode or Silent mode installation by choosing the respective installer script.

On Linux/Unix:

```
install_gui_dbconfig.sh
install_console_dbconfig.sh
install_silent_dbconfig.sh
```

On Windows:

```
install_gui_dbconfig.bat
install_console_dbconfig.bat
install_silent_dbconfig.bat
```

3. Select the installer package listed in the IM window.
- 4.
5. In the **Software License Agreement** window, click **I accept the IBM terms in the license agreement**, and then click **Next**.
6. In the **Install Path** field, type the fully qualified directory where you want to install the TBSM DbConfig files.

By default, this is set to the following locations:

```
UNIX /opt/IBM/tivoli/tbsmdb
Windows C:\Program Files\IBM\tivoli\tbsmdab
```

Directory restrictions: The directory names have these restrictions:

- 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.
 - Do not choose an installation directory name that contains an **accent** character (for example, . à, é, Ñ, ô). Otherwise, the installation fails.
7. Specify the database name and the other advance option parameters for the Data Server, Metric Marker, Metric History, and Demo databases.
 8. Select whether you want to create the schema in the database instance or if you just want install the configuration files on the host, and create the schema at a later time and click **Next**.
 9. Review the **Pre-Install Summary** and click **Install**.
 10. Click **Finish** when done.

Creating database schema after installation

If you decide to create the TBSM schema after you run the TBSM database configuration utility, you need to use the **tbsm_db** command to install the schema in your database instance.

Before you begin

Before you run the **tbsm_db** command, you need to install the TBSM database configuration files with the TBSM database configuration utility and you need to be logged in as a user who has permissions to create and drop tables in your database instance.

The **tbsm_db** command is in the `tbsmdb/bin` directory. For help, run the command:

```
tbsm_db -?
```

About this task

There are two types of commands you need to run with the **tbsm_db** script, one is for creating the databases and one is for creating the schema. The number of times you need to run the **tbsm_db** script depends on whether each of the TBSM databases resides within a single database or each TBSM database has its own separate database. The schema creation function of **tbsm_db** will be run once for each of the TBSM database schemas.

Procedure

1. To run the script, you need to have the correct permissions set, depending on your operating system.
 - Open a DB2 command window with the command: **db2cwadmin**.
 - Log in as a user with SYSADM or SYSCTRL authority.
2. Create the database with the command:

```
tbsm_db -s database -c
```

The *database* is the database you want to install in DB2.

Database	Value
Data server	ds
Metric marker	tm
Metric history	mh
Demo	de

For example, if you want to install the TBSM Data server database, the command is:

```
tbsm_db -s ds -c
```

Run this command for each database you created with the TBSM Database Configuration utility.

3. Create the schema with the command:

```
tbsm_db -s database_schema -f c -U db_userid -P db_pw
```

The *database_schema* is the database schema you want to install in DB2.

Database	Value
Data server	ds

<i>Table 7. TBSM database schema values (continued)</i>	
Database	Value
Metric marker	tm
Metric history	mh
Demo	de

For example, to create the metric marker schema, the command is:

```
tbsm_db -s tm -f c -U dbadmin -P dbapass123
```

When the command completes, you see the message:

```
DB20000I The SQL command completed successfully.
```

Run this command for each database schema you created with the TBSM Database Configuration utility.

Installing TBSM interactively

TBSM has an installation program that you can use to install the application interactively. TBSM 6.2 provides three separate installers to install the Database Configuration Utility, TBSM Data Server, and TBSM Dashboard Server.

Performing installations

When you perform an installation, you select the features to install on your TBSM system.

Before you begin

The following prerequisite software must be installed before installing TBSM 6.2.0, since these components are not part of the TBSM Installer package:

- IBM Tivoli Netcool/Impact 7.1 FP 13
- Jazz for Service Management 1.1.3.0 + CP5
- IBM DB2 Workgroup Server Edition 11.1.2.2
- IBM Tivoli Netcool/OMNIBus v8.1.0.15
- IBM Tivoli Netcool/OMNIBus 8.1.0.4-webgui FP12

For details about installing Netcool/OMNIBus, see https://www.ibm.com/support/knowledgecenter/en/SSHTQ_8.1.0/com.ibm.netcool_OMNIBus.doc_8.1.0/omnibus/wip/install/reference/omn_ins_im_omn_install_overview.html.

Procedure

- Select the installation procedure for the feature or features that you want to install.

Feature	Installation procedure
Data server	“Installing the Data Server component” on page 44
Dashboard server	“Installing the Dashboard server component” on page 55

Note: Network latency might cause delays when TBSM is loading or updating. To minimize potential network latency between the TBSM Data server and Dashboard server, and to promote optimal responsiveness in your network, locate your Data server and Dashboard servers in the same Local Area Network or network switch whenever possible.

Discovery Library Toolkit information

Specify information about the Discovery Library Toolkit installation in these prompts.

Purpose

Use the Discovery Library Toolkit to import data from:

- Tivoli Application Dependency Discovery Manager 7.2.2 or later

Note: If you want to integrate the Discovery Library Toolkit with Tivoli Application Dependency Discovery Manager 7.3, you must use TBSM 6.1.1 Fix Pack 1. You also need to use the migration utility to migrate your `classfilters.xml` and `NamingRules.xml` files. Discovery Library Toolkit also supports data import from Tivoli Application Dependency Discovery Manager 7.1.2 and above, however note that versions below 7.2.2 shall no longer be supported by the support team. If you decided to use Tivoli Application Dependency Discovery Manager 7.2 afterward, you need to use the same utility to revert the files to their previous versions. For more information about the utility and how to use it, see the TBSM developerWorks page at:

<https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Business%20Service%20Manager1/page/TBSM%20Discovery%20Library%20Toolkit%20Configuration%20-%20Integrating%20with%20TADDM%207.3>

- IBM common data model Discovery Library (IDML) books
- alternate name space books
- Discovery Library toolkit API
- Autopopulation rules.

During the simple installation you can choose whether Tivoli Application Dependency Discovery Manager and/or books will be accepted. The API is enabled by default, but can be disabled after installation through a property.

TADDM warning: If you are installing the Discovery Library Toolkit with a Tivoli Application Dependency Discovery Manager server as a data source, you need to make sure that you have only one Discovery Library Toolkit connection to the Tivoli Application Dependency Discovery Manager server. Otherwise, you will get incorrect data from the Tivoli Application Dependency Discovery Manager .

Note: If there is already a TBSM Discovery Library toolkit installed on this machine and it is installed into a non-standard location, this copy of the toolkit should be shutdown. A non-standard location is defined as not being in `./tivoli/tbsm/XMLtoolkit`. If a copy already resides at this location, the TBSM Data server install will skip the toolkit portion of the installation.

You also specify information to enable the toolkit to export Discovery Library books from TBSM.

Choices

You need to specify the data source information for the toolkit.

Please select the data source(s) that will be used.

Discovery Library books

The toolkit searches a directory you specify for new Discovery Library books (DLA files) and processes any files in that directory.

Tivoli Application Dependency Discovery Manager

If you are going to use Tivoli Application Dependency Discovery Manager as the source for your data, select this option. If you select this option, the installer prompts you for the server connection information.

Enter the naming service RMI registry port

The port number for the naming server RMI registry port.

The default is 12315.

Discovery Library Book Import Configuration

Configuration of the book import file system. Enter the directory name that the toolkit will monitor for new book files. When a new book file is detected in the directory you specify, the toolkit reads and processes the file.

Default: `$TBSM_HOME/tbsm/discovery/dlbooks`

TADDM Connectivity Configuration

If you selected the Tivoli Application Dependency Discovery Manager as a data source, the installer prompts you for the server information.

Enter the TADDM User ID

The Tivoli Application Dependency Discovery Manager user ID. Specify a user ID with at least supervisory authority.

Enter the TADDM password

The password associated with the user ID

Confirm the TADDM password

Re-enter the password.

Enter the TADDM server hostname or IP address

The host name or IP address of the system where the server is running. If the server is running on a private network and the TBSM server is not, then specify the external IP address of the TADDM server.

Enter TADDM port

The RMI port that Tivoli Application Dependency Discovery Manager (TADDM) is listening on. TADDM defines the RMI port in the `taddmInstall/etc/collation.properties` file. For TADDM versions earlier than 7.2.2, the property is `com.collation.api.port`. For TADDM versions 7.2.2 or later, the property is `com.ibm.cdb.service.registry.public.port`.

The default value is 9433.

Enter the SSL port that TADDM is listening on

The SSL RMI port that Tivoli Application Dependency Discovery Manager (TADDM) is listening on. TADDM defines the port in the `taddmInstall/etc/collation.properties` file. For TADDM versions earlier than 7.2.2, the property is `com.collation.api.ssl.port`. For TADDM versions 7.2.2 or later, the property is `com.ibm.cdb.service.registry.public.port`.

The default value is 9433.

Use SSL on the connection that TADDM is listening on

Specify whether you are using TADDM SSL listening port.

If you select SSL, you must copy the certificate `jssecacerts.certs` from the TADDM host to the Data server in this directory:

```
$TBSM_HOME\XMLtoolkit\sdk
```

You can do this after the installation is complete, but you need to copy the file before you start the toolkit.

TADDM Database Configuration

This set of prompts let you specify configuration definitions for the TADDM database.

Select database type

Specify **IBM DB2** or **Oracle**.

If you select **IBM DB2**, the JDBC files are found automatically in the TBSM installation package.

If you specify the **Oracle** database, the installer prompts you for the location of the directory containing the Oracle JDBC drivers. Check if the `$TBSM_HOME/dsa1ib` directory has the correct driver for your version of Oracle.

If you do not have this file on the Data server host, find the file and copy it to your system.

These files are typically provided by the database vendor with the database or the database client package. For example, you can find this file on your Oracle host system or as part of your Oracle client installation.

The installer looks in the directory you specify and collects the names of the jars that begin with "o" and end with ".jar" and adds these to the toolkit's classpath. If either `ojdbc6_g.jar`, `ojdbc6.jar` or `ojdbc5.jar` is found, it is added to the classpath; if not then all jars matching the pattern are added.

The reason for the additional checking on Windows is because problems have been seen with the 11.2.0.2.0 version of `ojdbc6.jar`. If this version of the JDBC driver is being used, it may be best to use `object6_g.jar` instead of `ojdbc6.jar`. The problem manifests itself on Windows with the following exception:

```
Exception in thread "main" java.lang.NoClassDefFoundError:
oracle.dms.console.DMSConsole
  at oracle.jdbc.driver.DMSFactory.<clinit>
    (DMSFactory.java:51)
  at java.lang.J9VMInternals.initializeImpl(Native Method)
  at java.lang.J9VMInternals.initialize(J9VMInternals.java:200)
  at oracle.jdbc.driver.PhysicalConnection.createDMSensors
    (PhysicalConnection.java:3821)
```

Enter the database User ID:

The TADDM database user id.

Note: The TADDM database user that TBSM is using needs only read authority on the TADDM database with two exceptions.

1. TBSM needs read/write/analyze permission on the `tbsm_change_history_table`. If this table has not yet been created, the DDL in `.../XMLtoolkit/sql/taddm_schema_setup_oracle.sql` or `taddm_schema_setup_db2.sql` can be used to create the table. TBSM uses this table during delta imports.
2. If the customer was using an older version of the toolkit that used the generated relationships (`taddm_explicitrel` script) in TADDM, these relationships must be deleted. After installing, run the `.../XMLtoolkit/bin/purgeexplicitrel` script to delete these relationships. The user used for this script needs write authority since it is deleting from the `relation`, `persobj`, and `cmdb_guid_alias` tables. Depending on the number of relationships that need to be deleted, this process can be lengthy. This is a one time process, so the user and password used by `purgeexplicitrel` can be different than that provided to TBSM for general use.

Note: On Oracle, the database user that TBSM uses requires read/write/analyze/truncate privileges to the table `tbsm_change_history_table` during delta import. Using the schema owner as the user is preferable to avoid table authorization issues during the toolkit import process. However, users who do not wish to use the schema owner but still want to enable delta import may use the property `DL_TADDM_DBManager.Schema_TCHT` to obtain access to the table instead. For more information please refer to the property description in the toolkit property file `xmltoolkitsvc.properties`.

Enter the database password:

The password for the user id

Confirm database password

Enter the password again to confirm.

Enter the database hostname:

The host name for the TADDM database

Enter the port used by the database

The default is 50000.

Enter the database name:

The default name is CMDB.

Enter the database schema

The default schema is `dbinst1`.

Discovery Library Book Export Configuration

Specify the toolkit export configuration settings on this screen.

Enter the directory name:

The directory where the toolkit writes book files created from the TBSM service models. This can be the same directory as import book directory where the toolkit reads books files.

Default: \$TBSM_HOME/tbsm/discovery/dlbooks

Enter dashboard server hostname or IP address

Enter the Dashboard server host name you want in the book files generated by TBSM. Other products use this information to enable a launch back to TBSM in context. That is, the other applications can launch a TBSM page from the specified Dashboard server.

If load balancing is set up for your Dashboard servers, specify the load balancer host.

If you enter the host name, enter the fully qualified name.

The default value is the Data server host IP address.

Enter the Dashboard server port:

The Dashboard sever HTTP port.

The default is 16310

Tivoli Event Integration Facility Probe Installation

About this task

The Probe for Tivoli EIF can receive Event Integration Facility (EIF) events sent from any Tivoli devices and sends them to the ObjectServer. This topic outlines the steps required to install and configure the Tivoli EIF Probe.

Procedure

1. Install the EIF Probe on the TBSM server. See:

https://www.ibm.com/support/knowledgecenter/en/SSSHTQ/omnibus/probes/all_probes/wip/concept/install_intro_install.html

2. Install the ITM Event Synchronization Component on the Netcool/OMNIbus ObjectServer. See:

https://www.ibm.com/support/knowledgecenter/en/SS3JRN_7.2.1/com.ibm.itm.doc_6.3/install/inst_itmsynch_comp.htm

3. Update the ObjectServer database for ITM events, and to start and stop the Situation Update Forwarder. See:

https://www.ibm.com/support/knowledgecenter/SSTFXA_6.3.0.1/com.ibm.itm.doc_6.3/install/inst_config.htm

4. Configure the Netcool/OMNIbus ObjectServer to receive events from ITM:

https://www.ibm.com/support/knowledgecenter/en/SSTFXA_6.3.0/com.ibm.itm.doc_6.3/install/omnibus_install.htm

5. Update the rules files to receive the events from ITM through probe:

https://www.ibm.com/support/knowledgecenter/SS3JRN_7.2.0/com.ibm.itm.doc_6.2.3/itm623_install876.htm#config_omni2_probe

Uninstalling the Tivoli Event Integration Facility probe

EIF Probe can be uninstalled independently of TBSM as it was not installed along with TBSM

About this task

See https://www.ibm.com/support/knowledgecenter/en/SSSHTQ_7.3.0/com.ibm.netcool_OMNIBus.doc_7.3.0/omnibus/wip/install/task/omn_ins_unixuninstallingprobesgtwys.html

Installing the Data Server component

You can use the installation program to install only the Data Server component, which includes the Discovery Library Toolkit. This is recommended in a production environment, where you want to install the Data Server component on a separate system from the other TBSM components.

In TBSM 6.2.0, the TBSM Data Server uses a separate installer with the name `data_<platform>.zip` which is available in a platform specific installer zip. Copy this zip to the server where the TBSM Data Server will be installed. Note, that the TBSM Data Server must have the Netcool/Impact server installed on the same system.

Before you begin

You need to do the following before install and configure these components:

- Configure the TBSM databases on a DB2 instance. You need to know the connection information for the database.
- Install IBM Tivoli Netcool/OMNIBus ObjectServer and configure the TBSM schema on an existing ObjectServer as described in the Planning section of this guide. The ObjectServer must be running before you install the Data Server. You need to know the connection information for the ObjectServer.
- Install Impact 7.1.0.13 with TBSM as the server name and TBSMCLUSTER as the cluster name and configure the required user repository (LDAP or Objectserver). The Impact Server and the Impact GUI Server should be running before installing Data Server.
- When installing Impact for the backup TBSM server, the server name must be TBSM_B and the cluster name TBSMCLUSTER.
- Read the sections that describe all the information you need to complete each screen in the installation program. Once you obtain all the information you need, fill out the Data Server installation worksheet and run the installer, using the worksheet as your guide.

Impact server details

In TBSM 6.2, since Impact is not installed along with TBSM, Netcool/Impact should be installed and running before installing the TBSM Data Server.

During the TBSM Data Server installation, in the 'Impact Server Details' panel, the Impact server hostname, install path and port numbers will be auto populated. User must enter the password for `impactadmin` user and `tbsmadmin` user.

Note: If you are installing the TBSM Data Server before installing the TBSM Dashboard Server, the 'tbsmadmin' user must be created in the user repository configured in Netcool/Impact before installing the Data Server.

TBSM 6.2 supports only Advanced installation.

If failover is set up and this Data Server will be the backup server, select the following check box.

If you are designating this host as the backup server in a failover environment, select the **Designated backup server** option.

Data server information

Specify the communication settings with the Data server.

Purpose

To successfully install the server, you need supply information that the Dashboard server needs to communicate with the Data server.

Restriction:

You cannot install a Backup Data Server and a Data Server at the same time. See **Configuring post installation > Configuring IBM Tivoli Business Service Manager for failover** for instructions about configuring a failover environment.

Choices

You can choose from the following options:

Communication port

The port number that the Dashboard Server components will use for communication with the Data Server component.

By default, this is set to 17542. The valid range for the port number is 1241–65535.

Impact Server Command Line Port

The port for the Netcool/Impact command line interface. The default value is 2000.

If failover is set up and this Data Server will be the backup server, select the following check box.

If you are designating this host as the backup server in a failover environment, select the **Designated backup server** option.

Note: If failover is selected and this server is used as the backup server, please ensure that the Impact server name is given as TBSM_B during installation of the Impact component for the backup TBSM server.

Database information

Specify information about the TBSM databases in this window.

Purpose

For each TBSM database you need to know the name, host, port, and user information. This database must be configured with the TBSM database configuration utility before you install TBSM.

To successfully install the server, you need supply information on the databases that was installed for TBSM using the database configuration utility.

Choices

You need to know this information for each database.

TBSM Database Information

For each TBSM database you need to know the name, host, port, and user information. This database must be configured with the TBSM database configuration utility before you install TBSM.

In many cases, all the TBSM databases share the same configuration. By default, the option **Use the same database as the TBSM Data Server** is selected for the Metric Marker and Metric History databases. If separate databases have been configured, de-select this option and enter the information for each database.

The Data Server database stores information such as services, templates, and the service component repository.

The Metric Marker database stores metric markers configured for overlaying historical values in the Time Window Analyzer.

The Metric History database stores the history of values for metrics that are collected for the Time Window Analyzer.

You need to know this information for each database.

Database name

The name for the database. The default is **TBSM**.

Database Hostname

The host name where the Data Server database is installed.

Database port Number

The default is **50000**.

Use the port number for the instance of DB2 where the TBSM database was configured.

Important: The TBSM installation program does not check if the port number is valid. As a result, you must validate the port number manually.

Database Username

The name of a user that has permission to update tables in the database.

Database password

The database user password.

Confirm Database password

Confirm the database user password.

Dashboard Application Service Hub Information

The user and password information for the administrative user who creates the Dashboard Application Service Hub profile in the Websphere Application Server. The default user name is `tbsmadmin`.

JazzSM

Note down the port number of JazzSM console.

ObjectServer configuration

Specify information about the Netcool/OMNIBus ObjectServer in this window.

Purpose

To successfully install the server, you need supply information the ObjectServer that sends events to Netcool/Impact. By default, the ObjectServer name is set to the host name of the system where you are running the installation program.

Important: You need to apply the TBSM schema changes before proceeding with the installation, See the Planning section for more information on Netcool/OMNIBus considerations.

Restriction: You can not specify the same ObjectServer for more that one TBSM Data server. Otherwise, your event data will be incorrect for both servers. The only time you can use the same ObjectServer is for the primary and backup Data servers in a failover environment.

Choices

You can choose from the following options:

ObjectServer host

The host name of the system where the ObjectServer is installed.

By default, this is the name of the local host where you are running the installer.

ObjectServer port

The port number for the ObjectServer.

By default, this is set to 4100. The valid range for the port number is 1024–65535. This port number must not be in use by another application.

ObjectServer User

The user name.

The default, this is set to root.

ObjectServer Password

The ObjectServer user password.

Note: TBSM can be installed with the default ObjectServer user as root and a null password, but users cannot use a null password to log in to the TBSM Dashboard Server. This is because the Integrated Solutions Console does not allow blank passwords.

Confirmation Password

Enter the ObjectServer password here again. If this does not match the **ObjectServer Password**, you are prompted to enter the password again.

TBSM Dataserver user registry information

Select the type of user management and authentication.

Purpose

To successfully install the server, you need supply information about your user registry. During the installation, choose one of the user registry options listed in the **TBSM DashServer User Registry Information** panel.

If the Impact Server is configured to use LDAP or the OMNIbus ObjectServer as the user repository, in the installation panel, you will not have any option to enter the user registry details. Instead a radio button will be auto selected to use the existing user repository. If Impact is configured to use file-based user repository, then this panel will give the option to change to the OMNIbus ObjectServer user repository by entering the OMNIbus server details.

Note: You will not be given the option to change to the LDAP user repository during the Data Server installation.

Jazz for SM server details

Specify the information about the JazzSM server details in this panel.

Purpose

The TBSM Data Server needs to connect with the TBSM Dashboard Server after installation. The Dashboard Application Service Hub, which the new TBSM UI is deployed on JazzSM application server. Enter the JazzSM host and the Port number where JazzSM server is running.

Discovery Library Toolkit information

Specify information about the Discovery Library Toolkit installation in the **Dataserver Datasource Information** panel of the installer.

Requirements

To use the Discovery Library Toolkit, you must install IBM Tivoli Business Service Manager 6.2.0.

To import business applications from Tivoli Application Dependency Discovery Manager 7.3, ensure that the `com.ibm.cdb.serviceinfrastructure.earlier.ver.compatibility` property in the `collation.properties` file is set to true. For Tivoli Application Dependency Discovery Manager servers that were upgraded to 7.3, the property is set to true by default.

In new installations of Tivoli Application Dependency Discovery Manager 7.3, the property is set to false by default. If you change the property from false to true, you need to generate the business applications again. You can wait for the scheduler to process all the grouping patterns or you can manually run the grouping patterns from the Tivoli Application Dependency Discovery Manager UI or command-line interface.

Purpose

Use the Discovery Library Toolkit to import data from:

- Tivoli Application Dependency Discovery Manager 7.2.2 or later.

Note: If you want to integrate the Discovery Library Toolkit with Tivoli Application Dependency Discovery Manager 7.3, you must use TBSM 6.1.1 Fix Pack 1. You also need to use the migration utility to migrate your `classfilters.xml` and `NamingRules.xml` files. Discovery Library Toolkit also supports data import from Tivoli Application Dependency Discovery Manager 7.1.2 and above, however note that versions below 7.2.2 shall no longer be supported by the support team. If you decided to use Tivoli Application Dependency Discovery Manager 7.2 afterward, you need to use the same utility to revert the files to their previous versions. For more information about the utility and how to use it, see the TBSM developerWorks page at:

<https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Business%20Service%20Manager1/page/TBSM%20Discovery%20Library%20Toolkit%20Configuration%20-%20Integrating%20with%20TADDM%207.3>

- IBM common data model Discovery Library (IDML) books
- alternate name space books
- Discovery Library toolkit API
- Autopopulation rules.

During the Data server installation you can choose whether Tivoli Application Dependency Discovery Manager and/or books will be accepted. The API is enabled by default, but can be disabled after installation through a property.

TADDM warning: If you are installing the Discovery Library Toolkit with a Tivoli Application Dependency Discovery Manager server as a data source, you need to make sure that you have only one Discovery Library Toolkit connection to the Tivoli Application Dependency Discovery Manager server. Otherwise, you will get incorrect data from the Tivoli Application Dependency Discovery Manager .

Note: If there is already a TBSM Discovery Library toolkit installed on this machine and it is installed into a non-standard location, this copy of the toolkit should be shutdown. A non-standard location is defined as not being in `./tivoli/tbsm/XMLtoolkit`. If a copy already resides at this location, the TBSM Data server install will skip the toolkit portion of the installation.

You also specify information to enable the toolkit to export Discovery Library books from TBSM.

Choices

You need to specify the data source information for the toolkit.

Please select the data source(s) that will be used.

Discovery Library books

The toolkit searches a directory you specify for new Discovery Library books (DLA files) and processes any files in that directory.

Tivoli Application Dependency Discovery Manager

If you are going to use Tivoli Application Dependency Discovery Manager as the source for your data, select this option. If you select this option, the installer prompts you for the server connection information.

Enter the naming service RMI registry port

The port number for the naming server RMI registry port.

The default is 12315.

Discovery Library Book Import Configuration

Configuration of the book import file system. Enter the directory name that the toolkit will monitor for new book files. When a new book file is detected in the directory you specify, the toolkit reads and processes the file.

Default: `$TBSM_HOME/tbsm/discovery/dlbooks`

TADDM Connectivity Configuration

If you selected the Tivoli Application Dependency Discovery Manager as a data source, the installer prompts you for the server information.

Enter the TADDM User ID

The Tivoli Application Dependency Discovery Manager user ID. Specify a user ID with at least supervisory authority.

Enter the TADDM password

The password associated with the user ID

Confirm the TADDM password

Re-enter the password.

Enter the TADDM server hostname or IP address

The host name or IP address of the system where the server is running. If the server is running on a private network and the TBSM server is not, then specify the external IP address of the TADDM server.

Enter TADDM port

The RMI port that Tivoli Application Dependency Discovery Manager (TADDM) is listening on. TADDM defines the RMI port in the `taddmInstall/etc/collation.properties` file. For TADDM versions earlier than 7.2.2, the property is `com.collation.api.port`. For TADDM versions 7.2.2 or later, the property is `com.ibm.cdb.service.registry.public.port`.

The default value is 9433.

Enter the SSL port that TADDM is listening on

The SSL RMI port that Tivoli Application Dependency Discovery Manager (TADDM) is listening on. TADDM defines the port in the `taddmInstall/etc/collation.properties` file. For TADDM versions earlier than 7.2.2, the property is `com.collation.api.ssl.port`. For TADDM versions 7.2.2 or later, the property is `com.ibm.cdb.service.registry.public.port`.

The default value is 9433.

Use SSL on the connection that TADDM is listening on

Specify whether you are using TADDM SSL listening port.

If you select SSL, you must copy the certificate `jssecacerts.certs` from the TADDM host to the Data server in this directory:

```
$TBSM_HOME\XMLtoolkit\sdk
```

You can do this after the installation is complete, but you need to copy the file before you start the toolkit.

TADDM Database Configuration

This set of prompts let you specify configuration definitions for the TADDM database.

Select database type

Specify **IBM DB2** or **Oracle**.

If you select **IBM DB2**, the JDBC files are found automatically in the TBSM installation package.

If you specify the **Oracle** database, the installer prompts you for the location of the directory containing the Oracle JDBC drivers. Check if the `$TBSM_HOME/dsa1ib` directory has the correct driver for your version of Oracle.

If you do not have this file on the Data server host, find the file and copy it to your system.

These files are typically provided by the database vendor with the database or the database client package. For example, you can find this file on your Oracle host system or as part of your Oracle client installation.

The installer looks in the directory you specify and collects the names of the jars that begin with "o" and end with ".jar" and adds these to the toolkit's classpath. If either `ojdbc6_g.jar`, `ojdbc6.jar` or `ojdbc5.jar` is found, it is added to the classpath; if not then all jars matching the pattern are added.

The reason for the additional checking on Windows is because problems have been seen with the 11.2.0.2.0 version of `ojdbc6.jar`. If this version of the JDBC driver is being used, it may be best to use `object6_g.jar` instead of `ojdbc6.jar`. The problem manifests itself on Windows with the following exception:

```
Exception in thread "main" java.lang.NoClassDefFoundError:
oracle.dms.console.DMSConsole
  at oracle.jdbc.driver.DMSFactory.<clinit>
    (DMSFactory.java:51)
  at java.lang.J9VMInternals.initializeImpl(Native Method)
  at java.lang.J9VMInternals.initialize(J9VMInternals.java:200)
  at oracle.jdbc.driver.PhysicalConnection.createDMSensors
    (PhysicalConnection.java:3821)
```

Enter the database User ID:

The TADDM database user id.

Note: The TADDM database user that TBSM is using needs only read authority on the TADDM database with two exceptions.

1. TBSM needs read/write/analyze permission on the `tbsm_change_history_table`. If this table has not yet been created, the DDL in `.../XMLtoolkit/sql/taddm_schema_setup_oracle.sql` or `taddm_schema_setup_db2.sql` can be used to create the table. TBSM uses this table during delta imports.
2. If the customer was using an older version of the toolkit that used the generated relationships (`taddm_explicitrel` script) in TADDM, these relationships must be deleted. After installing, run the `.../XMLtoolkit/bin/purgeexplicitrel` script to delete these relationships. The user used for this script needs write authority since it is deleting from the `relation`, `persobj`, and `cmdb_guid_alias` tables. Depending on the number of relationships that need to be deleted, this process can be lengthy. This is a one time process, so the user and password used by `purgeexplicitrel` can be different than that provided to TBSM for general use.

Note: On Oracle, the database user that TBSM uses requires read/write/analyze/truncate privileges to the table `tbsm_change_history_table` during delta import. Using the schema owner as the user is preferable to avoid table authorization issues during the toolkit import process. However, users who do not wish to use the schema owner but still want to enable delta import may use the property `DL_TADDM_DBManager.Schema_TCHT` to obtain access to the table instead. For more information please refer to the property description in the toolkit property file `xmltoolkitsvc.properties`.

Enter the database password:

The password for the user id

Confirm database password

Enter the password again to confirm.

Enter the database hostname:

The host name for the TADDM database

Enter the port used by the database

The default is 50000.

Enter the database name:

The default name is CMDB.

Enter the database schema

The default schema is `dbinst1`.

Discovery Library Book Export Configuration

Specify the toolkit export configuration settings on this screen.

Enter the directory name:

The directory where the toolkit writes book files created from the TBSM service models. This can be the same directory as import book directory where the toolkit reads books files.

Default: \$TBSM_HOME/tbsm/discovery/dlbooks

Enter dashboard server hostname or IP address

Enter the Dashboard server host name you want in the book files generated by TBSM. Other products use this information to enable a launch back to TBSM in context. That is, the other applications can launch a TBSM page from the specified Dashboard server.

If load balancing is set up for your Dashboard servers, specify the load balancer host.

If you enter the host name, enter the fully qualified name.

The default value is the Data server host IP address.

Enter the Dashboard server port:

The Dashboard sever HTTP port.

The default is 16310

Installation worksheet: Data Server

This worksheet helps guide you through the Data server installation.

Purpose

Fill out the worksheet before you run the installation program for the Data server. Save this worksheet for future reference.

Worksheet

You need to specify these settings:

Prompt title	Default value	Installed value
Installation Directory	/opt/IBM/tivoli/tbsm Windows C:\Program Files\IBM\tivoli\tbsm	
Data Server Information		
Communication port	17542	
Impact Server Command Line Port	2000	
Designated backup server	Not selected.	
Data Backup Information	Required for failover configuration	
Backup Data Server host		
Backup Data Server HTTP port	17310	
Backup Data Server communication port	17542	
User Registry Selection	ObjectServer	

Table 8. Data server installation information (continued)

Prompt title	Default value	Installed value
TBSM Data Server Database Information		
Database Name	TBSM	
Database Hostname		
Database Port Number	50000	
Database User Name		
Database Password		
TBSM Metric Marker Database Information		
Use the same database as the TBSM Data Server	Selected	
Database Name	TBSM	
Database Hostname		
Database Port Number	50000	
Database User Name		
Database Password		
TBSM Metric History Database Information		
Use the same database as the TBSM Data Server	Not selected	
Database Name	TBSMHIST	
Database Hostname		
Database Port Number	50000	
Database User Name		
Database Password		
Impact port information		
Starting port number provided	17310	
Modified port value file (optional)		
ObjectServer Configuration		
ObjectServer host	Local host name where installer is running.	
ObjectServer port	4100	
ObjectServer User	root	
ObjectServer Password		
Discovery Libray Toolkit Configuration		

Table 8. Data server installation information (continued)

Prompt title	Default value	Installed value
Discovery Library books data source	Selected	
Tivoli Application Dependency Discovery Manager data source	Not Selected	
Naming service RMI registry port	12315	
Discovery Library Book Import Configuration	\$TBSM_HOME/tbsm/discovery/dlbooks	
TADDM User ID	None	
TADDM password	None	
TADDM server hostname or IP address	None	
TADDM port	9443	
SSL port that TADDM is listening on	9433	
Use SSL on the connection that TADDM is listening on	Not selected.	
TADDM database type	IBM DB2	
Database User ID	None	
Database password:	None	
TADDM Database hostname	None	
TADDM database port	50000	
TADDM database name:	CMDB	
TADDM database schema	db2inst1	
Book Export directory	\$TBSM_HOME/tbsm/discovery/dlbooks	
Dashboard server hostname or IP address	Data server IP address	
Dashboard sever HTTP port.	16310	

Running the Data Server installer script

TBSM 6.2 provides the separate IIM based installer for Data Server, which can be installed in a separate system from the TBSM Data Server. This is recommended for a production environment where you want to install the Data Server component on a separate system from the other TBSM components.

Before you begin

Before you install the Data Server:

- Configure the TBSM databases on a DB2 instance. You need to know the connection information for the database.
- Install IBM Tivoli Netcool/OMNIbus ObjectServer and configure the TBSM schema on an existing ObjectServer as described in the Planning section of this guide. The ObjectServer must be running before you install the Data Server. You need to know the connection information for the ObjectServer.

- Install Impact 7.1.0.13 with TBSM as the server name and TBSMCLUSTER as the cluster name and configure the required user repository (LDAP or Objectserver). The Impact Server and the Impact GUI Server should be running before installing Data Server.
- Read the sections that describe all the information you need to complete each screen in the installation program. Once you obtain all the information you need, fill out the Data Server installation worksheet and run the installer, using the worksheet as your guide.

About this task

To install the Data Server:

Procedure

1. Download the platform specific TBSM 6.2 Data Server installation media (data_<OS>.zip) and copy it onto the install directory and unzip.
2. Run the installer script for Data Server.

Note: You can use GUI mode, Console mode or Silent mode installation by choosing the respective installer script.

On Linux/Unix:

```
install_gui_data.sh
install_console_data.sh
install_silent_data.sh
```

On Windows:

```
install_gui_data.bat
install_console_data.bat
install_silent_data.bat
```

3. Select the installer package listed in the IM window.
4. Read the Prerequisite disclaimer given and click **Next**.
5. Accept the IBM License Agreement and click **Next**.
6. Check the TBSM Data server package name listed and click **Next**.
7. In the **Install Path** field, a default path will be populated based on the Impact server install path. Verify the path and click **Next**.
8. In the **Impact Server Details** panel, verify the Impact server details auto-populated and enter the password for the `impactadmin` user and click **Next**.
9. In the **Data Server Information** panel, verify the ports auto-populated for Data Server.
10. Enter the TBSM Database details, in the respective database specific panels.
11. Enter the ObjectServer details in the **ObjectServer** panel.
12. Verify or choose the **TBSM Dataserver User** registry details in the respective panel.
13. Enter the host and port of the Jazz for SM server.
14. Enter the Discovery Library toolkit and TADDM server details in the respective panels.
15. Verify the preinstall summary and click **Install**.
16. Once the install is complete, click **Finish** when done.

Adding JDBC drivers to the shared library

Use this procedure to add a JDBC driver to the TBSM shared library.

About this task

Tivoli Business Service Manager supplies the following database JDBC drivers with this release:

- DB2
- HSQL
- Informix
- ObjectServer

If you want to use other databases as data sources, you need to obtain these drivers from the database manufacturer and copy them to your TBSM host.

These files are typically provided by the database vendor with the database or the database client package. For example, you can find the Oracle file on your Oracle host system or as part of your Oracle client installation.

Procedure

1. Obtain the appropriate JDBC driver according to the DSA specification.
2. Stop the server.
3. Copy the JDBC driver to the `$TBSM_HOME/impact/dsalib` directory.
This directory is created during the installation, and initially it is empty.
4. Restart the TBSM server.

What to do next

In a multi-host configuration you have to repeat this procedure for each server because JDBC drivers are not replicated between the servers. Stop the server while you are performing this procedure.

Installing the Dashboard server component

In TBSM 6.2, a separate IIM based installer for TBSM Dashboard server is provided. This installation is useful in a production environment, where you want to install the Dashboard server component on a separate system from the other IBM Tivoli Business Service Manager (TBSM) components.

Before you begin

Before you install the Dashboard server:

1. If TBSM Data Server is not installed before installing the Dashboard Server, you will get a warning for TBSM Data Server port not being active during the installation. But you can continue with the installation.
2. Install JazzSM with the Dashboard Application Service Hub.
3. Install OMNIbus WebGUI on JazzSM and configure it to use the proper ObjectServer (which is used as the Events server by TBSM Data Server).

Refer to the following link for details about OMNIbus WebGUI configuration using the config tool:
https://www.ibm.com/support/knowledgecenter/en/SSHTQ_8.1.0/com.ibm.netcool_OMNIbus.doc_8.1.0/webtop/wip/task/web_ins_im_configuimode.html
4. If you are planning to use LDAP as the user repository, make sure LDAP is configured as a user repository for JazzSM before installing the TBSM Dashboard server.
5. Complete the Dashboard server installation worksheet to help guide you through the installation.

OS Agent restriction: **Windows** If the Tivoli Monitoring Agent for Windows OS is installed and running on the same system as the Dashboard Application Services Hub server, it may lock certain Websphere Application Server dll files and cause the install to fail.

To avoid this problem, stop the agent before installing the server. To stop the agent:

1. In the **Manage Tivoli Enterprise Monitoring Services** application, select the **Monitoring Agent for Windows OS** service.

2. Select **Actions->Stop**.

You can also stop the Windows service for the agent:

In the Windows **Services** applet, stop both the "**Monitoring Agent for Windows OS - Primary**" and "**Monitoring Agent for Windows OS - Watchdog**" services

After the install has completed, you may restart the agent.

TBSM Dataserver user registry information

Select the type of user management and authentication.

Purpose

To successfully install the server, you need supply information about your user registry. During the installation, choose one of the user registry options listed in the **TBSM DashServer User Registry Information** panel.

If the Impact Server is configured to use LDAP or the OMNIbus ObjectServer as the user repository, in the installation panel, you will not have any option to enter the user registry details. Instead a radio button will be auto selected to use the existing user repository. If Impact is configured to use file-based user repository, then this panel will give the option to change to the OMNIbus ObjectServer user repository by entering the OMNIbus server details.

Note: You will not be given the option to change to the LDAP user repository during the Data Server installation.

Dashboard server configuration

Specify the communication settings with the Data server.

Purpose

To successfully install the server, you need to supply information that the Dashboard server needs to communicate with the Data server.

Restriction: You cannot install a Dashboard Server to connect to the Data Server that has been designated as a backup Data server. Please direct this dashboard server to connect to the primary Data server.

You cannot install a Backup Data Server and a Dashboard Server at the same time. See the Configuring post installation: Configuring failover and Load Balancing section of the TBSM Installation guide for instructions on configuring a failover environment.

Choices

You can choose from the following options:

Dashboard Server communication port

The port number that the Dashboard Server component uses for communication with the Data Server component.

By default, this number is set to 17543. The valid range for the port number is 1241–65535.

Data Server Host

The host name of the system where the Data Server is installed.

By default, this value is set to the host name of the local system.

Data Server HTTP Port

The port number for the Data server.

By default, this port is set to 17310. The valid range for the port number is 1241–65535.

Data Server communication port

The port number that the Data Server component uses for communication with the Dashboard Server component.

By default, this port is set to 17542. The valid range for the port number is 1241–65535.

Data Server HA/FailOver configured

If you have a failover environment, select this option and you will be prompted for information on the backup Data server.

Dashboard Backup Information

If you select the **Data Server HA/FailOver Configured** option, you need to provide this host and port information for the backup Data server on the next screen:

Backup Data Server host

The host name of the backup Data Server.

Backup Data Server HTTP port

Accept the default, or enter a port number.

By default, this is set to 17310. The valid range for the port number is 1241–65535.

Backup Data Server communication port

Accept the default, or enter a port number.

By default, this is set to 17542. The valid range for the port number is 1241–65535.

Data server Failover setup , Loadbalance configuration Information

When the **Data Server HA/FailOver Configured** checkbox is selected, you will be provided with one more checkbox options **Is Impact UI load balancing configured/planned for HA**. When checked, you need to enter the hostname and port of the IBM HTTP server which is used for load balancing.

JazzSM Server Details

Specify information about the Dashboard Application Service Hub administration user in this window.

Purpose

To successfully install the server, you supply the Dashboard Application Service Hub administration user information. This enables the installer to add a Dashboard Application Service Hub profile to the Websphere Application Server. The installer uses this user ID to log in to the Dashboard Application Service Hub server.

Settings

You need to specify these settings:

JazzSM Server Host

JazzSM server host name will be auto-populated.

JazzSM Install path

Enter the path where JazzSM is installed.

JazzSM Port Number

Enter the port number where JazzSM server is listening.

JazzSM User ID

The administrator user that is created and used to log into Websphere Application Server. If you are reusing an existing Dashboard Application Service Hub, provide the existing user ID.

Default: smadmin

JazzSM User password

The password for the user.

Confirm password

Enter the password here again. If this does not match the **DASH Password**, you are prompted to enter the password again.

tbsmadmin user password

You will be given the option of entering the password for the tbsmadmin user that will be created.

Omnibus WebGUI Home Location

You need to enter the path where OMNIBus WebGUI is installed in this field. Make sure that the path is the parent directory of omnibus_webgui directory. For example, if the WebGUI_HOME path is /opt/IBM/netcool/gui/omnibus_webgui, enter the path as /opt/IBM/netcool/gui in the **Omnibus WebGUI Home Location** field in the JazzSM server panel of the Dashboard server.

Note:

By default, you cannot use an ! (exclamation) character in the tbsmadmin password on Windows systems. It results in errors in the TBSM trace logs, RAD shell does not connect to the Data server, and no RAD shell prompt is displayed. For information about addressing this issue, see the Troubleshooting section of the *TBSM Installation Guide*.

If for any reason the tbsmadmin password changes (for example, if you switch from file-based user registry to LDAP), you must update the password details for the tbsmadmin user in the RAD_sla.props file. To do so, you must manually encrypt the password and then edit the RAD_sla.props file.

For example, edit the following file:

```
./JazzSM/profile/installedApps/JazzSMNode01Cell/isc.ear/sla.war/etc/  
RAD_sla.props
```

and add the following lines:

```
# VMM admin user credentials  
impact.sla.vmm.admin.username=tbsmadmin  
impact.sla.vmm.admin.password={AES}1C09D0508A96864E66E6AE731A68F005
```

ObjectServer configuration

Specify information about the Netcool/OMNIBus ObjectServer in this window.

Purpose

To successfully install the server, you need supply information the ObjectServer that sends events to Netcool/Impact. By default, the ObjectServer name is set to the host name of the system where you are running the installation program.

Important: You need to apply the TBSM schema changes before proceeding with the installation, See the Planning section for more information on Netcool/OMNIBus considerations.

Restriction: You can not specify the same ObjectServer for more that one TBSM Data server. Otherwise, your event data will be incorrect for both servers. The only time you can use the same ObjectServer is for the primary and backup Data servers in a failover environment.

Choices

You can choose from the following options:

ObjectServer host

The host name of the system where the ObjectServer is installed.

By default, this is the name of the local host where you are running the installer.

ObjectServer port

The port number for the ObjectServer.

By default, this is set to 4100. The valid range for the port number is 1024–65535. This port number must not be in use by another application.

ObjectServer User

The user name.

The default, this is set to root.

ObjectServer Password

The ObjectServer user password.

Note: TBSM can be installed with the default ObjectServer user as root and a null password, but users cannot use a null password to log in to the TBSM Dashboard Server. This is because the Integrated Solutions Console does not allow blank passwords.

Confirmation Password

Enter the ObjectServer password here again. If this does not match the **ObjectServer Password**, you are prompted to enter the password again.

Installation worksheet: Dashboard Server


This worksheet helps guide you through the Dashboard server installation.

Purpose

Fill out the worksheet before you run the installation program for the Dashboard server. Save this worksheet for future reference.

Worksheet

You need to specify these settings:

Prompt title	Default value	Installed value
Installation Directory	/opt/IBM/tivoli/ tbsmdash  C:\Program Files\IBM\tivoli \tbsmdash	
User Registry Selection	ObjectServer	
Dashboard Server Configuration		
Dashboard Server communication port	17543	
Data Server Host	Local host	
Data Server HTTP Port	17310	
Data Server communication port	17542	
Data Server is a primary in a failover configuration	Not selected	
Dashboard Backup Information	Required for failover configuration	
Backup Data Server host		
Backup Data Server HTTP port	17310	

<i>Table 9. Dashboard server installation information (continued)</i>		
Prompt title	Default value	Installed value
Backup Data Server communication port	17542	
JazzSM port information		
Starting port number provided	16310	
Modified port value file (optional)		
Name Server Configuration		
Host_Name:Port_Number	<i>TBSM Data_Server Host:17310</i>	
ObjectServer Configuration		
ObjectServer host	Local host name where installer is running.	
ObjectServer port	4100	
ObjectServer User	root	
ObjectServer Password		
Backup directory for Deployment Engine	Root directory	

Running the Dashboard Server installer script

This section describes how to start and run the Dashboard Server installation program.

Before you begin

Read the sections that describe all the information you need to complete each screen in the installation program. Once you obtain all the information you need, fill out the Dashboard Server installation worksheet and run the installer, using the worksheet as your guide.

Procedure

1. Download the platform specific TBSM 6.2 Dashboard Server installation media (dash_<OS>.zip) and copy it onto the install directory and unzip.
2. Run the installer script for Dashboard Server.

Note: You can use GUI mode, Console mode or Silent mode installation by choosing the respective installer script.

On Linux/Unix:

```
install_gui_dash.sh
install_console_dash.sh
install_silent_dash.sh
```

On Windows:

```
install_gui_dash.bat
install_console_dash.bat
install_silent_dash.bat
```

3. Select the installer package listed in the IM window and click **Next**.

4. Read the Prerequisite disclaimer given and click **Next**.
5. Accept the IBM License Agreement and click **Next**.
6. Check the TBSM Dashboard Server package name listed and click **Next**.
7. In the **Install Path** field, a default path will be populated. Verify the path and click **Next**.
8. Enter the host, path and port of the Jazz for SM server.
9. In the **Dashboard Server Information** panel, verify the auto-populated port number and click **Next**
10. In the **Impact Server Details** panel, enter the details of the Impact server where the TBSM Data Server is installed along with the password for the impactadmin user and click **Next**
11. Enter the details of the ObjectServer which is configured as an events server for the TBSM Dataserver in the **ObjectServer** panel.
12. Choose the **TBSM Dashboard Server User** registry details in the respective panel.
13. Verify the preinstall summary and click **Install**.
14. Once the install is complete, click **Finish** when done.

Installing TBSM in silent mode

A silent installation allows the TBSM installation to progress unattended.

Before you begin

To install TBSM in silent mode, you must modify the `setup.rsp` file that is included in the Installer package.

About this task

If the command does not start, you can display debugging information while you run it.

Displaying debugging information

Windows If the installer does not start in Windows, launch it while holding the **Ctrl** key to see debug information.

UNIX If the installer does not start in UNIX or LINUX, run the command: `export LAX_DEBUG=true` and rerun the installer.

Procedure

1. Download and unzip the TBSM installer (DBconfig, Data server or Dashboard server).
2. Update the sample response file in the path by giving input values based on the comments mentioned in the response file.

On Linux/Unix:

```
$IMAGE_HOME/scripts/sampleResponseFiles/DATA/<Platform>/Data_install_resp_<platform>.xml
$IMAGE_HOME/scripts/sampleResponseFiles/DASH/<Platform>/Dash_install_resp_<platform>.xml
$IMAGE_HOME/scripts/sampleResponseFiles/Dbconfig/<Platform>/
Dbconfig_install_resp_<platform>.xml
```

On Windows:

```
%IMAGE_HOME%\scripts\sampleResponseFiles\DATA\<Platform>\Data_install_resp_<platform>.xml
%IMAGE_HOME%\scripts\sampleResponseFiles\DASH\<Platform>\Dash_install_resp_<platform>.xml
%IMAGE_HOME%\scripts\sampleResponseFiles\DBConfig\<Platform>
\Dbconfig_install_resp_<platform>.xml
```

3. Run the command below from the installer package path.

On Linux/Unix:

```

./install_silent_data.sh
%IMAGE_HOME%/scripts/sampleResponseFiles/DATA/<Platform>/Data_install_resp_<platform>.xml
-acceptLicense

./install_silent_dash.sh
%IMAGE_HOME%/scripts/sampleResponseFiles/DASH/<Platform>/Dash_install_resp_<platform>.xml
-acceptLicense

./install_silent_dbconfig.sh
%IMAGE_HOME%/scripts/sampleResponseFiles/DBConfig/<Platform>/
DBConfig_install_resp_<platform>.xml
-acceptLicense

```

On Windows:

```

install_silent_data.bat
%IMAGE_HOME%\scripts\sampleResponseFiles\DATA\<Platform>\Data_install_resp_<platform>.xml
-acceptLicense

install_silent_dash.bat
%IMAGE_HOME%\scripts\sampleResponseFiles\DASH\<Platform>\Dash_install_resp_<platform>.xml
-acceptLicense

install_silent_dbconfig.bat
%IMAGE_HOME%\scripts\sampleResponseFiles\DBConfig\<Platform>
\DBConfig_install_resp_<platform>.xml
-acceptLicense

```

Dashboard Server response file

```

<?xml version='1.0' encoding='UTF-8'?>
<agent-input>
<!--
    When installing DASH Server, Installation Manager uses the share directory C:\Program Files\IBM\IBMIMShared
    If you want to use a different shared directory, uncomment this section and specify the sharedLocation directory you want to use

    <variables>
      <variable name='sharedLocation' value='/root/IBM/IBMIMShared' />
    </variables>

    <preference name='com.ibm.cic.common.core.preferences.eclipseCache' value='${sharedLocation}' />
-->

<server>
  <repository location='../output_dash' temporary='true' />
</server>

<profile id='IBM Tivoli Business Service Manager' installLocation='/opt/IBM/tivoli/tbsmdash'>
<!--*****
#JAZZ for SM Details
*****-->
<data key='user.DASHJazzSMHostName' value='localhost' />
<data key='user.JazzSMLocation' value='/root/IBM/JazzSM' />
<data key='user.DASHJazzSMPortNumber' value='26311' />
<!--JAZZ SM userid and password -->
<data key='user.DashServerJazzSMInfoUserID' value='smadmin' />
<data key='user.DashServerJazzSMInfoUserPassword' value='password' />
<data key='user.DashServerJazzSMInfoUserConfirmPassword' value='password' />
<data key='user.OmnibusWebGUIHome' value='/opt/IBM/netcool/gui' />
<!--*****
#Dashboard Server Communication
*****-->
<data key='user.DashServerCommunicationPortNumber' value='17543' />
<!--*****
#Impact Server Details
*****-->
<data key='user.ImpactHostNameForDash' value='localhost' />
<data key='user.ImpactRMIPortForDash' value='17542' />
<data key='user.ImpactHttpPortForDash' value='9080' />
<data key='user.ImpactHttpsPortForDash' value='9081' />
<data key='user.ImpactGUIHttpPortForDash' value='16310' />
<data key='user.ImpactGUIHttpsPortForDash' value='16311' />
<!-- Impact userid and password -->
<data key='user.ImpactUserForDash' value='impactadmin' />
<data key='user.ImpactPasswordForDash' value='password' />
<data key='user.ImpactConfirmPasswordForDash' value='password' />

<data key='user.DashFailoverCheckBox' value='false' />

<!--
    If HA/F0 is selected make DashFailoverCheckBox as true and uncomment below block.
-->
<!--
<data key='user.SecondaryImpactHostNameForDash' value='localhost' />
<data key='user.SecondaryImpactRMIPortForDash' value='17542' />
<data key='user.SecondaryImpactHttpPortForDash' value='9080' />
<data key='user.SecondaryImpactHttpsPortForDash' value='9081' />
<data key='user.SecondaryImpactGUIHttpPortForDash' value='16310' />
<data key='user.SecondaryImpactGUIHttpsPortForDash' value='16311' />
<data key='user.SecondaryImpactUserForDash' value='impactadmin' />
<data key='user.SecondaryImpactPasswordForDash' value='password' />
<data key='user.SecondaryImpactConfirmPasswordForDash' value='password' />

-->

<!--*****
#Dashboard Server Omnibus Details
*****-->
<data key='user.DashServerObjServerHost' value='localhost' />
<data key='user.DashServerObjServerPort' value='4100' />
<data key='user.DashServerObjServerUserID' value='root' />

```

```

<!--*****#TBSM Dashboard Server User Registry Details*****-->
<data key='user.DASHUserRegRadioButton' value='fileUserRegistryRadioButton' />

<!--
-->
In userregistry, if objectserver is selected comment above filebased block, and uncomment below block.
-->
<!--
<data key='user.DASHUserRegRadioButton' value='newObjectServerUserRegistryRadioButton' />
<data key='user.DashUserRegOSHostName' value='localhost' />
<data key='user.DashUserRegOSPortNumber' value='4100' />
<data key='user.DashUserRegOSUserID' value='root' />
-->

<!-- In userregistry, if LDAP is selected comment above filebased block and objectserver block, and uncomment below block.
-->
<!--
<data key='user.DASHUserRegRadioButton' value='ldapUserRegistryRadioButton' />
<data key='user.DashUserRegOSHostName' value='dc=IBM,dc=COM' />
<data key='user.DashUserRegLdapBaseDN' value='ou=SQLA,dc=HURSLEY,dc=IBM,dc=COM' />
-->

</profile>

<install>
<!-- IBM Tivoli Business Service Manager Dashboard Server 6.2.0.0 -->
<offering profile='IBM Tivoli Business Service Manager' id='com.ibm.tivoli.tbsm.dashboardserver' features='com.ibm.tivoli.DashServer' />
</install>

</agent-input>

```

Data Server response file

```

<?xml version='1.0' encoding='UTF-8'?>
<agent-input>
<!--
-->
When installing DASH Server, Installation Manager uses the share directory /root/IBM/IBMIMShared
If you want to use a different shared directory, uncomment this section and specify the sharedLocation directory you want to use

<variables>
<variable name='sharedLocation' value='/root/IBM/IBMIMShared' />
</variables>

<preference name='com.ibm.cic.common.core.preferences.eclipseCache' value='${sharedLocation}' />
-->

<server>
<repository location='../output_data' temporary='true' />
</server>
<profile id='IBM Tivoli Business Service Manager_1' installLocation='/opt/IBM/tivoli/tbsm'>
<!--*****#Impact Server Details*****-->
*****#Impact Server Details*****-->

<!--
-->
TBSM Services uses Netcool/Impact specific models, datasource, types and services. Please enter the Impact details.
-->
<data key='user.ImpactHostName' value='localhost' />
<data key='user.ImpactLocation' value='/opt/IBM/tivoli/impact' />
<data key='user.ImpactGuiPortNumber' value='16310' />
<data key='user.ImpactGuiHTTPSPortNumber' value='16311' />
<data key='user.ImpactBackEndPortNumber' value='9080' />
<data key='user.ImpactBackEndHTTPSPortNumber' value='9081' />
<data key='user.ImpactuserID' value='impactadmin' />
<data key='user.ImpactuserIDPWD' value='password' />
<data key='user.ImpactctuserIDCONFPWD' value='password' />

<!--
-->
If failover will be setup and this Data server will be the backup server, change DSFailoverCheckButtonUserData value as true.

Note: When you designate this data server as a backup server, it is only the first step in the data server failover-configuration process.
To finish the failover configuration, you need to run the fo_config script after you complete the installation. Do not install a dashboard
server to connect with this data server until the failover procedure has been completed on this machine. See the Configuring post
installation:Configuring failover and Load Balancing section of the TBSM installation guide for instructions on configuring a failover
environment.

-->
<data key='user.DSFailoverCheckButtonUserData' value='false' />

<!--*****#Data Server Information*****-->
*****#Data Server Information*****-->
<!-- Provide communication port for DataServer -->
<data key='user.DSCommunicationPortUserData' value='17542' />

<!--*****#TBSM Data Server Database Information*****-->
*****#TBSM Data Server Database Information*****-->
<!--
-->
Enter the information for the TBSM Data Server Database. This is the database where TBSM stores its services, templates, and the service
component registry
-->
<data key='user.DSDBName' value='TBSM' />
<data key='user.DSDBHostName' value='localhost' />
<data key='user.DSportNumber' value='50000' />
<data key='user.DSdbuserID' value='db2inst1' />
<data key='user.DSdbuserIDPWD' value='password' />
<data key='user.DSdbuserIDCONFPWD' value='password' />

<!--*****#TBSM Metric Marker Database Information*****-->
*****#TBSM Metric Marker Database Information*****-->
<!--
-->
Enter the information for the TBSM Metric History Database. This is the database TBSM uses to store the history of the values for metrics
that are configured for the collection and display with the Time Window analyzer
-->
<data key='user.MARKDBName' value='TBSM' />
<data key='user.MARKDBHostName' value='localhost' />
<data key='user.MARKportNumber' value='50000' />
<data key='user.MARKdbuserID' value='db2inst1' />
<data key='user.MARKdbuserIDPWD' value='password' />
<data key='user.MARKdbuserIDCONFPWD' value='password' />

<!--*****#TBSM Metric History Database Information*****-->
*****#TBSM Metric History Database Information*****-->
<!--
-->
Enter the information for the TBSM Metric History Database. This is the database TBSM uses to store the history of the values for metrics
that are configured for the collection and display with the Time Window analyzer
-->
<data key='user.HISTDBName' value='TBSMHIST' />

```

```

<data key='user.HISTDBHostName' value='localhost' />
<data key='user.HISTportNumber' value='50000' />
<data key='user.HISTdbuserID' value='db2inst1' />
<data key='user.HISTdbuserIDPWD' value='password' />
<data key='user.HISTdbuserIDCONFPWD' value='password' />

<!--*****
#Object Server Details
*****-->
<!--
TBSM will need an event source. By default, it will connect to Netcool/OMNIBus. Other types of event sources require manual configuration.
Please enter the Host and Port for your Netcool/OMNIBus server.
-->
<data key='user.OmniBUSHostName' value='localhost' />
<data key='user.OmnibusportNumber' value='4100' />
<data key='user.OmnibususerID' value='root' />

<!--*****
#TBSM DataServer User Registry Details
*****-->
<!--If UserRegistry is configured as filebased, then use below existingUserRegistryRadioButton block and comment
DSUserRegRadioGROUP block -->
<!-- <data key='user.DSUserRegRadioButton' value='existingUserRegistryRadioButton' /> -->

<!--If UserRegistry is configured as ObjectServer, then use below DSUserRegRadioGROUP block and comment
existingUserRegistryRadioButton block -->
<data key='user.DSUserRegRadioGROUP' value='DSUserRegRadioGROUP' />
<data key='user.DSUserRegRadioButton' value='DSUserRegRadioButton' />
<data key='user.DSUserRegOSHostName' value='localhost' />
<data key='user.DSUserRegOSPortNumber' value='4100' />
<data key='user.DSUserRegOSUserID' value='root' />

<data key='user.DSUserRegOSImpactAdminUserID' value='impactadmin' />

<!--*****
#JAZZ for SM Server Details
*****-->
<data key='user.JazzSMHostName' value='localhost' />
<data key='user.JazzSMPortNumber' value='26311' />

<!--*****
#DataServer DataSource and DB Selection
*****-->
<!--
The toolkit supports two data sources:Tivoli Application Dependency Discovery Manager (TADDMM) and Discovery Library Books.
-->
<data key='user.DataServerDLBCheckButton' value='true' />

<!--
If TADDMM is not to be selected, change DataServerTADDMMCheckButton value to false.
Note : If DataServerTADDMMCheckButton is false, TADDMM Connectivity Configuration and TADDMM Database Configuration sections has to be commented.
-->
<data key='user.DataServerTADDMMCheckButton' value='false' />
<data key='user.DataServerIsTaddmSelected' value='No' />
<data key='user.DataServerRMIRegistryPort' value='12315' />

<!--*****
#TADDMM Connectivity Configuration
*****-->
<!--
Configuration definitions for connecting to Tivoli Application Dependency Discovery Manager (TADDMM) server.
If a hostname is specified, it must be a fully qualified hostname.
-->

<!-- <data key='user.TADDMMConnectivityConfigUserID' value='administrator' />
<data key='user.TADDMMConnectivityConfigUserIDPWD' value='password' />
<data key='user.TADDMMConnectivityConfigUserIDCONFPWD' value='password' />
<data key='user.TADDMMConnectivityConfigHostName' value='localhost' />
<data key='user.TADDMMConnectivityConfigPortNumber' value='9530' />
<data key='user.TADDMMConnectivityConfigSSLPort' value='9531' /> -->

<!--
If SSL is not used on Connection with TADDMM server, use below block. Any one block should be available for dataserver installation.
Note: If SSL is selected, you must copy the certificate jssecacerts.cert from the TADDMM server to the TBSM Discovery Library toolkit server,
placing it in the directory $TBSM_HOME/XMLtoolkit/sdk/etc. This should be done after the installation is finished and before starting the
Discovery Library toolkit.
-->

<!-- <data key='user.TADDMMConfigureSSL' value='No' />
<data key='user.TADDMMConnectivityConfigRadioButton' value='noSSLForTADDMM' /> -->

<!-- If SSL is used on Connection with TADDMM server, use below block. Any one block should be available for dataserver installation. -->
<!--<data key='user.TADDMMConfigureSSL' value='Yes' />
<data key='user.TADDMMConnectivityConfigRadioButton' value='SSLForTADDMM' /> -->

<!--*****
#TADDMM Database Configuration
*****-->
<!--
Configuration definitions for connecting to the TADDMM database.
-->
If DB Type is DB2, comment above oracle values block and uncomment the below DB2 values block

<!-- <data key='user.TADDMMDBSELECTEDTYPE' value='DB2' />
<data key='user.TADDMMDBConfigRadioButton' value='Db2Button' />

<data key='user.TADDMMDBConnectionUserID' value='db2inst1' />
<data key='user.TADDMMDBConnectionUserIDPWD' value='password' />
<data key='user.TADDMMDBConnectionUserIDCONFPWD' value='password' />
<data key='user.TADDMMDBConnectionHostName' value='localhost' />
<data key='user.TADDMMDBConnectionPortNumber' value='50000' />
<data key='user.TADDMMDBConnectionDBName' value='taddm' />
<data key='user.TADDMMDBConnectionDBSchema' value='db2inst1' /> -->

<!-- If DB Type is Oracle, comment above DB2 values block and uncomment the below oracle values block -->
<!--
<data key='user.TADDMMDBSELECTEDTYPE' value='ORACLE' />
<data key='user.TADDMMDBConfigRadioButton' value='OracleButton' />

<data key='user.TADDMMDBConnectionUserID' value='oracleDB' />
<data key='user.TADDMMDBConnectionUserIDPWD' value='password' />
<data key='user.TADDMMDBConnectionUserIDCONFPWD' value='password' />
<data key='user.TADDMMDBConnectionHostName' value='localhost' />
<data key='user.TADDMMDBConnectionPortNumber' value='50000' />
<data key='user.TADDMMDBConnectionDBName' value='taddm' />
<data key='user.TADDMMDBConnectionDBSchema' value='oracleschema' />
-->

<!--*****
#Discovery Library Book Import Configuration
*****-->
<!--
Configuration of the Discovery Library Book import file system. This is the directory that the toolkit will monitor for new books, when a
new book is detected it will be read and processed.
-->
<data key='user.DataServerDLBImportConfig' value='/opt/IBM/tivoli/tbsm/discovery/dlbooks' />

<!--*****
#Discovery Library Book Export Configuration
*****-->

```

```

<!--
Configuration of the Discovery Library book export file system. This is the directory that the toolkit will write books to.
This can be the same directory that the toolkit reads books from.
-->
<data key='user.DataServerDLExportConfig' value='/opt/IBM/tivoli/tbsm/discovery/dlbooks' />

<!--
The following information will be contained in the Discovery Library book optionally generated by TBSM. This information may be used by
other products to launch-in-context back to TBSM.

If a load balancer has been used in conjunction with multiple dashboard servers, specify the load balancer. If a hostname is specified,
it must be a fully qualified hostname.
-->
<data key='user.DataServerDLExportDashBoardServerI' value='localhost' />
<data key='user.DataServerDLExportDashBoardServerPort' value='26311' />

<!-- User who runs the installer -->
<data key='user.DataServerInstallUser' value='root' />
</profile>
</install>
<!-- IBM Tivoli Business Service Manager Data Server 6.2.0.0 -->
<offering profile='IBM Tivoli Business Service Manager_1' id='com.ibm.tivoli.tbsm.Dataserver' features='com.ibm.tivoli.DataServer' />
</install>
</agent-input>

```

DBConfig response file

```

<?xml version='1.0' encoding='UTF-8'?>
<agent-input>

<!--
When installing DBConfigurator, Installation Manager uses the share directory $home/db2inst1/IBM/IBMIMShared
If you want to use a different shared directory, uncomment this section and specify the sharedLocation directory you want to use
</agent-input>
</agent-input>

<!--
When installing DBConfigurator, Installation Manager uses the share directory $home/db2inst1/IBM/IBMIMShared
If you want to use a different shared directory, uncomment this section and specify the sharedLocation directory you want to use
-->
</agent-input>
</agent-input>

<variables>
<variable name='sharedLocation' value='home/db2inst1/IBM/IBMIMShared' />
</variables>
<preference name='com.ibm.cic.common.core.preferences.eclipseCache' value='${sharedLocation}' />

-->

<server>
<repository location='../output_dbconf' temporary='true' />
</server>

<profile id='IBM Tivoli Business Service Manager' installLocation='/home/db2inst2/IBM/tivoli/tbsmdb'>

<!--*****
# TBSM Data Server Database configuration
*****-->
<data key='user.DBName' value='TBSM' />
<data key='user.hostName' value='localhost' />
<data key='user.portNumber' value='50000' />

<!--Data Server Database userid and password -->
<data key='user.dbuserID' value='db2inst1' />
<data key='user.dbuserIDPWD' value='password' />
<data key='user.dbuserIDCONFPWD' value='password' />

<!--
If Installer should create the schema for Data Server database set the variable radioButtonUserData to 'button1', otherwise set to 'button2'.
If 'button1' is selected database userid and password should provide.
-->
<data key='user.radioButtonUserData' value='button1' />

<!--
Database Path on which to create the database.
For Windows, this must be a drive letter (for example c:).
For Linux, this must be an absolute and existing path (for example /opt/IBM)
A null value or '<default>' will indicate that the default database path specified in the database manager configuration will be used.
If multiple Paths are specified, they must be comma separated and the Path containing the database must be the first Path specified.
-->
<data key='user.dbPath' value='<default>' />

<!--
The LogPath Name should be on a physical disk which does not have high I/O for optimal performance.
The log path must be an absolute path.
For Windows, this must include the drive letter and path separator (for example c:).
For Unix, it must begin with the path separator. (for example /opt/IBM)
A null value or '<default>' indicates the default path specified in the database manager configuration will be used.
-->
<data key='user.TBSM_LOG_PATH' value='<default>' />

<!--
Table Space Configuration (maximum 12 characters) -->
<data key='user.TBSM_16K_TABLE_SPACE_NAME' value='TBSM16KTS' />
<data key='user.TBSM_32K_TABLE_SPACE_NAME' value='TBSM32KTS' />

<!-- Buffer Pool Configuration (maximum 18 characters) -->
<data key='user.TBSM_16K_BUFFER_POOL_NAME' value='TBSM16KBP' />
<data key='user.TBSM_32K_BUFFER_POOL_NAME' value='TBSM32KBP' />

<!--
If No. of Instances is Large in Data Server DB, use below block for default values and comment remaining two blocks of values.
If you want to use different values, change the value section below according to the condition mentioned.

For Transaction Log Configuration
Database Heap used for Log Buffer (minimum value should be 256)
Log file Size (minimum value should be 16000)
Number of Primary Log files (minimum value should be 3)
Number of Secondary Log files (minimum value should be 2)
-->
<!-- Buffer Pool Configuration (maximum 18 characters) -->
<data key='user.TBSM_16K_BUFFER_POOL_SIZE' value='20000' />
<data key='user.TBSM_32K_BUFFER_POOL_SIZE' value='2000' />

<!-- Transaction Log Configuration -->
<data key='user.TBSM_DB_HEAP_FOR_LOG_BUFFER' value='512' />
<data key='user.TBSM_LOG_FILE_SIZE' value='64000' />
<data key='user.DEFAULT_TBSM_NO_PRIMARY_LOG_FILES' value='6' />
<data key='user.DEFAULT_TBSM_NO_SECONDARY_LOG_FILES' value='10' />

<!--
If No. of Instances is Medium in Data Server DB, use below block for default values and comment remaining two blocks of values.
If you want to use different values, change the value section below according to the condition mentioned.

For Transaction Log Configuration
Database Heap used for Log Buffer (minimum value should be 256)
Log file Size (minimum value should be 16000)
Number of Primary Log files (minimum value should be 3)
Number of Secondary Log files (minimum value should be 2)
-->
<!-- Buffer Pool Configuration (maximum 18 characters) -->
<!-- <data key='user.TBSM_16K_BUFFER_POOL_SIZE' value='15000' />
<data key='user.TBSM_32K_BUFFER_POOL_SIZE' value='2000' /> -->

```

```

<!-- Transaction Log Configuration -->
<!-- <data key='user.TBSM_DB_HEAP_FOR_LOG_BUFFER' value='384'/>
<data key='user.TBSM_LOG_FILE_SIZE' value='32000'/>
<data key='user.DEFAULT_TBSM_NO_PRIMARY_LOG_FILES' value='6'/>
<data key='user.DEFAULT_TBSM_NO_SECONDARY_LOG_FILES' value='10'/> -->

<!--
If No. of Instances is Small in Data Server DB, use below block for default values and comment remaining two blocks of values.
If you want to use different values, change the value section below according to the condition mentioned.

For Transaction Log Configuration
Database Heap used for Log Buffer (minimum value should be 256)
Log file Size (minimum value should be 16000)
Number of Primary Log files (minimum value should be 3)
Number of Secondary Log files (minimum value should be 2)
-->
<!-- Buffer Pool Configuration (maximum 18 characters) -->
<!-- <data key='user.TBSM_16K_BUFFER_POOL_SIZE' value='3000'/>
<data key='user.TBSM_32K_BUFFER_POOL_SIZE' value='1000'/> -->

<!-- Transaction Log Configuration -->
<!-- <data key='user.TBSM_DB_HEAP_FOR_LOG_BUFFER' value='256'/>
<data key='user.TBSM_LOG_FILE_SIZE' value='16000'/>
<data key='user.DEFAULT_TBSM_NO_PRIMARY_LOG_FILES' value='10'/>
<data key='user.DEFAULT_TBSM_NO_SECONDARY_LOG_FILES' value='2'/> -->

<!--*****
#TBSM Metric Marker Database configuration
*****-->
<data key='user.MarkerDBName' value='TBSM'/>
<data key='user.MarkerHostName' value='localhost'/>
<data key='user.MarkerportNumber' value='50000'/>

<!--Metric Marker Database userid and password -->
<data key='user.MarkerdbuserID' value='dbinst1'/>
<data key='user.MarkerdbuserIDPWD' value='password'/>
<data key='user.MarkerdbuserIDCONFPWD' value='password'/>

<!--
If Installer should create the schema for Data Server database set the variable radioButtonUserData to 'button1', otherwise set to 'button2'.
If 'button1' is selected database userid and password should provide.
-->
<data key='user.MarkerradioButtonUserData' value='button1'/>

<!--
Database Path on which to create the database.
For Windows, this must be a drive letter (for example c:).
For Linux, this must be a absolute and existing path (for example /opt/IBM)
A null value or '<default>' will indicate that the default database path specified in the database manager configuration will be used.
If multiple Paths are specified, they must be comma separated and the Path containing the database must be the first Path specified.
-->
<data key='user.MarkerdbPath' value='<default>'/>

<!-- Table Space Configuration (maximum 12 characters) -->
<data key='user.MARK_4K_TABLE_SPACE_NAME' value='TMM4KTS'/>

<!-- Buffer Pool Configuration (maximum 18 characters) -->
<data key='user.MARK_4K_BUFFER_POOL_NAME' value='TMM4KBP'/>
<data key='user.MARK_4K_BUFFER_POOL_SIZE' value='5000'/>

<!--
For Transaction Log Configuration
Database Heap used for Log Buffer (minimum value should be 256)
Log file Size (minimum value should be 16000)
Number of Primary Log files (minimum value should be 3)
Number of Secondary Log files (minimum value should be 2)
-->
<!-- Transaction Log Configuration -->
<data key='user.Marker_DB_HEAP_FOR_LOG_BUFFER' value='384'/>
<data key='user.Marker_LOG_FILE_SIZE' value='32000'/>
<data key='user.DEFAULT_Marker_NO_PRIMARY_LOG_FILES' value='6'/>
<data key='user.DEFAULT_Marker_NO_SECONDARY_LOG_FILES' value='10'/>

<!--*****
#TBSM Metric History Database configuration
*****-->
<data key='user.HISTDBName' value='TBSMHIST'/>
<data key='user.HISTDBHostName' value='localhost'/>
<data key='user.HISTDBportNumber' value='50000'/>

<!--Metric History Database userid and password -->
<data key='user.HISTDBdbuserID' value='dbinst1'/>
<data key='user.HISTDBdbuserIDPWD' value='password'/>
<data key='user.HISTDBdbuserIDCONFPWD' value='password'/>

<!--
If Installer should create the schema for Data Server database set the variable radioButtonUserData to 'button1', otherwise set to 'button2'.
If 'button1' is selected database userid and password should provide.
-->
<data key='user.radioButtonHISTUserData' value='button1'/>

<!--
Database Path on which to create the database.
For Windows, this must be a drive letter (for example c:).
For Linux, this must be a absolute and existing path (for example /opt/IBM)
A null value or '<default>' will indicate that the default database path specified in the database manager configuration will be used.
If multiple Paths are specified, they must be comma separated and the Path containing the database must be the first Path specified.
-->
<data key='user.HISTDBdbPath' value='<default>'/>

<!--
The LogPath Name should be on a physical disk which does not have high I/O for optimal performance.
The log path must be an absolute path.
For Windows, this must include the drive letter and path separator (for example c:).
For Unix, it must begin with the path separator.
A null value or '<default>' indicates the default path specified in the database manager configuration will be used.
-->
<data key='user.HIST_LOG_PATH' value='<default>'/>

<!-- Table Space Configuration (maximum 12 characters) -->
<data key='user.HIST_16K_TABLE_SPACE_NAME' value='TMH16KTS'/>

<!-- Buffer Pool Configuration (maximum 18 characters) -->
<data key='user.HIST_16K_BUFFER_POOL_NAME' value='TMH16KBP'/>

<!--
If No. of Instances is Large in Data Server DB, use below block for default values and comment remaining two blocks of values.
If you want to use different values, change the value section below according to the condition mentioned.

For Transaction Log Configuration
Database Heap used for Log Buffer (minimum value should be 256)
Log file Size (minimum value should be 16000)
Number of Primary Log files (minimum value should be 3)
Number of Secondary Log files (minimum value should be 2)
-->
<!-- Buffer Pool Configuration (maximum 18 characters) -->
<data key='user.HIST_16K_BUFFER_POOL_SIZE' value='20000'/>

<!-- Transaction Log Configuration -->
<data key='user.HIST_DB_HEAP_FOR_LOG_BUFFER' value='512'/>

```



```

<data key='user.HIST_LOG_FILE_SIZE' value='64000' />
<data key='user.DEFAULT_HIST_NO_PRIMARY_LOG_FILES' value='6' />
<data key='user.DEFAULT_HIST_NO_SECONDARY_LOG_FILES' value='10' />

<!--
If No.of Instances is Medium in Data Server DB, use below block for default values and comment remaining two blocks of values.
If you want to use different values, change the value section below according to the condition mentioned.
For Transaction Log Configuration
Database Heap used for Log Buffer (minimum value should be 256)
Log file Size (minimum value should be 16000)
Number of Primary Log files (minimum value should be 3)
Number of Secondary Log files (minimum value should be 2)
-->
<!-- Buffer Pool Configuration (maximum 18 characters) -->
<!-- <data key='user.HIST_16K_BUFFER_POOL_SIZE' value='5000' /> -->

<!-- Transaction Log Configuration -->
<!-- <data key='user.HIST_DB_HEAP_FOR_LOG_BUFFER' value='384' />
<data key='user.HIST_LOG_FILE_SIZE' value='32000' />
<data key='user.DEFAULT_HIST_NO_PRIMARY_LOG_FILES' value='6' />
<data key='user.DEFAULT_HIST_NO_SECONDARY_LOG_FILES' value='10' /> -->

<!--
If No.of Instances is Small in Data Server DB, use below block for default values and comment remaining two blocks of values.
If you want to use different values, change the value section below according to the condition mentioned.
For Transaction Log Configuration
Database Heap used for Log Buffer (minimum value should be 256)
Log file Size (minimum value should be 16000)
Number of Primary Log files (minimum value should be 3)
Number of Secondary Log files (minimum value should be 2)
-->
<!-- Buffer Pool Configuration (maximum 18 characters) -->
<!-- <data key='user.HIST_16K_BUFFER_POOL_SIZE' value='3000' /> -->

<!-- Transaction Log Configuration -->
<!-- <data key='user.HIST_DB_HEAP_FOR_LOG_BUFFER' value='256' />
<data key='user.HIST_LOG_FILE_SIZE' value='16000' />
<data key='user.DEFAULT_HIST_NO_PRIMARY_LOG_FILES' value='10' />
<data key='user.DEFAULT_HIST_NO_SECONDARY_LOG_FILES' value='2' /> -->

<!--*****
#TBSM Demo/Sample Database configuration
*****-->
<data key='user.DemoDBName' value='TBSM' />
<data key='user.DemohostName' value='localhost' />
<data key='user.DemoportNumber' value='50000' />

<!--Demo/Sample Database userid and password -->
<data key='user.DemodbuserID' value='db2inst1' />
<data key='user.DemodbuserIDPWD' value='password' />
<data key='user.DemodbuserIDCONFPWD' value='password' />

<!--
If Installer should create the schema for Data Server database set the variable radioButtonUserData to 'button1', otherwise set to 'button2'.
If 'button1' is selected database userid and password should provide.
-->
<data key='user.DemoradioButtonUserData' value='button1' />

<!--
Database Path on which to create the database.
For Windows, this must be a drive letter (for example c:).
For Linux, this must be a absolute and existng path (for example /opt/IBM)
A null value or '<default>' will indicate that the default database path specified in the database manager configuration will be used.
If multiple Paths are specified, they must be comma separated and the Path containing the database must be the first Path specified.
-->
<data key='user.DemodbPath' value='<default>' />

<!-- Table Space Configuration (maximum 12 characters) -->
<data key='user.DEMO_16K_TABLE_SPACE_NAME' value='DEM16KTS' />

<!-- Buffer Pool Configuration (maximum 18 characters) -->
<data key='user.DEMO_16K_BUFFER_POOL_NAME' value='DEM16KBP' />
<data key='user.DEMO_16K_BUFFER_POOL_SIZE' value='5000' />

<!-- User who runs the installer -->
<data key='user.InstallUserID' value='db2inst1' />

</profile>
<install>
<!-- IBM Tivoli Business Service Manager Database Configuration utility 6.2.0.0 -->
<offering profile='IBM Tivoli Business Service Manager' id='com.ibm.tivoli.tbsm.dbconfig' features='ant.feature' />
</install>
</agent-input>

```

Installing TBSM using the console

Use the console installation when the graphical user interface is not available. Console installation provides command-line prompts to input data.

About this task

TBSM installer package includes a separate script to install in Console mode.

UNIX For UNIX systems, at the command prompt, enter the following commands to install TBSM in console mode:

1. Unzip the platform specific TBSM Installers ZIP package.

Note: Each platform specific ZIP file contains separate ZIP files for Dbconfig, Data Server and Dashboard Server.

2. Use the following scripts to install TBSM Dbconfig, Data Server and Dashboard Server from the respective installer package path.
 - `install_console_dbconfig.sh`
 - `install_console_data.sh`
 - `install_console_dash.sh`

Windows

For Windows systems, you need to run the scripts as an administrator:

1. Unzip the platform specific TBSM Installers ZIP package.

Note: Each platform specific ZIP file contains separate ZIP files for Dbconfig, Data Server and Dashboard Server.

2. Use the following scripts to install TBSM Dbconfig, Data Server and Dashboard Server from the respective installer package path.
 - `install_console_dbconfig.bat`
 - `install_console_data.bat`
 - `install_console_dash.bat`

The console installation program prompts you for the same information as the GUI mode installation program.

Uninstalling TBSM

To uninstall TBSM 6.2, you need to uninstall the Data Server, Dashboard Server, and DbConfig, separately. DbConfig should be uninstalled last. You can uninstall in GUI mode, Console mode or Silent mode.

The uninstall procedure is only used to remove an installation that completes successfully. If the install fails before it completes, use the following below steps:

1. Uninstall the product from Installation Manager, to remove the registry entry created.
2. Delete the following folders if created (this depends on which installer failed):

On Linux:

```
/opt/IBM/tivoli/tbsm  
/opt/IBM/tivoli/tbsmdash  
/opt/IBM/tivoli/tbsmdb
```

On Windows:

```
C:\Program Files\IBM\tivoli\tbsm  
C:\Program Files\IBM\tivoli\tbsmdash  
C:\Program Files\IBM\tivoli\tbsmdb
```

Uninstalling TBSM Data Server

To uninstall TBSM Data Server, you need to login to the server using the same user that installed Data Server.

You can uninstall in GUI mode, Console mode or Silent mode.



Warning: There is a known issue whereby TBSM Data Server will not uninstall when the Impact Data Server is not running. To workaround this issue, start Impact before starting the uninstall procedure. If you cannot start Impact for whatever reason, contact the support team.

While uninstalling the Data Server in TBSM failover setup, you need to uninstall the Secondary Data server before uninstalling the Primary Data server.

GUI mode

Linux

To uninstall TBSM Data Server in GUI mode:

1. Launch the Installation manager from x-window terminal as the same user that installed Data Server.

```
cd /opt/IBM/InstallationManager/eclipse/  
./IBMIM
```

Installation Manager will launch.

2. Click **Uninstall**
3. Scroll the list of **Installed Packages** and select **IBM Tivoli Business Service Manager Data Server 6.2.0** and then click **Next**.
4. In the **Review Summary Information** window, click **Uninstall**.

Windows

To uninstall TBSM Data Server in GUI mode:

1. Login to windows server as the same user that installed Data Server.
2. Open a command window and enter the command:

```
cd C:\Program Files (x86)\ibm\Installation Manager\eclipse\
```

3. Run `IBMIM.exe`
Installation Manager will open.
4. Click **Uninstall**
5. Scroll the list of **Installed Packages** and select **IBM Tivoli Business Service Manager Data Server 6.2.0** and then click **Next**.
6. In the **Review Summary Information**, click **Uninstall**.

Console mode

Linux

To uninstall TBSM Data Server in Console mode:

1. Login to the server as the same user that installed Data Server.
2. Enter the following commands:

```
cd /opt/IBM/InstallationManager/eclipse/tools  
./imcl -c
```

3. Select the **Uninstall** option listed.
4. Select the **IBM Tivoli Business Service Manager Data Server 6.2.0**.
5. Select the **Uninstall** option and **Enter**.

Windows

To uninstall TBSM Data Server in Console mode:

1. Login to windows server as the same user that installed Data Server.
2. Open a command window and enter the command:

```
cd C:\Program Files (x86)\ibm\Installation Manager\eclipse\tools  
imcl -c
```

3. Select the **Uninstall** option listed.
4. Select the **IBM Tivoli Business Service Manager Data Server 6.2.0**.

5. Select the **Uninstall** option and **Enter**.

Silent mode

Linux

To uninstall TBSM Data Server in Silent mode:

1. Login to the server as the same user that installed Data Server.
2. Enter the following commands:

```
cd /opt/IBM/InstallationManager/eclipse/tools
./imcl -silent input $IMAGE_HOME/scripts/sampleResponseFiles/DATA/Linux
/Data_uninstall_resp_platform.xml
```

Windows

To uninstall TBSM Data Server in Silent mode:

1. Login to windows server as the same user that installed Data Server.
2. Open a command window and enter the command:

```
cd C:\Program Files (x86)\ibm\Installation Manager\eclipse\tools
imcl -silent input %IMAGE_HOME%\scripts\sampleResponseFiles\DATA\Win
\Data_uninstall_resp_platform.xml
```

Uninstalling TBSM Dashboard Server

To uninstall TBSM Dashboard Server, you need to login to the server using the same user that installed Dashboard Server.

You can uninstall in GUI mode, Console mode or Silent mode.

Note: Make sure the JazzSM server is up and running while uninstalling the Dashboard Server.

GUI mode

Linux

To uninstall TBSM Dashboard Server in GUI mode:

1. Launch the Installation manager from x-window terminal as the same user that installed Dashboard Server.

```
cd /opt/IBM/InstallationManager/eclipse/
./IBMIM
```

Installation Manager will launch.

2. Click **Uninstall**
3. Scroll the list of **Installed Packages** and select **IBM Tivoli Business Service Manager Dashboard Server 6.2.0** and then click **Next**.
4. In the **Review Summary Information** window, click **Uninstall**.

Windows

To uninstall TBSM Dashboard Server in GUI mode:

1. Login to windows server as the same user that installed Dashboard Server.
2. Open a command window and enter the command:

```
cd C:\Program Files (x86)\ibm\Installation Manager\eclipse\
```

3. Run **IBMIM.exe**

Installation Manager will open.

4. Click **Uninstall**
5. Scroll the list of **Installed Packages** and select **IBM Tivoli Business Service Manager Dashboard Server 6.2.0** and then click **Next**.
6. In the **Review Summary Information**, click **Uninstall**.

Console mode

Linux

To uninstall TBSM Dashboard Server in Console mode:

1. Login to the server as the same user that installed Dashboard Server.
2. Enter the following commands:

```
cd /opt/IBM/InstallationManager/eclipse/tools
./imcl -c
```

3. Select the **Uninstall** option listed.
4. Select the **IBM Tivoli Business Service Manager Dashboard Server 6.2.0**.
5. Select the **Uninstall** option and **Enter**.

Windows

To uninstall TBSM Dashboard Server in Console mode:

1. Login to windows server as the same user that installed Dashboard Server.
2. Open a command window and enter the command:

```
cd C:\Program Files (x86)\ibm\Installation Manager\eclipse\tools
imcl -c
```

3. Select the **Uninstall** option listed.
4. Select the **IBM Tivoli Business Service Manager Dashboard Server 6.2.0**.
5. Select the **Uninstall** option and **Enter**.

Silent mode

Linux

To uninstall TBSM Dashboard Server in Silent mode:

1. Login to the server as the same user that installed Dashboard Server.
2. Enter the following commands:

```
cd /opt/IBM/InstallationManager/eclipse/tools
./imcl -silent input $IMAGE_HOME/scripts/sampleResponseFiles/DATA/Linux
/Dash_uninstall_resp_<platform>.xml
```

Windows

To uninstall TBSM Dashboard Server in Silent mode:

1. Login to windows server as the same user that installed Dashboard Server.
2. Open a command window and enter the command:

```
cd C:\Program Files (x86)\ibm\Installation Manager\eclipse\tools
imcl -silent input %IMAGE_HOME%\scripts\sampleResponseFiles\DATA\Win
\Dash_uninstall_resp_<platform>.xml
```

Uninstalling DbConfig

To uninstall TBSM DbConfig, you need to login to the server using the same user that installed DBconfig installer

You can uninstall in GUI mode, Console mode or Silent mode.

GUI mode

Linux

To uninstall TBSM DbConfig in GUI mode:

1. Launch the Installation manager from x-window terminal.

```
cd /home/db2inst1/IBM/InstallationManager/eclipse/  
./IBMIM
```

Installation Manager will launch.

2. Click **Uninstall**
3. Select the TBSM DbConfig package listed and click **Next**
4. Click the **Uninstall** button
5. After the uninstall is completed successfully, delete the TBSM and TBSMHIST databases manually

Windows

To uninstall TBSM DbConfig in GUI mode:

1. Login to windows server as user db2.
2. Open a command window and enter the command:

```
db2cwadmin  
cd C:\Program Files (x86)\ibm\Installation Manager\eclipse\
```

3. Run IBMIM.exe

Installation Manager will open.

4. Click **Uninstall**
5. Select the **IBM Tivoli Business Service Manager Database Configuration Utility 6.2.0** and click **Next**
6. In the **Review Summary Information**, click the **Uninstall** button.
7. After the uninstall is completed successfully, delete the TBSM and TBSMHIST databases manually

Console mode

Linux

To uninstall TBSM DbConfig in Console mode:

1. Login to the server as user db2.
2. Enter the following commands:

```
cd /home/db2inst1/IBM/InstallationManager/eclipse/tools  
./imcl -c
```

3. Select the **Uninstall** option listed.
4. Select the **TBSM DbConfig** and click **Next**
5. Select the **Uninstall** option.
6. After the uninstall is completed successfully, delete the TBSM and TBSMHIST databases manually

Windows

To uninstall TBSM DbConfig in Console mode:

1. Login to windows server as user db2.
2. Open a command window and enter the command:

```
db2cwadmin
cd C:\Program Files (x86)\ibm\Installation Manager\eclipse\tools
imcl -c
```

3. Select the **Uninstall** option listed.
4. Select the **TBSM DbConfig** and click **Next**
5. Select the **Uninstall** option.
6. After the uninstall is completed successfully, delete the TBSM and TBSMHIST databases manually

Silent mode

Linux

To uninstall TBSM DbConfig in Silent mode:

1. Login to the server as user db2.
2. Enter the following commands:

```
cd /home/db2inst1/IBM/InstallationManager/eclipse/tools
./imcl -silent input $IMAGE_HOME/scripts/sampleResponseFiles/DATA/Linux
/DBConfig_uninstall_resp.xml
```

3. After the uninstall is completed successfully, delete the TBSM and TBSMHIST databases manually

Windows

To uninstall TBSM DbConfig in Silent mode:

1. Login to windows server as user db2.
2. Open a command window and enter the command:

```
db2cwadmin
cd C:\Program Files (x86)\ibm\Installation Manager\eclipse\tools
imcl -silent input %IMAGE_HOME%\scripts\sampleResponseFiles\DATA\Win
\DBConfig_uninstall_resp_<platform>.xml
```

3. After the uninstall is completed successfully, delete the TBSM and TBSMHIST databases manually

Reinstalling TBSM

Before performing the reinstall after the successful uninstallation, make sure you restart the Impact server, Impact GUI server, JazzSM server and OMNIbus ObjectServer. Also ensure that previous TBSM installation directories are deleted.

Reinstalling TBSM after a failed installation

If the TBSM installer fails, you can cleanup the system and try to reinstall.

Procedure

1. Uninstall the product from Installation Manager, to remove the registry entry created.
2. Delete the following folders if created (this depends on which installer failed):

On Linux:

```
/opt/IBM/tivoli/tbsm  
/opt/IBM/tivoli/tbsmdash  
/opt/IBM/tivoli/tbsmdb
```

On Windows:

```
C:\Program Files\IBM\tivoli\tbsm  
C:\Program Files\IBM\tivoli\tbsmdash  
C:\Program Files\IBM\tivoli\tbsmdb
```

3. Restart Impact server, Impact GUI server, JazzSM server and ObjectServer.

Chapter 7. Restoring system from a backup

TBSM 6.2 is a new full install and TBSM 6.1.1 cannot be restored if the install fails. You will have to reinstall TBSM 6.2.

Chapter 8. Installing and configuring the TBSM Agent

The Tivoli Business Service Management agent (TBSM agent) is an IBM Tivoli Monitoring distributed agent using a distributed monitoring server.

These topics describe how to install and configure the agent. They also describes the components of the agent such as its work spaces, attribute groups and predefined situations.

Installing the TBSM agent

The Tivoli Business Service Management agent (TBSM agent) is an IBM Tivoli Monitoring distributed agent using a distributed monitoring server.

Before you begin

If there are no other agents or Tivoli Monitoring components installed prior to installing the TBSM agent, then the TBSM agent can be installed as root or non-root. In this case, shared library files will have to correct permissions.

However, if you previously installed other agents or Tivoli Monitoring components, you need to install the TBSM agent as the same user that installed the other Tivoli Monitoring components. Otherwise, permissions on files such as shared libraries may be incorrect, and the TBSM agent may not start.

You can install and deploy the TBSM agent just like any other Tivoli Monitoring agent in your environment. The topics in this section describe the TBSM-specific information you need to know to install and deploy the agent. For more information on installing and deploying Tivoli Monitoring agents, see [Tivoli Monitoring information center](#).

Important: It is easier to install the agent as the `root` user on UNIX systems. If you want to install as a non-root user, see the Tivoli Monitoring information center for detailed instructions.

About this task

Install the TBSM agent if you have IBM Tivoli Monitoring installed at your location and you want to do either or both of the following:

- Use IBM Tivoli Monitoring to monitor the status of TBSM.
- Use the TBSM Historical Reporting function, which uses the Data Warehouse feature of IBM Tivoli Monitoring to record historical TBSM data.

You can install and deploy the TBSM agent just like any other Tivoli Monitoring agent in your environment. The topics in this section describe the TBSM-specific information you need to know to install and deploy the agent. For more information on installing and deploying Tivoli Monitoring agents, see [Tivoli Monitoring information center](#).

To install the TBSM agent:

Procedure

1. Download the TBSM Agent installer as it is not included in the TBSM 6.2 package.
2. Unzip the installer package and start the installation:
 - On UNIX, run the `install.sh` script.
 - On Windows, run `setup.exe` as Administrator.
3. Follow the prompts of the installation using the following reference sections as your guide for TBSM-specific information. For most of the prompts, the installation program provides all the information and instructions you need to install the agent.

Important: When you are prompted for the **IP.PIPE host or IP Address** value, the default value is the local host name. This value needs to be the host where the Tivoli Enterprise Monitoring Server (TEMS) is installed. Otherwise, the agent will not work. If necessary, correct the **IP.PIPE host or IP Address** value from the **Manage Tivoli Monitoring Services** window. Right-click the agent and select **Reconfigure** from the menu that opens.

4. After you install the TBSM Agent and TBSM, you must stop and start the TBSM data server to enable the connection between TBSM and the agent.

Configuring the TBSM agent

The following configuration items must be performed before using the TBSM agent or the TBSM reporting system.

About this task

The TBSM Agent must be configured, and you can configure the agent using one of the following three methods:

- During installation, using the **Agent Configuration** panel.
- After installation, using the Manage Tivoli Monitoring Services program.
- After installation, using the `itmcmd config` command (Linux or UNIX only).

The use of any one of these methods requires you to provide the following information:

- Communication configuration information for the agent to contact the Tivoli Enterprise Monitoring Server.
- The TBSM Server installation directory value.

For Windows systems, the default value is `C:\Program Files\IBM\tivoli\impact\wlp\usr\servers\TBSM\logs`.

For non-Windows systems, the default value is `/opt/IBM/tivoli/impact/wlp/usr/servers/TBSM/logs`.

In addition, the `itmcmd config` command requires you to provide a product code for the agent. The product code for the TBSM agent is **r9**.

The following configuration fields are specific to the TBSM agent:

Custom Provider Client (CPC) port

The CPC port is used to communicate between the base agent and any custom-data provider code that the agent uses. You can configure the TBSM Agent CPC port number as part of the agent configuration. When the TBSM agent is installed and started for the first time, it creates a file that specifies the CPC port named `kr9_cps.properties`.

UNIX On UNIX systems, the file is created in `/tmp`.

On Windows systems, this file is located in the `CANDLE_HOME\TMAITM6` dir.

As an example on Windows the file is located in `C:\IBM\ITM\TMAITM6` and the file sets the value of the CPC port. The default value is: `CP_PORT=2092`

TBSM_tbsmomnibuseventreader.log file monitoring and failover

In previous versions of TBSM, only the path to the `TBSM_tbsmomnibuseventreader.log` file was requested during agent configuration. To allow the agent to be installed and configured correctly on a TBSM backup server, the name of the log file that also needs to be configured as its name is typically `TBSM_tbsmomnibus_B_eventreader.log`. Having just the path value will not work on a TBSM backup data server. The default values for the TBSM agent are now:

```
$IMPACT_HOME/logs/TBSM_tbsmomnibuseventreader.log
```

```
%IMPACT_HOME%\logs\tbsm\TBSM_tbsmomnibuseventreader.log
```

Discovery Library Toolkit installation directory

To monitor the msgGTM_XT.log file of the Discovery Library Toolkit, the path to where the msgGTM_XT.log is located must be configured properly. The default value is:

```
/opt/IBM/tivoli/XMLtoolkit/log
```

```
C:\Program Files\IBM\tivoli/XMLtoolkit\log
```

TBSM Process Control Agent process name

Indicates whether the TBSM Process Control Agent is installed. The default value of java_not_installed indicates that the Process Control Agent is not installed. In previous versions, the agent reported the Process Control Agent was down when it was not installed.

TBSM Discovery Library Toolkit process name

Indicates whether the TBSM Discovery Library Toolkit is installed. The default value of java_not_installed indicates that the Discovery Library Toolkit is not installed.

CPC Port

The Custom Provider Client port. The default is 2092.

Windows TBSM Data Server Service Name

The name of the TBSM data server service.

Windows TBSM Dashboard Server Service Name

The name of the TBSM Dashboard server service.

What to do next

On Windows, after installing the agent, if the Windows service "Monitoring Agent for Business Service Manager Agent - Primary" fails to start, the service might need to be configured to run as "Administrator" and not the "Local System Account".

Enabling historical reporting

If you intend to use the TBSM Historical Reporting feature, you must enable recording of data from the TBSM agent into the Tivoli Monitoring for the Tivoli Data Warehouse database.

About this task

Configure Tivoli Monitoring to record the data by using the Tivoli Enterprise Portal user interface.

Note: For data warehousing to collect historical data for TBSM, the Tivoli Monitoring for the Tivoli Data Warehouse database components must be installed, correctly configured, and working according to the *Tivoli Monitoring 6.0 (or later) Installation Guide*.

Procedure

1. Select **Edit -> History Configuration** from the main menu.
2. Choose **Business System Manager Common Agent** from the Monitored Application list.
3. Select any of the attribute groups listed and configure the attribute group for historical data collection (the tree shows the short name for the attribute group).
4. In the **Configuration** panel set the **Collection Interval**, **Collection Location** and **Warehouse Interval**.
5. On the **Distribution** tab add the **Available Systems** and/or **Available Managed Systems** groups to the **Start Collection** on list.
6. Optionally on the **Filter** tab create a filter to be applied to the historical data collection.

Enabling Agent Management Services for TBSM

This topic describes how to enable Agent Management Services for TBSM on your systems.

About this task

The TBSM agents can be managed by the IBM Tivoli Monitoring 6.3.0 Agent Management Services. These services are available in the Tivoli Monitoring OS Monitoring Agent for Windows and the Tivoli Monitoring 6.3.0 OS Monitoring Agent for Linux.

To enable Agent Management Services for TBSM on your systems with the Tivoli Monitoring 6.3.0 OS Monitoring Agent for Windows or Linux, copy the `kr9.xml` file from the TBSM install media to the CAP file subdirectory created in the OS Monitoring Agent's file tree. These services are designed to keep the TBSM agents available and to provide information about their status to the Tivoli Enterprise Portal.

For more information about Agent Management Services, see:

- Chapter 14 of the IBM Tivoli Monitoring *Administration Guide* at:
https://www.ibm.com/support/knowledgecenter/en/SSTFXA_6.3.0/com.ibm.itm.doc_6.3/itm63_admin.pdf
- The 6.3.0 *Windows OS User's Guide* at:
https://www.ibm.com/support/knowledgecenter/en/SSTFXA_6.3.0/com.ibm.itm.doc_6.3/winosagent63_user.pdf
- The 6.3.0 *UNIX OS User's Guide* at:
https://www.ibm.com/support/knowledgecenter/en/SSTFXA_6.3.0/com.ibm.itm.doc_6.3/unixosagent63_user.pdf

Setting up TBSM agents as Windows services

This topic discusses how to set up TBSM agents as Windows services.

About this task

The TBSM agent will only show status for Windows when JazzSM and OMNIBus are set up as services.

- Jazz needs to be configured as a Windows service and you need to configure the service name in the agent for it to show up. Follow these steps to configure Jazz as service on Windows:

https://www.ibm.com/support/knowledgecenter/en/SSGSPN_9.4.0/com.ibm.tivoli.itws.doc_9.4/distr/src_ad/awsadJazzSMWinservice.htm

Set the Jazz service name in the Agent configuration as **IBMWAS85Service - JazzSMProfile** and restart the agent. It will then show up in TEPS.

- The Data Server service name needs to be set as follows in the agent configuration:

```
NetcoolImpactTBSM_9080 for primary data server
NetcoolImpactTBSM_B_9080 for secondary data server
```

- For the ObjectServer, `NC00bjectServer` needs to be set up as a Windows service. Follow these steps:

https://www.ibm.com/support/knowledgecenter/en/SSSHTQ_8.1.0/com.ibm.netcool_OMNIBus.doc_8.1.0/omnibus/wip/common/task/omnweb_wininstallingcompservices.html

TBSM Agent installation reference

This topic describes information that is specific to the TBSM agent.

Purpose

The TBSM agent installation program helps you install the agent in your environment. See the *IBM Tivoli Monitoring Installation and Setup Guide* for additional installation requirements.

Prerequisites

The installation program automatically detects if you need any prerequisite software and makes the appropriate selections on the Install Prerequisites panel. Accept the defaults.

Select Features prompt

Select **Tivoli Enterprise Monitoring Agents**. Also select both the **Tivoli Enterprise Monitoring Agent Framework** and **Business Service Manager Agent**.

Tivoli Monitoring links

For information about installing, configuring, and administering monitoring agents, see the Tivoli Monitoring information center and these topics:

- [Deploying monitoring agents across your environment](#) in the Tivoli Monitoring *Installation and Setup Guide*
- [Working with monitoring agents](#) in the IBM Tivoli Monitoring *Administrator's Guide*.

For more information about Agent Management Services, see:

- In the IBM Tivoli Monitoring *Administration Guide* see:
https://www.ibm.com/support/knowledgecenter/SSTFXA_6.3.0/com.ibm.itm.doc_6.3/itm63_admin.pdf
- The 6.3.0 *Windows OS User's Guide* at:
https://www.ibm.com/support/knowledgecenter/SSTFXA_6.3.0/com.ibm.itm.doc_6.3/winosagent63_user.pdf
- The 6.3.0 *UNIX OS User's Guide* at:
https://www.ibm.com/support/knowledgecenter/SSTFXA_6.3.0/com.ibm.itm.doc_6.3/unixosagent63_user.pdf

Workspaces reference

This topic describes the navigator items and workspaces for the Business Service Management Agent.

Business Service Management Agent Navigator item

This navigator item opens the Business Service Management Agent workspace. This workspace contains the following view:

TBSMMAIN

The TBSM Agent Main Workspace, shows a tabular view of the historical daily status values for the services monitored by TBSM.

Availability Navigator item

This navigator item opens the Availability workspace. The Availability workspace displays the overall health of the application and include logical view for both the TBSM Database Server and TBSM Dashboard server.

This workspace contains the following views:

Availability

Displays the state of each component in the application. Each process is displayed using a descriptive name, the name of the running process, and the state of the process (UP, DOWN, or PROCESS_DATA_NOT_AVAILABLE). Each service is displayed using a descriptive name, the short name of the service, and the state of the service (UP, DOWN, or UNKNOWN). The state is UP if the service is running, DOWN if the service exists but is not running. UNKNOWN indicates that the service is not installed, so these elements are filtered from the view. When the state of the component is DOWN (for a process, or service) it is highlighted with a red background.

Processor

Displays the amount of CPU used by each process that is a component of the application. This displays the 2 main components of CPU usage, privileged time which is time spent in the kernel on behalf of the process and user mode time, which is the time spent running the process code.

Threads

Displays the number of threads used by each process that is a component of the application.

Memory

Displays the amount of memory being consumed by each process that is a component of the application. This total (virtual) size of the process and the size of the process in memory (working set) are displayed.

TBSM Event Broker Log navigator item

This navigator item opens the TBSM Event Broker Log workspace. It provides log file monitoring of the RAD_eventbroker.log file for TBSM.

This workspace contains the following view:

TBSMWSL

Displays the number of events read per second and the amount of memory used by the TBSM event broker based on data from the RAD_eventbroker.log file.

TBSM Service Indicators navigator item

This navigator item opens the TBSM Service Indicators workspace. This workspace contains the following view:

TBSMKPI

Shows the value(s) of key performance indicators that may be optionally configured for monitoring by TBSM. These KPIs are defined by writing the output value of a TBSM rule to an additional attribute in the service template.

TBSM Service Status navigator item

This navigator item opens the TBSM Service Status workspace. This workspace contains the following view:

TBSMWS

The main workspace shows the status of services monitored by TBSM, root cause events for the status change, and the status of the TBSM servers.

TBSM Status Change Event Navigator item

This navigator item opens the Status Change Event workspace. This workspace contains the following view:

TBSMWE

The Status Change Event workspace shows information about events that affected the status of TBSM

TBSM URL Monitor Navigator item

This navigator item opens the TBSM URL Monitor workspace. This provides basic URL monitoring for TBSM. This workspace contains the following view:

TBSMWSU

Displays response time metrics for URL monitored for the TBSM server.

Discovery Library Toolkit Log Navigator item

This navigator item opens the TBSM Discovery Library Toolkit Log workspace. This workspace contains the following view:

TBSMXML

Provides log file monitoring of the TBSM Discovery Library Toolkit msgGTM_XT.log.

Attribute groups and attributes for Business Service Management Agent

Attributes are the application properties being measured and reported by the Business Service Management Agent (TBSM Agent). Attribute groups are tables that group a specific set of attributes.

This monitoring agent contains the following attribute groups.

- Availability
- Performance Object Status
- TBSM Event Broker Log
- TBSM Service Indicators
- TBSM Service Status
- TBSM Status Change Event
- TBSM URL Monitor
- Discovery Library Toolkit Log

The remaining sections of this chapter contain descriptions of these attribute groups, which are listed alphabetically. The following information is provided for each attribute group:

Attributes

List of attributes that belong to the attribute group

Historical group

Whether the attribute group is a historical type that you can roll off to a data warehouse

Attribute descriptions

Description and type for each attribute in the attribute group

Availability attribute group

This table contains the availability data for all processes and services that make up this application. If the warehouse default setting is enabled, data for this attribute group is stored in Tivoli® Data Warehouse.

Attributes

Node - key attribute

The managed system name of the agent.

Type: String

Timestamp

The local time at the agent when the data was collected.

Type : String

Application Component - key attribute

The descriptive name of a part of the application.

Type: String

Name

The name of the process, service, or functionality test. This name matches the executable name of the process, the service short name or the name of the process used to test the application.

Type: String

Status

The status of the application component.

- For processes 'UP', 'DOWN', 'WARNING', or 'PROCESS_DATA_NOT_AVAILABLE': 'PROCESS_DATA_NOT_AVAILABLE' is displayed for a process when the matching process is running but the resource use information cannot be collected for that process.
- For services 'UP', 'DOWN', or 'UNKNOWN': 'UNKNOWN' is displayed when the service is not installed.
- For functionality tests: 'PASSED' or 'FAILED' is displayed.

Type: Integer with enumerated values. The strings are displayed in the Tivoli Enterprise™ Portal. The warehouse and queries return the values shown in parentheses. The following values are defined:

- DOWN (0)
- UP (1)
- WARNING (2)
- UNKNOWN (3)
- PASSED (4)
- FAILED (5)
- PROCESS_DATA_NOT_AVAILABLE (6)

Any other values will display the actual value returned by the agent in the Tivoli Enterprise Portal.

Full Name

The full name of the process including the path.

Type: String

Type attribute

The type of the application component. Components are processes, services, or functionality tests.

Type: Integer with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values shown in parentheses. The following values are defined:

- PROCESS (0)
- SERVICE (1)
- FUNCTIONALITY_TEST (2)

Any other values will display the actual value returned by the agent in the Tivoli Enterprise Portal.

Virtual Size

The virtual size (in MB) of the process.

Type: Integer (Gauge)

Page Faults per Sec

The rate of page faults for the process measured in faults per second. This attribute only contains valid data for processes.

Type : Integer (Gauge)

Working Set Size

The working set size of the process in MB. This attribute only contains valid data for processes.

Type: Integer (Gauge)

Thread Count

The number of threads currently allocated by this process. This attribute only contains valid data for processes.

Type : Integer (Gauge)

PID

The process ID associated with the process. This attribute only contains valid data for processes.

Type : Integer (Gauge)

Percent Privileged Time

The percentage of the available CPU time that is being used by this process for privileged operation.

Type : Integer (Gauge)

Percent User Mode Time

The percentage of the available CPU time that is being used by this process for user mode operation.

Type:Integer (Gauge)

Percent Processor Time

The percentage of the elapsed time that this process used the processor to execute instructions.

Type: Integer (Gauge)

Command Line

The program name and any arguments specified on the command line when the process was started. This has the value N/A if this is a Service, or Functionality test.

Type: String

Functionality Test Status

The return code of the functionality test. When the monitored application is running correctly, 'SUCCESS' is displayed. 'NOT_RUNNING' is displayed when it is not running correctly. 'N/A' is displayed when the row does not represent a functionality test.

Type : Integer with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values shown in parentheses. The following values are defined:

- SUCCESS (0)
- N/A (1)
- GENERAL_ERROR (2)
- WARNING (3)
- NOT_RUNNING (4)
- DEPENDENT_NOT_RUNNING (5)
- ALREADY_RUNNING (6)
- PREREQ_NOT_RUNNING (7)
- TIMED_OUT (8)
- DOESNT_EXIST (9)
- UNKNOWN (10)
- DEPENDENT_STILL_RUNNING (11)
- INSUFFICIENT_USER_AUTHORITY (12)

Any other values will display the actual value returned by the agent in the Tivoli Enterprise Portal.

Functionality Test Message

The text message that corresponds to the Functionality Test Status. This is only valid for functionality tests.

Type: String

Performance Object Status attribute group

This table reflects the status of other attribute groups so you can see the status of all of the performance objects that make up this application all at once.

Attributes

Each of these other performance attribute groups is represented by a row in this table (or other type of view). The status for an attribute group reflects the result of the last attempt to collect data for that attribute group, which allows you to see whether the agent is performing correctly. Unlike other attribute groups, the Performance Object Status attribute group does not reflect the state of the monitored application. This attribute group is most often used to determine why data is not available for one of the performance attribute groups. If the warehouse default setting is enabled, data for this attribute group is stored in Tivoli® Data Warehouse.

Node - key attribute

The managed system name of the agent.

Type: String

Timestamp

The local time at the agent when the data was collected.

Type : String

Query Name - This attribute is a key attribute.

The name of the attribute group.

Type :String

Object Name

The name of the performance object.

Type :String

Object Type

The type of the performance object.

Type: Integer with enumerated values. The strings are displayed in the Tivoli Enterprise™ Portal. The warehouse and queries return the values shown in parentheses. The following values are defined:

- WMI (0)
- PERFMON (1)
- WMI_ASSOCIATION_GROUP (2)
- JMX (3)
- SNMP (4)
- SHELL_COMMAND (5)
- JOINED_GROUPS (6)
- CIMOM (7)
- CUSTOM (8)
- ROLLUP_DATA (9)
- WMI_REMOTE_DATA (10)
- LOG_FILE (11)

Any other values will display the actual value returned by the agent in the Tivoli Enterprise Portal.

Object Status

The status of the performance object.

Type : Integer with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values shown in parentheses. The following values are defined:

- ACTIVE (0)

- INACTIVE (1)

Any other values will display the actual value returned by the agent in the Tivoli Enterprise Portal.

Error Code

The error code associated with the query

Integer with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values shown in parentheses. The following values are defined:

- NO_ERROR (0)
- GENERAL_ERROR (1)
- OBJECT_NOT_FOUND (2)
- COUNTER_NOT_FOUND (3)
- NAMESPACE_ERROR (4)
- OBJECT_CURRENTLY_UNAVAILABLE (5)
- COM_LIBRARY_INIT_FAILURE (6)
- SECURITY_INIT_FAILURE (7)
- PROXY_SECURITY_FAILURE (9)
- NO_INSTANCES_RETURNED (10)
- ASSOCIATOR_QUERY_FAILED (11)
- REFERENCE_QUERY_FAILED (12)
- NO_RESPONSE_RECEIVED (13)
- CANNOT_FIND_JOINED_QUERY (14)
- CANNOT_FIND_JOIN_ATTRIBUTE_IN_QUERY_1_RESULTS (15)
- CANNOT_FIND_JOIN_ATTRIBUTE_IN_QUERY_2_RESULTS (16)
- QUERY_1_NOT_A_SINGLETON (17)
- QUERY_2_NOT_A_SINGLETON (18)
- NO_INSTANCES_RETURNED_IN_QUERY_1 (19)
- NO_INSTANCES_RETURNED_IN_QUERY_2 (20)
- CANNOT_FIND_ROLLUP_QUERY (21)
- CANNOT_FIND_ROLLUP_ATTRIBUTE (22)
- FILE_OFFLINE (23)
- NO_HOSTNAME (24)
- MISSING_LIBRARY (25)
- ATTRIBUTE_COUNT_MISMATCH (26)
- ATTRIBUTE_NAME_MISMATCH (27)
- COMMON_DATA_PROVIDER_NOT_STARTED (28)
- CALLBACK_REGISTRATION_ERROR (29)
- MDL_LOAD_ERROR (30)
- AUTHENTICATION_FAILED (31)

Any other values will display the actual value returned by the agent in the Tivoli Enterprise Portal.

TBSM Event Broker Log attribute group

This attribute group monitors the RAD_eventbroker.log for information related to the reading of events from the TBSM ObjectServer. If the warehouse default setting is enabled, data for this attribute group is stored in Tivoli® Data Warehouse.

Attributes

Node - key attribute

The managed system name of the agent.

Type: String

Timestamp

The local time at the agent when the data was collected.

Type : String

eb read

Reserved for internal use.

Type: String

Events Read

The number of events read during a read cycle.

Type: Integer (Counter)

eb newread

Reserved for internal use.

Type : String

New Events Read

The number of new events read during a read cycle.

Type: Integer (Counter)

eb update

Reserved for internal use.

Type : String

Updates

The number of updates processed during a poll interval.

Type: Integer (Counter)

Queue Name

The name of the event queue.

Type: String

Queued Events

Number of events on the event queue.

Type: Integer (Counter)

eb readbuf

Reserved for internal use.

Type : String

Read Buffer

The size of the event broker read buffer.

Type: Integer (Counter)

eb polltime

Reserved for internal use.

Type : String

Poll Time

The event broker poll time.

Type: Integer (Counter)

eb epersec

Reserved for internal use.

Type : String

Events Read Per Second

The number of events read per second.

Type: Integer (Counter)

New Events Read Per Second

Reserved for internal use.

Type: String

eb nepersecval

Number of new events read per second.

Type: Integer (Counter)

eb memory

Reserved for internal use.

Type: String

Memory Usage

The current amount of memory used by the event broker in bytes.

Type: Integer (Counter)

TBSM Service Indicators attribute group

The TBSM service indicators attribute group contains information about key performance indicators (KPIs) on the services managed by TBSM. The KPIs are user-defined rules and additional attributes in TBSM. If the warehouse default setting is enabled, data for this attribute group is stored in Tivoli® Data Warehouse.

Attributes

Node - key attribute

The managed system name of the agent.

Type: String

Timestamp

The local time at the agent when the data was collected.

Type : String

ServiceName - This is a key attribute.

The service name in TBSM. This name is unique for each service.

Type: String

Primary Template Name

The name of the primary service template assigned to the service instance. The template includes the rules that determine the status thresholds for the service.

Type: String

StatusTime

The time when the status was recorded in TBSM in the MM/DD/YY HH:MM:SS format.

Type: Timestamp

Indicator Name - This is a key attribute.

The name of key performance indicator attribute as defined in a TBSM rule. To pass a KPI attribute from TBSM, select the **Store Data for this rule for TDW** option in the Metric Collection section of the editor window for the rule.

Alternately, you can name the rule with a suffix of `_KPI`.

Type: String

Indicator Value

The value of key performance indicator attribute defined in TBSM. This attribute contains the output value from a TBSM service template rule.

Type: Integer (Counter)

Previous Indicator Value

The previous value of key performance indicator attribute defined in TBSM.

Type: Integer (Counter)

TBSM Service Status attribute group

The Service Status attribute group reports the current values of selected attributes of TBSM service instances. These values are used in the main workspace under TBSM Service Status History. If the warehouse default setting is enabled, data for this attribute group is stored in Tivoli® Data Warehouse.

Attributes**Node - key attribute**

The managed system name of the agent.

Type: String

Timestamp

The local time at the agent when the data was collected.

Type : String

ServiceName - This is a key attribute.

The service name in TBSM. This name is unique for each service.

Type: String

DisplayName

The label of the service in the TBSM console. This can be different than the service name.

Type: String

ParentServiceName

Name of parent service. The child service defined by the Service Name attribute affects the status of this parent service.

Type: String

LongParentServiceName

Long name of parent service, used for extra long Parent Names. The child service defined by the Service Name attribute affects the status of this parent service.

Type: String

OverallAttribute

The overall status of the service. The values can be Clear (0), Marginal (3), Unknown (1), Maintenance (2), or Bad (5).

Type : Integer with enumerated values. The strings are displayed in the Tivoli Enterprise™ Portal. The warehouse and queries return the values shown in parentheses. The following values are defined:

- Clear (0)
- Unknown (1)
- Maintenance (2)
- Marginal (3)
- Major (4)
- Bad (5)

Any other values will display the actual value returned by the agent in the Tivoli Enterprise Portal.

BSM Identity

Reserved. BSM Identity value correlates external events with TBSM services. For example, if an event is from IBM® Tivoli Monitoring, the BSM_Identity attribute matches the value of the \$situation_origin attribute for the situation event. This field is always blank.

Type: String

BSM ClassName

Class identity reserved for future use.

Type: String

Primary Template Name

The name of the primary service template assigned to the service instance. The template includes the rules that determine the status thresholds for the service.

Type: String

StatusTime

The time when the status was recorded in TBSM in the MM/DD/YY HH:MM:SS format.

Type: Timestamp

CurrentDowntimeDuration

Duration of current downtime in seconds.

Type: Integer (Counter)

DowntimeInCurrentHour

Reserved. downtime within the current hour in seconds.

Type: Integer (Counter)

DowntimeInCurrentDay

Reserved. downtime within current day in seconds.

Type Integer (Counter)

DowntimeInCurrentMonth

Reserved. downtime within current month in seconds.

Type : Integer (Counter)

TBSM Service Status Change Event attribute group

The BSM service status change event attribute group contains information on the IBM® Netcool/Omnibus events that affect the status of TBSM services. If the warehouse default setting is enabled, data for this attribute group is stored in Tivoli® Data Warehouse.

Attributes**Node - key attribute**

The managed system name of the agent.

Type: String

Timestamp

The local time at the agent when the data was collected.

Type : String

ServiceName - This is a key attribute.

The service name in TBSM. This name is unique for each service.

Type: String

DisplayName

The label of the service in the TBSM console. This can be different than the service name.

Type: String

BSM Identity

Reserved. BSM Identity value correlates external events with TBSM services. For example, if an event is from IBM® Tivoli Monitoring, the BSM_Identity attribute matches the value of the \$situation_origin attribute for the situation event. This field is always blank.

Type: String

BSM ClassName

Class identity reserved for future use.

Type: String

ServerName

The ServerName field from the event.

Type: String

ServerSerial

The ServerSerial field from the event.

Type : Integer (Counter)

StatusTime

The time when the status was recorded in TBSM in the MM/DD/YY HH:MM:SS format.

Type: Timestamp

StatusChange

The time when the status-change event occurred.

Type : Timestamp

Identifier - This is a key attribute.

Identifier field in the event. Each event must have a unique value in the Identifier field.

Type: String

FirstOccurrence

First Occurrence field in the event. This value shows when the event first occurred.

Type: Timestamp

LastOccurrence

LastOccurrence field in the event. This value shows when the event last occurred.

Type: Timestamp

Severity

The Severity level of the event. The values can be Clear (0), Indeterminate (1), Warning (2), Minor (3), Major (4), or Critical (5).

Type: Integer with enumerated values. The strings are displayed in the Tivoli Enterprise™ Portal. The warehouse and queries return the values shown in parentheses. The following values are defined:

- Clear (0)
- Indeterminate (1)
- Warning (2)
- Minor (3)
- Major (4)
- Critical (5)

Any other values will display the actual value returned by the agent in the Tivoli Enterprise Portal.

Summary

The Summary field from the event.

Type: String

AlertGroup

The Alert Group field from the event.

Type: String

AlertKey

The Alert Key field from the event.

Type: String

TBSM URL Monitor attribute group

Response time metrics for the TBSM WEB application are maintained in the TBSM URL Monitor attribute group. If the warehouse default setting is enabled, data for this attribute group is stored in Tivoli® Data Warehouse.

Attributes**Node - key attribute**

The managed system name of the agent.

Type: String

Timestamp

The local time at the agent when the data was collected.

Type : String

URL

Reports the TCP/IP hostname of the TBSM server. The default is http://localhost:8080.

Type: String

HTTPResponseCode

The HTTP response code returned from the URL availability test.

Type: Integer (Counter)

HTTPResponseMessage - This is a key attribute.

The HTTP response message from the URL availability test.

Type: String

ResponseTime

Response time in milliseconds. Reports the number of seconds that was required for the transaction to complete.

Type: Integer (Counter)

AvgResponseTime

Average response time in milliseconds. Reports the average elapsed time between connecting with the Web server and downloading the Web page.

Type: Integer (Average over time, calculated by dividing the sum of the response time values by the number of responses)

MinResponseTime

Minimum response time in milliseconds. Reports the minimum response time for a single transaction instance during the data interval.

Type: Integer (The lowest response time among the number of responses received)

MaxResponseTime attribute

Maximum response time in milliseconds. Reports the maximum response time for a single transaction instance during the data interval.

Type: Integer (The highest response time among the number of responses received)

Discovery Library Toolkit Log attribute group

This attribute group monitors the TBSM Discovery Library Toolkit msgGTM_XT .log for information related to the processing of DLA books.

Attributes

xml_message_id - this is a key attribute

The TBSM Discovery Library Toolkit message id.

Type: String

xml_message_txt

The TBSM Discovery Library Toolkit message text.

Type: String

Disk capacity planning for historical data

Disk capacity planning for a monitoring agent is a prediction of the amount of disk space to be consumed for each attribute group whose historical data is being collected.

Disk capacity factors

Required disk storage is an important factor to consider when you are defining data collection rules and your strategy for historical data collection.

The table in this chapter provides the following information required to calculate disk space for this agent:

- *Table* is the table name as it is displayed in the warehouse database, if the attribute group is configured to be written to the warehouse.
- *Attribute group* is the name of the attribute group as it is displayed in the warehouse configuration panel.
- *Bytes per instance (agent)* is an estimate of the record length for each row or instance written to the agent disk for historical data collection. This estimate can be used for agent disk space planning purposes.
- *Database bytes per instance (warehouse)* is an estimate of the record length for detailed records written to the warehouse database, if the attribute group is configured to be written to the warehouse. Detailed records are those that have been uploaded from the agent for long-term historical data collection. This estimate can be used for warehouse disk space planning purposes.
- *Aggregate bytes per instance (warehouse)* is an estimate of the record length for aggregate records written to the warehouse database, if the attribute group is configured to be written to the warehouse. Aggregate records are created by the Summarization agent for attribute groups that have been configured for summarization. This estimate can be used for warehouse disk space planning purposes.

In addition to the information in the tables, you must know the number of instances of data that you plan to collect. An attribute group can have single or multiple instances of data depending on the application environment that is being monitored. For example, if your attribute group is monitoring each processor in your computer and you have a dual processor computer, the number of instances is 2.

The following table contains capacity planning information for the data logged by the Business Service Management Agent.

Table	Attribute group	Bytes per instance (agent)	Database bytes per instance (warehouse)	Aggregate bytes per instance (warehouse)
KR9AVAIL	KR9_AVAILABILITY	3272	3296	3606

Table 10. Capacity planning for historical data logged by the Business Service Management Agent (continued)

Table	Attribute group	Bytes per instance (agent)	Database bytes per instance (warehouse)	Aggregate bytes per instance (warehouse)
KR9POBJST	KR9_PERFORMANCE_OBJECT_STATUS	288	289	326
KR9RADLOG	KR9_TBSM_EVENT_BROKER_LOG	337	351	523
KR9KR9KPI	KR9_TBSM_SERVICE_INDICATORS	481	483	550
KR9KR9STAT	KR9_TBSM_SERVICE_STATUS	1264	1277	1374
KR9KR9SCHG	KR9_TBSM_STATUS_CHANGE_EVENT	2063	2078	2130
KR9KR9URLC	KR9_TBSM_URL_MONITOR	451	454	590
KR9TBSMXML	KR9_TBSM_DISCOVERY_LIBRARY_TOOLKIT_LOG	300	310	400

For more information about historical data collection, see the IBM Tivoli Monitoring *Administrator's Guide*.

Predefined situations

This monitoring agent contains predefined situations, which are organized by Navigator item.

Navigator item situations

This monitoring agent contains the following predefined situations, which are organized by Navigator item:

Business Service Management Agent

o Not applicable

Availability

- KR9_Process_Data_Unavailable
- KR9_TBSM_Critical

TBSM Event Broker Log

Not applicable

TBSM Service Indicators

Not applicable

TBSM Service Status

Not applicable

TBSM Status Change Event

Not applicable

TBSM URL Monitor

KR9_TBSM_Web_App_Critical

Discovery Library Toolkit Log

Not applicable

The remaining sections of this chapter contain descriptions of each of these situations. The situations are organized by Navigator item. The following information is provided about each situation:

Description

Information about the conditions that the situation tests

Formula

Syntax that contains one or more logical expressions describing the conditions for the situation to monitor Run at startup Whether the situation is automatically distributed to instances of the agent or is available for manual distribution.

Sampling interval

Number of seconds that elapses between one sample of data that the monitoring agent collects for the server and the next sample

Situation persistence

Whether the conditions specified in the situation evaluate to "true" for the defined number of occurrences in a row before the situation is raised. The default of 1 means no persistence checking takes place.

Severity

Severity of the event: Warning, Informational, or Critical Clearing conditions Controls when a true situation closes: after a period of time, when another situation is true, or whichever occurs first if both are selected.

Availability navigator item situations

The Availability Navigator item contains these predefined situations:

KR9_Process_Data_Unavailable situation**Description**

Unable to gather process data for this process.

Formula

```
*IF *VALUE KR9_AVAILABILITY.Type *EQ PROCESS *AND
*VALUE KR9_AVAILABILITY.Status
*EQ PROCESS_DATA_NOT_AVAILABLE
```

Distribution type

This situation is automatically distributed to instances of this agent.

Sampling interval

1 minute

Situation persistence

The number of times the conditions of the situation must occur for the situation to be true is 3.

Severity

Informational

Clearing conditions

The situation clears when the condition becomes false.

KR9_TBSM_Critical situation**Description**

Indicates TBSM may not be available.

Formula

```
**IF *VALUE KR9_AVAILABILITY.Status *EQ DOWN
```

Distribution type

This situation is automatically distributed to instances of this agent.

Sampling interval

1 minute

Situation persistence

The number of times the conditions of the situation must occur for the situation to be true is 1.

Severity

Critical

Clearing conditions

The situation clears when the condition becomes false.

URL Monitor navigator item situations

The URL Monitor navigator item contains these predefined situations:

KR9_TBSM_Web_App_Critical situation**Description**

Indicates that the TBSM Web application may not be responding.

Formula

```
***IF *VALUE KR9_TBSM_URL_MONITOR.HTTPResponseCode *NE 200
```

Distribution type

This situation is automatically distributed to instances of this agent.

Sampling interval

1 minute

Situation persistence

The number of times the conditions of the situation must occur for the situation to be true is 1.

Severity

Critical

Clearing conditions

The situation clears when the condition becomes false.

Predefined take action commands

The Business Service Management Agent does not provide predefined Take Action commands.

Predefined policies

The Business Service Manager Common Agent does not provide predefined policies.

Chapter 9. National Language Support

English is the default language for Tivoli Business Service Manager (TBSM). If you want TBSM to display a language other than English, follow the instructions in this section.

In addition to English, the following languages are supported in this release of TBSM:

- Spanish
- Brazilian Portuguese
- German
- French
- Italian
- Japanese
- Korean
- Simplified Chinese
- Traditional Chinese
- Czech
- Hungarian
- Polish,
- Russian

Turkish support: To use Turkish, install TBSM in the English language. After installation, change the language to Turkish.

Note: If you encounter errors while running the database configuration utility in Czechoslovakian, Hungarian, or Russian, the error text may contain unreadable characters. If you get this error, rerun the database configuration utility in English.

For more information on Tivoli Event Integration Facility probe language support, see this topic on the Netcool/OMNIBus information center under the Tivoli Event Integration Facility probe section: [Internationalization support](#).

Installing the language pack for TBSM Agent

This topic describes how to successfully install the language pack for TBSM Agent using a Windows or UNIX/Linux platform.

Before you begin

Before starting the National Language Support installation program, the TBSM agent must be installed. Also, the installation must be run under the same user ID as the TBSM agent installation. To install the language pack, first verify that you have already installed the IBM Tivoli Monitoring (English) product and a JDK/JRE, and then perform the following steps. You will receive an error message, and the installation will stop if the prerequisites are not met.

Procedure

1. Retrieve the installation package from installation image.

The package is at `<install_image>\<operating_system>\NLS\Agent`

2. Launch the installation program.

- a)  Execute `lpinstaller.bat`.

- b) **UNIX**
Execute `lpinstaller.sh -c ITM Home Directory` where *ITM Home Directory* is the directory where you installed the IBM Tivoli Monitoring product.
3. Select the language of the installer and click **OK**.
4. Click **Next** on the **Introduction** panel.
5. Click **Add/Update** and click **Next**.
6. Select the folder in which the National Language Support package (NLSPackage) files are located.
Note: `KR9_NLS.nlspkg` is the default stored location.
7. Select the language support for the agent of your choice and click **Next**.
Note: Hold the **Ctrl** key while selecting if you want multiple language supports.
8. Select the languages that you want to install and click **Next**.
9. Examine the **installation summary** page and click **Next** to begin installation.
10. Click **Finish** after installation completes to exit the installation program.
11. Restart Tivoli Enterprise Portal Desktop Client, Tivoli Enterprise Portal Server, and/or Eclipse Help Server if they are installed.
Note: If the agent is installed on Tivoli Enterprise Monitoring Server, you need to install the language pack on both Tivoli Enterprise Portal Server and Tivoli Enterprise Monitoring Server.

Uninstalling the language pack for TBSM Common Agent

Use the uninstall program to uninstall the language pack for IBM Tivoli Business Service Manager Common Agent.

About this task

To uninstall the language pack, follow the steps to invoke the installation package for the TBSM Common Agent provided in [“Installing the language pack for TBSM Agent”](#) on page 99, but on step 5, select **Remove** rather than **Add/Update**.

Procedure

1. Retrieve the installation package from DVD #2.
The package is at `<DVD root>\NLS\Agent`
2. Launch installation package.
 - a) **Windows**
Execute `lpinstaller.bat`.
 - b) **UNIX**
Execute `lpinstaller.sh -c ITM Home Directory` where *ITM Home Directory* is the directory where you installed the IBM Tivoli Monitoring product.
3. Select the language of the installer and click **OK**.
4. Click **Next** on the **Introduction** panel.
5. Click **Remove** and click **Next**.
6. Examine the **installation summary** page and click **Next** to begin uninstall.
7. Click **Finish** after uninstall completes to exit the installation program.

Chapter 10. Migrating to TBSM 6.2.0

This topic describes the components that will be migrated to TBSM 6.2.0 from TBSM 6.1.1.5, along with the procedure to follow to migrate the data.

Since the TBSM dashboard has moved to DASH from TIP in 6.2.0, most of the existing custom pages or custom portlets will not be migrated. You must create the pages and portlets again in DASH. Also, the users and user permissions created in TBSM 6.1.1.5 in TIP must to be created again manually on DASH.

Note: During the export or import process, if you enter the wrong DB2 or password, or face any errors due to environment issues, you can restart the migration process after rectifying environment issues. During the Import process, you may see exceptions or errors in the console due to duplicate records being imported. These can safely be ignored.

Before you begin

- All users must be logged off.
- Stop all event feeds and disable all data fetchers, if feasible. These actions are necessary only if auto-population rules are actively creating services. You should not stop the ObjectServer, because doing so might adversely affect the functioning of the TBSM servers.
- Stop the XML Toolkit Service to prevent Discovery Library Books from creating services.
- Stop any scheduled processes that may be using the `rad_radshell` utility to populate the TBSM database.

About this task

The following TBSM components will be migrated:

- composites category artifacts
- eventidentifiers category artifacts
- labeling category artifacts
- maintenance
- menuactions
- notifications
- scrbase
- sla
- taddmfilters
- tbsmattributes
- treetemplate
- scrconfig
- crossnaming
- Services
- Templates and Rules
- Data Fetchers
- Data Sources
- Impact data

The following TBSM components will not be migrated:

- Custom Pages/Portlets
- Users and Permissions

- SCR data
- OMNIbus Data
- viewdefinition
- customcanvas
- chart

Exporting data on Linux and Windows platforms

Note: On the Windows server, you must have the 7z .exe utility installed and the path should be defined in the PATH variable. The latest 7zip installer is included in the migration package provided if needed.

1. Create a directory on the server where the TBSM 6.1.1 primary Data Server is installed. Unzip the zip file provided in the TBSM 6.2 FP2 package.
 - On Windows unzip the following file: TBSM_Dataserver_Migration_Windows.zip
 - On Linux unzip the following file: TBSM_Dataserver_Migration_Linux.zip
2. Export the data from the source Data Server: Go to the directory where you unzipped the package and run the export command .
 - On Windows run the following command:
`TBSM_Dataserver_Export.bat %TBSM_HOME% db2username %NCHOME%`
 - On Linux run the following command:
`./TBSM_Dataserver_Export.sh $TBSM_HOME db2username $NCHOME`

Note:

- You will be prompted for the password for the DB2 user. This should be the user who installed the TBSM database.
- You will be prompted for the tbsmadmin password.
- The absolute path of TBSM installation (/opt/IBM/tivoli/tbsm) and NCHOME (/opt/IBM/tivoli) must be included in the command.

The script will create the following folders with the corresponding exported data and a .zip file (containing data from all these folders, and which will be created under the directory from where the script is run).

- Radshell_Export
- TBSM_Artifacts
- Impact_Export
- Final zip file: TBSM_Export_Data.zip

Note: After the export script is run, the TBSM Data Server will be down. To rerun the script, you need to restart the TBSM Data Server and then run the above script.

Importing data on Linux and Windows platforms

Note: On the Windows server, you must have the 7z .exe utility installed and the path must be defined in the PATH variable.

1. On the TBSM 6.2 Data Server, create a directory and in it, unzip TBSM_Dataserver_Migration_Linux.zip.
2. Copy the TBSM_Export_Data.zip file created by the export procedure to the same directory. Do not unzip the file.
3. To import the data, run the import script.
 - On Windows run the following script:
`TBSM_Dataserver_Import.bat %TBSM_HOME% db2username`
 - On Linux run the following script:

```
./TBSM_Dataserver_Import.sh $TBSM_HOME db2username
```

Note:

- Enter the DB2 password and `impactadmin` user password when prompted.
 - In the command line, enter the absolute path of TBSM installation directory (`/opt/IBM/tivoli/tbsm`) in place of `$TBSM_HOME`.
4. This script will unzip the exported data into the directory where it is copied and import the data to the TBSM Data Server.
 5. To import the menu actions, you need to use the `putArtifact` utility: Go to `$XMLToolkit\bin\` and run the `getArtifact` command for the selected menu actions.

Note: During the export or import process, if you enter the wrong DB2 or password, or face any error due to the environment issues, the script will exit giving errors on the console. You can restart the migration process after rectifying environment issues.

After the completion of the data import, custom data sources from the old system may need to be edited using the Impact GUI on target system if required.

Chapter 11. Configuring TBSM: post installation

This section contains information about configuring your TBSM system.

Installing the Historical Reports

To install the TBSM Historical Reports, you need to import them through IBM Cognos Administration to Cognos Analytics V11.x.

Before you begin

Important:

Tivoli Common Reporting (TCR) is now out of support. Consequently, the TBSM reports install script will not work. On Cognos Analytics Server V11.x and above, TBSM historical reports must be imported through Cognos Analytics Server V11.x Administration.

The following TCR features do not work on Cognos Analytics Server V11.x

- Single sign on from Dashboard Application Services Hub (DASH) to Cognos V11.x
- Command line interface commands
- BIRT reports

See [End of support of Tivoli Common Reporting \(TCR\) service which is part of Jazz for Service Management \(JazzSM\) on the IBM Support site.](#)

To use TBSM reports, you must install Cognos Analytics Server V11.x. For details, refer to the following links:

- [How to Install Cognos Analytics 11.1.x](#)
- [How to migrate Tivoli Common Reporting v3.1.* to Cognos v.11x](#)

About this task

You must Import TBSM reports through Cognos Administration to Cognos Analytics Server V11.x. The reports use the data in the Tivoli Data Warehouse collected by the Tivoli Business Service Management agent. You can use Cognos V11.x functionality to modify and enhance reports using the Cognos-based report packages.

Note: BIRT reports are now out of support. If you have BIRT reports developed, you will need to do some customization on top of the equivalent Cognos reports that ship with TBSM 6.2.0.

All TBSM 6.2.0 Cognos reports must be exported from the built-in Cognos V10 administrator in TCR, then imported to Cognos V11.x. This consists of the following steps:

1. [Exporting the Tivoli Common Reporting 3.1.x Content Store](#)
2. [Importing the Content Store into Cognos V11.x](#)
3. [Creating a Datasource](#)

Note: If you have deployed TBSM Cognos reports, you need to export them from the built-in Cognos V10 administrator in TCR, then import to Cognos V11.x. If you have never deployed any TBSM reports and try to deploy now, you can bypass the export steps and start from import steps.

Procedure

1. Exporting the Tivoli Common Reporting 3.1.x Content Store
 - a. In DASH, open **Common Reporting** and select **Launch > Administration**.
 - b. Run a **New Export** from **Configuration > Content Administration**.

c. Follow the process to create a **New Export**:

- 1) Select **IBM Tivoli Business Service Manager History Agent** under **Public Folders** and click **OK**.
- 2) Complete and click **Next** on the successive export settings panels, then click **Finish**.
- 3) Click **Run**.
- 4) Click **OK** to confirm the export settings and start the export.

The export of the Content Store is stored on the TCR System under the following folder:

```
/opt/IBM/JazzSM/reporting/cognos/deployment
```

2. Importing the Content Store into Cognos V11.x

a. Copy the exported Content Store to the new Cognos V11.x system and save it under the Cognos V11.x deployment folder:

```
[root@cognosv11 deployment]# cp /mnt/export/TBSM-History-Agent-Export.zip /opt/ibm/cognos/analytics/deployment
```

If you have never deployed any TBSM reports and try to deploy one now, copy the TBSM reports package from the TBSM 6.2.0 GA image and save it under the Cognos V11.x deployment folder:

```
[root@cognosv11 deployment]# cp /mnt/TBSM620image/data_linux/Reports/package/TBSM_History_Agent.zip /opt/ibm/cognos/analytics/deployment
```

b. Access your Cognos BI using the Web browser URL and open the **Manage > Administration** console.

c. From the Cognos Administration, run a **New Import** from **Configuration > Content Administration** and select the Content Store from your exported package.

d. Select `TBSM-History-Agent-Export.zip` which you exported. Otherwise select TBSM 6.2.0 out of box reports package `TBSM_History_Agent.zip`.

e. Select what you want to import by completing and clicking **Next** on the successive import selection panels, then click **Finish**.

f. Click **Run**.

g. Click **OK** to confirm the import settings and start the import..

Once imported, all reports appears under the **Team content** section.

3. Creating a Datasource

a. Click **Manage > Administration** to start the **Administration** console.

b. Click **Configuration > Data Source Connection > New Data Source** to add the database connection.

Compare the existing data sources in the old TCR Cognos and create the same in Cognos Analytics.

c. Add the following connection:

- 1) Set the **Data Source Name**, for example to TDW, select the data source **Type** IBM Db2, and click **Next**.
- 2) Set the **DB2 Database Name**, for example to WAREHOUSE, complete the **Signon** details, and click **Next**.

d. Enter the details for the DB2 server and click the **Test** button.

If successful, you will see a message in the results table with the **Status** Succeeded.

e. Click **Close** and then **Finish**.

Note: Copy the required images into Cognos V11.x, then icons in reports can display successfully. Refer to the *Troubleshooting Guide*: **TBSM reports icon** issue.

f. Configure LDAP for Authentication.

By default, authentication is anonymous. To enable authentication, use the steps in the following link:

[Configuring LTPA using an LDAP namespace](#)

Specifying the schema name

About this task

This task describes how to specify the schema name for the Tivoli Data Warehouse.

Procedure

1. The TBSM History Agent Cognos-based reports are configured for the schema name of ITMUSER for the Tivoli Data Warehouse tables.
2. If the schema name is not ITMUSER, the database administrator can create a database alias for the tables in the TBSM History Agent data model. DB2 and Oracle let you create an alias (synonym in Oracle). This example shows the DB2 commands that are used to create the alias that is required for the TBSM History Agent tables. In this example, TESTITMUSER is the user-defined schema for the tables.

```
create alias ITMUSER."KR9_TBSM_SERVICE_STATUS" for
TESTITMUSER."KR9_TBSM_SERVICE_STATUS";
create alias ITMUSER."KR9_TBSM_SERVICE_INDICATORS" for
TESTITMUSER."KR9_TBSM_SERVICE_INDICATORS";
create alias ITMUSER."KR9_TBSM_STATUS_CHANGE_EVENT" FOR
TESTITMUSER."KR9_TBSM_STATUS_CHANGE_EVENT";
```

3. If your database is MS SQL or the database administrator does not want to create alias names, you must update the data model with the schema name as follows:
 - a) Install and configure the Cognos Framework Manager, which is the data modeling tool. See the instructions for Cognos Framework Manager: https://www.ibm.com/support/knowledgecenter/en/SSEP7J_11.0.0/com.ibm.swg.ba.cognos.inst_cr_winug.doc/c_install_fm.html?view=kc.
 - b) Open the Framework Manager. Select **File > Open. Browse** to the extracted TBSM reports package. Browse to the "model" folder and select the TBSM_History_Agent.cpf file.
 - c) After the TBSM History Agent model in the Framework Manager opens, expand **Data Sources** under **TBSM History Agent** in the Project Viewer.
 - d) Select **TBSM_HIST_DATA** under **Data Sources**.
 - e) When you select **TBSM_HIST_DATA**, the Properties view is updated with information about the data source. By default, the Properties view is located at the bottom center of the screen. If you do not see it there, select **View > Properties**.
 - f) In the **Properties**, edit the **Schema** field, add your schema name.
 - g) Save the project.
 - h) In the Project Viewer, expand **Packages**.
 - i) Right-click **IBM Tivoli Business Service Manager History Agent**.
 - j) Select **Publish Packages**.
 - k) The Publish Wizard opens.
 - l) Keep the default selection and click **Next**.
 - m) Click **Next** on the next screen.
 - n) Clear the **Verify the package before publishing** check box.
 - o) Click **Publish**.
 - p) A window is displayed that alerts you that A package with that name already exists and asks Do you want to publish this package?

- q) Click **Yes**.
- r) After the package is published, you are informed that the package has been published.
- s) Go back to Cognos Analytics V11.x Administration and check if the **Modified** field of "IBM Tivoli Business Service Manager History Agent" in the Public Folders of IBM Cognos Connection shows the time of publishing.

Note: You can leave the schema name blank when you modify the Framework Manager. If it is empty, then Cognos assumes that the schema name is the same as the user ID specified for the Tivoli Data Warehouse data source connection. If you want to specify a data source connection user ID different from the schema name, you must explicitly specify the schema name in the Framework Manager data source property.

Configuring data source failover

The failover capability associated with TBSM data sources used in ESDAs and data fetchers and is separate from the failover support of the TBSM Data server. If a data source (or database) has a primary and a backup server, you can enable the use of the backup database.

Before you begin

Before you begin, the following conditions must be met:

- You must have completed all failover configurations for your data source.
- All the TBSM servers must be running.

Procedure

1. In the left navigation pane, click **BSM** and then click **Service Configuration**. The **Service Configuration** page is displayed.
2. Under **Service Navigation**, select the **Data** navigation view.
3. Select the data source for which you want to enable the use of the backup server.
4. On the **Edit** tab in the *Service Editor*, scroll down until the primary source information is displayed.
5. Select **Failover** or **Failback** or **Disabled**.
6. Enter information for the backup source.

Configuring IBM Tivoli Business Service Manager for failover

You can set up a TBSM environment that contains multiple instances of the TBSM Data Server. When one of the primary servers fails, the backup server takes over as the primary server. Such an environment provides protection against data loss and ensures that TBSM is running and available. To provide redundancy, load balancing, or both for the Dashboard server, you can configure load-balancing support for one or more Dashboard servers. This load-balancing support applies only to the user interface; it does not affect TBSM event or status processing.

About this task

When failover is configured for TBSM data servers, it does not mean DB2 must also be in failover.

Overview of the failover process

A failover environment contains a primary and backup Data server and optionally a primary and backup Netcool/OMNIBus ObjectServer. The ObjectServer and Data server can be located on the same or on different computers. If the primary host fails or the network connection between the primary and backup servers is lost, the backup server takes over the processes of the primary server.

TBSM Data server and ObjectServer failover provides support for TBSM event and status processing, ObjectServer event and status processing, or both in the event of a hardware or software failure that affects one of these capabilities. You can configure a single backup Data server, a single backup

ObjectServer, or both that takes over processing if the primary server fails. These backup servers do not perform any operational processing when the primary server is functional. Therefore, these backup servers do not perform any load-balancing function.

Within a few minutes of startup time, the backup server or servers loads its database and resynchronizes its status from events. If the primary server resumes function while the backup server is running as the primary server, the original primary server assumes the role of backup server.

You can configure the system so that the original server assumes the role of primary, and the backup server returns to its backup role. This behavior is called fail back.

If there is network connectivity loss between the primary and backup servers, the backup server assumes the role of primary server and the original primary server still functions as the primary server. When connectivity resumes, the backup server detects that the primary server is running again and it transitions back to the role of backup server. If the backup server is restarted, it resumes its backup role.

A data fetcher failure in the primary server does not trigger the failover process. If the primary data fetcher cannot connect to the database, the backup data fetcher probably cannot connect either.

Figure 1 on page 109 illustrates the architecture of a failover environment.

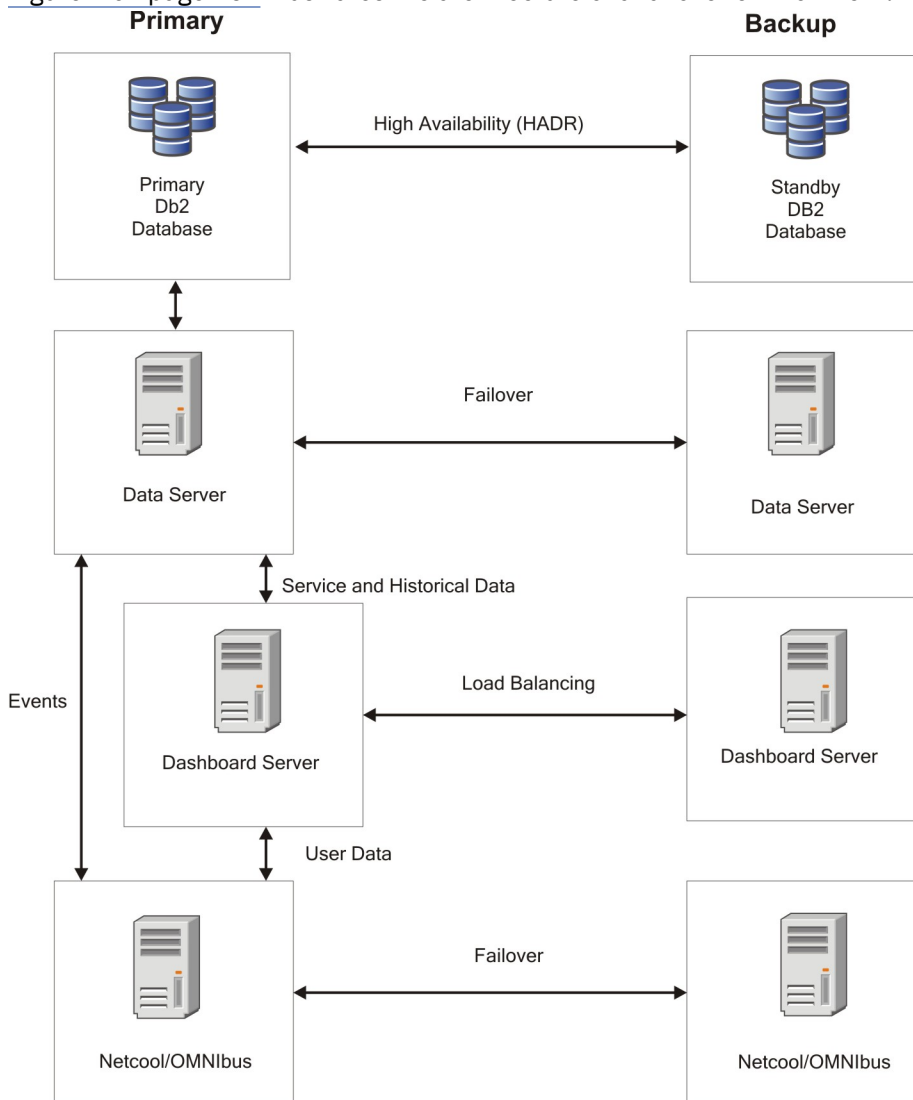


Figure 1. Failover architecture

By default, both the primary and backup TBSM Data and Dashboard servers use the primary instances of Tivoli Netcool/OMNIBus. The backup servers for the supporting applications also use the primary servers of the other applications by default.

Time Window Analyzer metric data store

As with all properties files, all the TBSM Metric Collection property files must be kept in sync between the primary and backup TBSM Data servers. The TBSM Metric Collection Component does not provide any automatic processes for syncing these files.

Requirements for setting up a failover environment

When you install a Data server as a backup server, you must indicate this during the install by selecting a check box.. These requirements are in addition to the requirements for using the TBSM database, Discovery Library Toolkit, EIF probe, and SLA events.

When the servers are installed, the following requirements must be met for a failover configuration to function correctly:

- The primary and backup servers must be running the same operating system and the same version and release of TBSM.
- The same non-root user ID and password must be used to install TBSM on the primary and backup servers.
- The primary and backup server must use the same authentication method, either LDAP or OMNIBus. The file-based authentication method is not supported for failover configurations.
- For Netcool/OMNIBus authentication, you need to configure all the Data and Dashboard servers in your failover configuration for both the Primary (`host1/port1`) and backup (`host2/port2`) ObjectServers. For instructions on how to configure the TBSM servers for external authentication, see the *TBSM Administration Guide* > Configuring TBSM > Manually configuring TBSM for external user repositories.
- If any default port values were changed during installation, use these same values for both installations.

Discovery Library Toolkit requirements

On both the primary and backup Data servers, both instances of the Discovery Library Toolkit must point to the same data source: the Discovery Library books, IBM Tivoli Application Dependency Discovery Manager, or both.

Tivoli Event Integration Facility probe requirements

If the EIF probe is installed on the primary ObjectServer, it must also be installed on the backup ObjectServer.

DB2 high availability configuration

This topic provides information on DB2 high availability.

When failover is configured for TBSM data servers, it does not mean DB2 must also be in failover.

The recommended way to configure failover for DB2 in a TBSM environment is to use High Availability and Disaster Recovery (HADR). This automatically replicates all changes from the active database to the standby database, such that when the standby database takes over it has the same data as was on the active database. A detailed description of the HADR configuration procedure can be found in the IBM redbook at:

<http://www.redbooks.ibm.com/redbooks/pdfs/sg247363.pdf>

Follow these guidelines when configuring HADR:

Primary and standby systems need the same operating system and level.

Ensure that the primary and standby systems have the same operating system and level.

Primary and standby user credentials

You must assign the same username and password of the instance owners on both the primary and secondary DB2 servers.

HADR setup chapter notes

The section on configuring HADR on the user interface is described in *HADR setup* chapter of the redbook.

- The configuration is started by running **db2cc**, right clicking on the TBSM database and selecting the option for **High Availability Disaster Recovery**.

You need to configure HADR separately on each database where you need failover. In a typical TBSM installation there are two databases: the TBSM database (which by default contains all TBSM configuration) and the TBSMHIST database (which by default contains metric history data).

For most of rest of the steps you should be able to follow the instructions outlined in the redbook. For the screen titled *Copy objects to the Standby System* (p.64), you can simply click **Next** and not configure anything.

- For the screen titled "Specify synchronization mode for peer state log writing", select "Near synchronous" as the synchronization mode.
- In the screen titled Specify TCP/IP communication parameters (p. 65) enter the fully qualified domain names rather than short host names or IP addresses. The screen capture in the redbook (p. 67) does not contain the Peer Window configuration at the bottom of this screen. You should set this value to be 60 (do not leave it 0, which is the default).

DB2 HADR does not provide automatic takeover of the standby when the primary database becomes unavailable. In order to enable automatic failover for DB2, TSA (Tivoli System Automation) must be configured. Read the guidelines for UNIX and Windows systems.

UNIX TSA configuration guidelines

This topic provides information on how to configure TSA (Tivoli System Automation) for DB2 on UNIX systems.

In order to enable automatic failover for DB2 on UNIX systems, you need to configure TSA (Tivoli System Automation) using these guidelines.

On UNIX platforms, TSA is installed with DB2 and it is configured by the IBM DB2 High Availability Instance Configuration Utility (db2haicu).

The db2haicu utility is in the `sql1lib/bin` directory.

Documentation on using the db2haicu utility can be found in the whitepaper at:

<http://www.ibm.com/developerworks/data/library/long/dm-0907hadrd2haicu/>

Note: To configure TSA for DB2 9.7, you need to install fixpack3 before running **db2haicu**.

Automatic failover configuration guidelines

Pages 11-19 in this document describe how to configure automatic failover with **db2haicu** (from section 4.1.3 until the end of section 4). Follow these guidelines when you set up TSA:

Quorum configuration

For the quorum configuration (Section 4.2 p. 14) the simplest approach is to specify a single ip address that is reachable from both primary and standby DB2 machines as the ip for the quorum device. This is also the approach described in the white paper. Once the db2haicu utility has finished, your high availability DB2 configuration should be complete.

Creating a Cluster Domain

In section 4.2 (Creating a Cluster Domain subsection), if a domain already exists then it should be deleted using `db2haicu -delete'` on both instances. Also in this section, step 3 - the fully qualified domain name must be entered here.

Network Setup

In section 4.2 (Network Setup subsection) - if primary and standby instances are in different subnets then this section should not be run.

Virtual IP Address Setup

In section 4.2 (Virtual IP Address Setup subsection) - As above, a VIP cannot be created if the primary and standby instances are in different subnets.

Automating HADR failover & Primary Instance Setup

In section 4.2 (Automating HADR failover & Primary Instance Setup subsections) - the listed output will be different (i.e a subset) if the Network Setup subsection has not been run.

Windows TSA configuration guidelines

This topic provides information on how to configure TSA (Tivoli System Automation) for DB2 on Windows systems.

In order to enable automatic failover for DB2 on Windows systems, you need to configure TSA (Tivoli System Automation) using these guidelines.

On Windows platforms, TSA does not come embedded in the DB2 installation, and you need to download and install IBM Tivoli System Automation for Multi Platform (SAMP). This can be obtained online at:

<http://www-01.ibm.com/software/tivoli/products/sys-auto-multi/>

On win2008 x64 platform, SAMP depends on SAU, which can be downloaded from the MicroSoft website at:

http://www.microsoft.com/downloads/en/results.aspx?freetext=subsystem+for+unix-based+applications&displaylang=en&stype=s_basic

Automating DB2 HADR failover on Windows document notes

To configure TSA for Windows (SAMP), review the online document *Automating DB2 HADR Failover on Windows using Tivoli System Automation for Multiplatforms*. This document can be downloaded from the url:

http://public.dhe.ibm.com/software/data/sw-library/db2/papers/hadr_tsa_win.pdf

This document describes the procedure for installing and configuring SAMP.

Script patches

In Appendix C, the document details copying some necessary scripts into the DB2 installation. Two of those scripts that come with the SAMP package described earlier contain defects (mkhadr and hadr_monitor.ksh), and need to be updated with fixed versions of the scripts. These fixed versions can be found in the TBSM install in the directory: SAMP directory.

INSTALL_HOME/tbsm/contrib/SAMP

Script notes

With regard to the scripts mentioned in Appendix C, the user must ensure they exist on both cluster nodes, otherwise the command: `./mkdb2` will fail.

Script permissions

The owner/group of directory sapolicies must be changed to Administrator/Administrators, and execution permission must be given to it. To change permissions, run the commands:

```
$ pwd
/usr/sbin/rsct/
$ chmod -R 777 sapolicies
$ chown -R Administrator:Administrators sapolicies
```

Configuring the ObjectServer communication information for failover

If you are using a primary and a backup Netcool/OMNIbus ObjectServer, you must configure communication information for both the primary and backup ObjectServer to participate in failover.

Before you begin

Check that your ObjectServer host has this subdirectory:

```
$OMNIHOME/var
```

If this directory is missing, create it.

Replicating OMNIbus authentication data

If you use the Netcool/OMNIbus ObjectServer for authentication and the ObjectServer is set up for failover, you need replicate the security information in the ObjectServer.

To replicate the security, you need to uncomment some properties in these bidirectional gateway files located the directory:

```
$NCHOME/omnibus/gates/objserv_bi/
```

objserv_bi.map

Find this note about 130 lines down in the file.

```
#####  
# NOTE: If replication of the user related system tables is required,  
# uncomment  
# the table mapping definitions below. The associated table replication  
# definitions will also need to be uncommented.  
#####
```

Uncomment all the properties and between this note and the next note in the file.

objserv_bi.objectservera.tblref.def

Find this note about 30 lines down in the file:

```
#####  
# NOTE: If replication of the user related system tables is required,  
# uncomment  
# the replication definitions below. The associated maps will also need  
# to be uncommented.  
#####
```

Uncomment all the properties and between this note and the next note in the file.

objserv_bi.objectservb.tblref.def

Find this note about 30 lines down in the file:

```
#####  
# NOTE: If replication of the user related system tables is required,  
# uncomment  
# the replication definitions below. The associated maps will also need  
# to be uncommented.  
#####
```

Uncomment all the properties and between this note and the next note in the file.

Setting the FAILOVERPOLICY parameter

If you want to configure IBM Tivoli Netcool/OMNIbus for failback with TBSM, you must ensure that the FAILOVERPOLICY parameter is configured correctly.

About this task

To configure this parameter:

Procedure

1. Using a text editor, open the `$TBSM_HOME/etc/TBSM_ObjectServer_DS.ds` file.
2. Locate the `ObjectServer_DS.ObjectServer.FAILOVERPOLICY` property and change it to `FAILBACK`. This change ensures that the Data server switches to the backup IBM Tivoli Netcool/OMNIBus server if the primary IBM Tivoli Netcool/OMNIBus server fails. If the Primary IBM Tivoli Netcool/OMNIBus server comes back up it takes over.

Configuring Data servers for failover

In TBSM 6.2, you must install the Impact Server in cluster mode (setting the primary Impact Server instance name as TBSM and the cluster name as TBSMCLUSTER and the secondary Impact server with TBSM_B as the instance name and the same cluster name as the primary).

While installing the TBSM Data Server as the backup, select the **Designated Backup Server** checkbox in the **Impact Server** details panel. There is no need to run any post install steps to complete the failover configuration.

Please note that, while installing the TBSM Data Server as Primary, the secondary Impact server should be down and while installing as Secondary, the primary Impact server should be up.

Manually configuring the ObjectServer for failover

If the backup ObjectServer is on a separate computer, you cannot use the `fo_config` script; you must perform a manual configuration.

About this task

This procedure uses the default values for the primary ObjectServer (NCOMS), the backup ObjectServer (NCOMS_BKUP) and the bidirectional gateway (NCO_GATE).



Attention: If you plan to use an existing Netcool OMNIBus ObjectServer instead of the one included in your TBSM installation, see [“IBM Tivoli Netcool OMNIBus Considerations”](#) on page 20 for information about using an existing ObjectServer.

To manually configure the ObjectServer for failover, perform the following steps:

Procedure

1. Configure the backup ObjectServer:
 - a) Stop the existing ObjectServer, if it is running.
 - b) Issue the following command to create the backup ObjectServer and to name the backup ObjectServer NCOMS_BKUP: `install_directory/netcool/omnibus/bin/nco_dbinit -server NCOMS_BKUP`
The `install_directory/netcool/omnibus/etc/NCOMS_BKUP.props` file is created.
 - c) Issue the following command to start the backup ObjectServer: `install_directory/netcool/omnibus/bin/nco_objserv -name NCOMS_BKUP`
 - d) Issue the following commands to update the alerts.status table schema:

```
UNIX cat install_directory/netcool/omnibus/etc/tbsm_db_update.sql  
install_directory/netcool/omnibus/bin/nco_sql -server NCOMS_BKUP -user  
username -password password
```

```
Windows type install_directory\netcool\omnibus\etc\tbsm_db_update.sql  
install_directory\netcool\omnibus\bin\isql.bat -server NCOMS_BKUP -user  
username -password password
```

Note: The `tbsm_db_update.sql` file is only available in this location if Omnibus was installed using the TBSM Installer. If the Omnibus was installed without using the TBSM installer, this file is available in the DVD media in `arch/TBSM/omnibus/schema_files`.

- e) Update the value of BackupObjectServer from FALSE to TRUE in the *install_directory/netcool/omnibus/etc/NCOMS_BKUP.props* file.
See *NCOMS_BKUP.props* file in the Reference section of the *TBSM Administration Guide* for an example of this file.
2. Configure the bidirectional gateway for the backup ObjectServer:
- a) Create a subdirectory NCO_GATE in the *install_directory/netcool/omnibus/gates* directory.
 - b) Copy all the files in the *install_directory/netcool/omnibus/gates/objserv_bi* directory to the *install_directory/netcool/omnibus/gates/NGO_GATE* directory.
 - c) Rename the *install_directory/netcool/omnibus/gates/NGO_GATE/objserv_bi.map* file to *install_directory/netcool/omnibus/gates/NGO_GATE/NGO_GATE.map*.
 - d) Rename the *install_directory/netcool/omnibus/gates/NGO_GATE/objserv_bi.props* file to *install_directory/netcool/omnibus/gates/NGO_GATE/NGO_GATE.props*.
 - e) Edit the *install_directory/netcool/omnibus/gates/NGO_GATE/NGO_GATE.props* file to create required entries.
See *NGO_GATE.props* file in the Reference section of the *TBSM Administration Guide* for an example of this file. On each line, adjust the values shown on the right side of the colon (:) to appropriate values for your installation.
Important: Use the UNIX file and variable syntax in this file, even on Windows systems.
 - f) Copy the *install_directory/netcool/omnibus/gates/NGO_GATE/NGO_GATE.props* file to *install_directory/netcool/omnibus/etc/NGO_GATE.props*.
 - g) Edit the *install_directory/netcool/omnibus/gates/NGO_GATE/NGO_GATE.map* file on the backup ObjectServer host.
See *NGO_GATE.map* file in the Reference section of the *TBSM Administration Guide* for an example of the entries to add for the default TBSM OMNIbus schema.
Important: You must update this map whenever the alerts.status table schema changes. If you do not, the row in the alerts.status table is not synchronized.

Manually adding TBSM_B to an existing Dashboard server

After installing the primary server (TBSM) and secondary server (TBSM_B), you can manually add TBSM_B to the existing Dashboard server (if the Dashboard was not installed in a failover setup).

To manually add TBSM_B to an existing Dashboard server, use the following steps:

1. Open the following file:

```
../JazzSM/profile/installedApps/JazzSMNode01Cell/isc.ear/sla.war/etc/RAD_sla.props
```

- a. Change the line:

```
impact.sla.backup.serverhost=__BACKUP_BACKENDHOST__
```

to:

```
impact.sla.backup.serverhost=<Your TBSM_B server fully qualified hostname>
```

2. Open the following file:

```
../JazzSM/profile/installedApps/JazzSMNode01Cell/isc.ear/sla.war/etc/nameserver.props
```

- a. Change the line:

```
impact.registry.1.host=
```

to:

```
impact.registry.1.host=<Your TBSM_B server fully qualified hostname>
```

b. Change the line:

```
impact.registry.1.port=
```

to:

```
impact.registry.1.port=<Your TBSM_B server backend server port>
```

For example: 9080

c. Change the line:

```
impact.nameserver.count=1
```

to:

```
impact.nameserver.count=2
```

d. Add the following lines:

```
impact.nameserver.1.host=<Your TBSM_B server fully qualified hostname>
impact.nameserver.1.port=<Your TBSM_B server backend server port>
impact.nameserver.1.location=/nameserver/services
```

3. Open the following file:

```
../JazzSM/profile/installedApps/JazzSMNode01Cell/isc.ear/sla.war/etc/
RAD_server.props
```

a. Change the line:

```
impact.rmiupdateserver.hostname.backup=
```

to:

```
impact.rmiupdateserver.hostname.backup=<Your TBSM_B server fully qualified
hostname>
```

b. Change the line:

```
impact.rmiupdateserver.port.backup=
```

to:

```
impact.rmiupdateserver.port.backup=<rmi port>
```

This will be the same as **impact.server.rmiport** on the backend and is usually the default 17542.

4. Restart the JazzSM Server, TBSM_B Data Server and TBSM_B Impact UI Server.

Replicating the alerts.service_deps table required for OMNIBus failover

You must complete this procedure to ensure that the `alerts.service_deps` table is also replicated. This is required to ensure that the Service Details portlet operates correctly.

About this task

To complete this procedure, you must edit the `NCO_GATE.map` file. The `NCO_GATE.map` file is typically located in the `installdirectory/netcool/omnibus/gates/NCO_GATE/` directory. For more information about this and other configuration files, see the *Administrator's Guide*.

Procedure

1. Using a text editor, open the `NCO_GATE.map` file. Edit the file to add a section:

```
CREATEMAPPING Service_deps Map
(
  'EventKey'='@EventKey',
  'KeyField'='@KeyField',
```

```
'Service'=@Service'  
);
```

2. Save and close the file.
3. You must ensure that the new mapping is referenced in the definition file for the gateway, for example:

```
$OMNIHOME/gates/NC0_GATE/objserv_bi.objectservera.tblrep.def  
$OMNIHOME/gates/NC0_GATE/objserv_bi.objectserverb.tblrep.def
```

Using a text editor open the file and add the reference:

```
REPLICATE ALL FROM TABLE 'alerts.service_deps'  
USING MAP 'Service_depsMap';
```

4. Save and close the file.

Starting the servers in a failover environment

To start TBSM in a failover configuration, you need to start a primary server before you start the corresponding backup server.

About this task

Start the following processes:

- Primary ObjectServer
- Backup ObjectServer
- Backup OMNIbus gateway
- Primary Data server
- Backup Data server
- One of more Dashboard servers

You must start the processes in this order; a backup server stops if it cannot contact the primary server when it is initially started. Likewise, the OMNIbus gateway might stop if it cannot connect to the primary ObjectServer when it is initially started.

Procedure

1. **UNIX**

To start the TBSM Data Server use the following command :

```
$IMPACT_HOME/bin/startImpactserver.sh  
$IMPACT_HOME/bin/startGUIserver.sh
```

2. **Windows**

To start individual TBSM services, issue one or more of the following commands:

- To start the primary ObjectServer: `net start NC0ObjectServer`
- To start the backup ObjectServer: `net start NC0ObjectServer`
- To start the OMNIbus gateway (on the backup computer if a backup OMNIbus is configured): `net start :NC0ObjectServerGatewayBi"`
- To start the primary Data server: `%IMPACT_HOME%\bin\startImpactserver.bat`
- To start the backup Data server: `%IMPACT_HOME%\bin\startImpactserver.sh`
- To start one or more Dashboard servers: `%JAZZ_HOME%\profile\bin\startServer.bat server1`

Verifying that the servers failover successfully

This section describes how to verify the entire TBSM failover environment, which includes the ObjectServer and the TBSM servers.

About this task

The verification process tests the following items:

- The TBSM failover and fail back
- The ObjectServer synchronization through the ObjectServer bi-directional gateway

If completed successfully, this procedure demonstrates that the TBSM server performs failover and fail back without losing data. It also demonstrates that the events sent to the backup ObjectServer are propagated to the primary ObjectServer through the synchronization process provided by the bi-directional gateway

In this example test event, the incoming status rule uses the Node field for the service name and the AlertKey and Severity field values to determine the service status.

Procedure

1. Open a Web browser and connect to the TBSM Dashboard server.
You can log in and see the service and template in the **Service Administration** view.
2. Send a test event to a service that is assigned to a template with event-based incoming status rules.
For an example of creating a service, service templates, and incoming status rules, see the *TBSM Scenarios Guide*.
3. Stop the primary Data server.
The backup Data server assumes the active role and start providing service. Depending on the number of service instances, this might take a few minutes.

After the failover process completes, the Dashboard server automatically connects to the backup Data server. The display might show a `reconnecting...` message; if this occurs, the display refreshes itself. The service is still displayed.
4. Send test events to the service. Right-click the icon for `auto_webserver2`, and then select **Show > Send Test Event**.
The message counter changes, based on the event that you sent.
5. Restart the primary Data server by:

Enter:

```
$IMPACT_HOME/bin/startImpactserver.sh
```

Start the service:

```
%IMPACT_HOME%\bin\startImpactserver.bat
```

After this server starts, it must be in a backup role.

6. Return to the Web browser session to ensure that the services are still displayed and that the system is responsive.
7. Stop the backup Data server by issuing the command `kill -9 PID of TBSM_B server process`.
The primary Data server assumes the active role and start providing services. This might take a few minutes.
8. Return to the Web browser session to ensure that the services are still displayed and that the system is responsive.
9. To test the ObjectServer, issue the **rad_sendevent** command to send a message to the primary ObjectServer:

- a) Issue the following command:

```
$TBSM_HOME/bin/rad_sendevent ObjectServerHost port userid password
```

- b) At the READY prompt, type the following field name and value pairs. Separate each item by pressing **Enter**:

```
Identifier
tbsmprimary-Id001
Node
node for test service
AlertKey
AlertKey for test service
Severity
5
```

Note: The field name **tbsmprimary** is used as an example only.

The message count for the service increases.

10. Examine the message content and verify that the `Identifier` field is `Id001`.
11. To test the backup ObjectServer, issue the **rad_sendevent** command to send a message to the backup ObjectServer.

- a) Issue the following command: .

```
username@backup_object_server_hostname
installhome/tbsm/bin/rad_sendevent ObjectServer port root"
```

- b) At the READY prompt, type the following field name and value pairs. Separate each item by pressing **Enter**:

```
Identifier
chin8-Id001
Node
node for test service
AlertKey
AlertKey for test service
Severity
5
```

The message count for the service is increased again, because the primary and backup ObjectServers are synchronized by the bi-directional gateway. It might take several minutes for the increase to occur.

12. Examine the message content and verify that the `Identifier` field value is `Id001`.

Load balancing

You can setup a load balancing cluster of portal nodes with identical configurations to evenly distribute user sessions.

Load balancing is ideal for *Dashboard Application Service Hub* installations with a large user population. When a node within a cluster fails, new user sessions are directed to other active nodes.

You can create a load balanced cluster from an existing stand-alone application server instance, but must export its data before you configure it for load balancing. The exported data is subsequently imported to one of the nodes in the cluster so that it is replicated across the other nodes in the cluster.

Work load is distributed by session, not by request. If a node in the cluster fails, users who are in session with that node must log back in to access the *Dashboard Application Service Hub*. Any unsaved work is not recovered.

Synchronized data

After load balancing is set up, changes in the console that are stored in global repositories are synchronized to all of the nodes in the cluster using a common database. The following actions cause

changes to the global repositories used by the console. Most of these changes are caused by actions in the **Settings** folder in the console navigation.

- Creating, restoring, editing, or deleting a page.
- Creating, restoring, editing, or deleting a view.
- Creating, editing, or deleting a preference profile or deploying preference profiles from the command line.
- Copying a portlet entity or deleting a portlet copy.
- Changing access to a portlet entity, page, external URL, or view.
- Creating, editing, or deleting a role.
- Changes to portlet preferences or defaults.
- Changes from the **Settings** applications, including assigning users and groups to roles.

Note: Global repositories should never be updated manually.

During normal operation within a cluster, updates that require synchronization are first committed to the database. At the same time, the node that submits the update for the global repositories notifies all other nodes in the cluster about the change. As the nodes are notified, they get the updates from the database and commit the change to the local configuration.

If data fails to be committed on any given node, a warning message is logged into the log file. The node is prevented from making its own updates to the database. Restarting the Dashboard Application Service Hub Server instance on the node rectifies most synchronization issues, if not, the node should be removed from the cluster for corrective action. See [“Monitoring a load balancing cluster” on page 135](#) for more information.

Note: If the database server restarts, all connections from it to the cluster are lost. It may take up to five minutes for connections to be restored, so that users can again perform update operations, for example, modifying or creating views or pages.

Manual synchronization and maintenance mode

Updates to deploy, redeploy, or remove console modules are not automatically synchronized within the cluster. These changes must be performed manually at each node. For deploy and redeploy operations, the console module package must be identical at each node.

When one of the deployment commands is started on the first node, the system enters *maintenance mode* and changes to the global repositories are locked. After you finish the deployment changes on each of the nodes, the system returns to an unlocked state. There is not any restriction to the order that modules are deployed, removed, or redeployed on each of the nodes.

While in maintenance mode, any attempts to make changes in the portal that affect the global repositories are prevented and an error message is returned. The only changes to global repositories that are allowed are changes to a user's personal portlet preferences. Any changes outside the control of the portal, for example, a form submission in a portlet to a remote application, are processed normally.

The following operations are also not synchronized within the cluster and must be performed manually at each node. These updates do not place the cluster in maintenance mode.

- Deploying, redeploying, and removing wires and transformations
- Customization changes to the console user interface (for example, custom images or style sheets) using `consoleProperties.xml`.

To reduce the chance that users could establish sessions with nodes that have different wire and transformation definitions or user interface customizations, schedule these changes to coincide with console module deployments.

Requirements

The following requirements must be met before load balancing can be enabled:

- If you are creating a cluster from a stand-alone instance of Dashboard Application Service Hub, you must export its data before you configure it for load balancing. Once you have configured the cluster, you can import the data to one of the nodes for it to be replicated across the other nodes.
- Lightweight Directory Access Protocol (LDAP) or OMNIbus ObjectServer must be installed and configured as the user repository for each node in the cluster. See [Configuring LDAP user registries](#) for instructions on how to enable LDAP for each node.
- A front-end network dispatcher (for example, IBM HTTP Server) must be setup to handle and distribute all incoming session requests. See [Setting up intermediary services](#) for more information about this task.
- DB2 Version 9.7 must be installed within the network to synchronize the global repositories for the console cluster.
- Each node in the cluster must be enabled to use the same LDAP using the same user and group configuration.
- All console nodes in load balancing cluster must be installed in the same cell name. After console installation on each node, use the **-cellName** parameter on the **manageprofiles** command.
- All console nodes in load balancing cluster must have synchronized clocks.
- The Websphere application server and Dashboard Application Service Hub Server versions must have the same release level, including any fix packs. Fixes and upgrades for the runtime must be applied manually at each node.
- Before joining nodes to a cluster, in each case make sure the node uses the same file-based repository user ID, which has been assigned the role of *iscadmins*.

Exporting data from a stand-alone server to prepare for load balancing

You can export data from an existing stand-alone application server instance to create a data file that can be imported to a load balanced cluster.

About this task

When you are creating a new load balanced cluster from a stand-alone instance, you must first export all data from the stand-alone instance and subsequently import the previously exported data once the cluster is set up.

Note: If you are joining the server to an existing cluster, the other nodes in the cluster should not contain custom data, that is, each node in the cluster should be clean installations. When you import data from the stand-alone server it is replicated across all other nodes.

Procedure

1. At the command line, change to the following directory:

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

2. Run the following command to export the stand-alone server's data:

- **Linux** **UNIX** `restcli.sh export -username tbsmadmin -password tbsmadmin_password -destination data_file`
- **Windows** `restcli.bat export -username tbsmadmin -password tbsmadmin_password -destination data_file`

Where:

tbsmadmin

Specifies the administrator user ID.

tbsmadmin_password

Specifies the password associated with the administrator user ID.

data_file

Specifies the path and file name for the exported data, for example, `c:/tmp/data.zip`.

3. Create a new load balanced cluster using the stand-alone server, or join it to an existing cluster.
4. Import the previously exported data to any node in the cluster.

- a) At the command line, if necessary, change to the following directory:

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

- b) On one of the nodes in the cluster, run the following command to import the stand-alone server's data:

```
restcli.sh import -username tbsmadmin -password tbsmadmin_password -source data_file
```

Where:

tbsmadmin

Specifies the administrator user ID.

tbsmadmin_password

Specifies the password associated with the administrator user ID.

data_file

Specifies the path and file name for the data to be imported, for example, `c:/tmp/data.zip`.

Results

Create a new load balanced cluster using the stand-alone application server, or join it to an existing cluster. Once the cluster is configured, you can import the data file to one of the nodes in the cluster.

What to do next**Setting up a load balancing cluster**

You can configure a Dashboard Application Service Hub Server instance to use a database as a file repository instead of a local directory.

Before you begin

If you are creating a cluster from an existing Dashboard Application Service Hub Server instance that contains custom data, ensure that you have exported its data before you begin to configure it for load balancing. Once it is configured, you can import the data to one of the nodes in the new cluster.

Dashboard Application Service Hub is installed on a machine using the cell name designated for all console nodes within the cluster. You have installed and setup a network dispatcher (for example, IBM HTTP Server), DB2, and an LDAP as explained in [“Requirements” on page 120](#).

Procedure

1. On the machine where DB2 is installed, create a DB2 database (see [Creating databases](#)).
2. Check that you have the JDBC driver for DB2 on the computer where Dashboard Application Service Hub is installed. The JDBC driver should be available at: `/opt/IBM/WebSphere/AppServer/universalDriver/lib`.
3. Configure load balancing, see https://www.ibm.com/support/knowledgecenter/SSSHTQ_8.1.0/com.ibm.netcool_OMNIBus.doc_8.1.0/webtop/wip/task/web_con_lb_configure.html.
4. In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:
 - **Windows** `stopServer.bat server1`
 - **Linux** **UNIX** `stopServer.sh server1`

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

5. Make sure your database is empty and the server is not started.

Problems may occur if you try to setup load balancing on a non-empty database or active server.

6. From a command prompt, change to the `/opt/IBM/WebSphere/AppServer/bin` directory and issue this command:

- **Windows** `..\ws_ant.bat -f install.ant configHA -Dusername=DB2_username -Dpassword=DB2_password`

- **Linux** **UNIX** `../ws_ant.sh -f install.ant configHA -Dusername=DB2_username -Dpassword=DB2_password`

7. In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:

- **Windows** `startServer.bat server1`

- **Linux** **UNIX** `startServer.sh server1`

Results

The load balancing cluster is created and the console node is joined to the cluster as the first node.

What to do next

Add (or join) additional nodes to the cluster.

Joining a node to a load balancing cluster

You can configure a Dashboard Application Service Hub Server to join an existing load balancing cluster.

Before you begin

1. If you are joining a stand-alone Dashboard Application Service Hub Server instance to a cluster, ensure that you first export all of its data. Once you have joined it to the cluster, you can then import the previously exported data. Other nodes in the cluster should not contain any custom data and should effectively be new installed instances.
2. Make sure you have successfully enabled load balancing following the steps in [“Setting up a load balancing cluster”](#) on page 122.
3. Dashboard Application Service Hub should be installed to the node using the same cell name that is designated for the cluster.
4. All console modules deployed to the cluster must be already deployed to the node that you intend to join.
5. You should deploy any wires or transformations used by the nodes in the cluster.
6. If the cluster is using any customization changes in `consoleProperties.xml` you must copy these changes and this file to the same location on the node that you intend to join.
7. The node must be configured to the same LDAP with the same user and group definitions as all other nodes in the cluster.

About this task

The following parameters are used on the `join` option when a node is added:

- **-Dusername** - specify the DB2 administrator's username
- **-Dpassword** - specify the DB2 administrator's password

Procedure

1. Check that you have the JDBC driver for DB2 on the computer where Dashboard Application Service Hub is installed. The JDBC driver should be available at: `/opt/IBM/WebSphere/AppServer/universalDriver/lib`.
2. Configure load balancing, see https://www.ibm.com/support/knowledgecenter/SSHTQ_8.1.0/com.ibm.netcool_OMNIBus.doc_8.1.0/webtop/wip/task/web_con_lb_configure.html.
3. In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:

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

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

4. Make sure the Dashboard Application Service Hub Server is not started.
5. At a command prompt, change to the `/opt/IBM/WebSphere/AppServer/bin` directory and issue this command

- **Windows** `..\ws_ant.bat -f install.ant configHA -Dusername=DB2_username -Dpassword=DB2_password`
- **Linux** **UNIX** `../ws_ant.sh -f install.ant configHA -Dusername=DB2_username -Dpassword=DB2_password`

6. In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:

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

Results

The console node is joined to the cluster.

What to do next

Add another node to the cluster, or if you have completed adding nodes, enable server to server trust for each node to every other node in the cluster.

Depending on the network dispatcher (for example, IBM HTTP Server) that you use, you might have further updates to get session requests routed to the new node. Refer to the documentation applicable to your network dispatcher for more information.

Enabling server-to-server trust

Use this procedure to enable load balanced nodes to connect to each other and send notifications.

About this task

These steps are required to enable load balancing between the participating nodes. Complete these steps on each node.

Procedure

1. In a text editor, open the `ssl.client.props` file from the `/opt/IBM/WebSphere/AppServer/profileTemplates/management/documents/properties/` directory.

2. Uncomment the section that starts with **com.ibm.ssl.alias=AnotherSSLSettings** so that it looks like this:

```
com.ibm.ssl.alias=AnotherSSLSettings
com.ibm.ssl.protocol=SSL_TLS
com.ibm.ssl.securityLevel=HIGH
com.ibm.ssl.trustManager=IbmX509
com.ibm.ssl.keyManager=IbmX509
com.ibm.ssl.contextProvider=IBMJSSE2
com.ibm.ssl.enableSignerExchangePrompt=true
#com.ibm.ssl.keyStoreClientAlias=default
#com.ibm.ssl.customTrustManagers=
#com.ibm.ssl.customKeyManager=
#com.ibm.ssl.dynamicSelectionInfo=
#com.ibm.ssl.enabledCipherSuites=
```

3. Uncomment the section that starts with **com.ibm.ssl.trustStoreName=AnotherTrustStore** so that it looks like this:

```
# TrustStore information
com.ibm.ssl.trustStoreName=AnotherTrustStore
com.ibm.ssl.trustStore=${user.root}/config/cells/JazzSMNode01Cell/nodes/
    JazzSMNode01/trust.p12
com.ibm.ssl.trustStorePassword={xor}CDo9Hgw=
com.ibm.ssl.trustStoreType=PKCS12
com.ibm.ssl.trustStoreProvider=IBMJCE
com.ibm.ssl.trustStoreFileBased=true
com.ibm.ssl.trustStoreReadOnly=false
```

4. Update the location of the trust store that the signer should be added to in the `com.ibm.ssl.trustStore` property of `AnotherTrustStore` by replacing the default value **com.ibm.ssl.trustStore=\${user.root}/etc/trust.p12** with the correct path for your trust store. Example:

```
com.ibm.ssl.trustStore=${user.root}/config/cells/JazzSMNode01Cell/nodes/
    JazzSMNode01/trust.p12
```

After the update, the section must look like this:

```
com.ibm.ssl.trustStoreName=AnotherTrustStore
com.ibm.ssl.trustStore=${user.root}/config/cells/TIPCell/nodes/TIPNode/trust.p12
com.ibm.ssl.trustStorePassword={xor}CDo9Hgw=
com.ibm.ssl.trustStoreType=PKCS12
com.ibm.ssl.trustStoreProvider=IBMJCE
com.ibm.ssl.trustStoreFileBased=true
```

5. Save your changes to `ssl.client.props`.
6. Stop and restart the Dashboard Application Service Hub Server:
 - a) In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:
 - **Windows** `stopServer.bat server1`
 - **Linux** | **UNIX** `stopServer.sh server1`

Note: On UNIX and Linux systems, you are prompted to provide an administrator username and password.
 - b) In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:
 - **Windows** `startServer.bat server1`
 - **Linux** | **UNIX** `startServer.sh server1`
7. Complete all of the steps so far on each node before you continue with the rest of the steps.
8. Run the following command on each node for each *myremotehost* (that is, for every node that you want to enable trust with) in the cluster:

```
Windows C:\Program Files\IBM\JazzSM\profile\bin\retrieveSigners.bat
NodeDefaultTrustStore AnotherTrustStore -host myremotehost -port
remote_SOAP_port
```

```
Linux UNIX /opt/IBM/JazzSM/profile/bin/retrieveSigners.sh
NodeDefaultTrustStore AnotherTrustStore -host myremotehost -port
remote_SOAP_port
```

where `myremotehost` is the name of the computer to enable trust with; `remote_SOAP_port` is the SOAP connector port number (16313 is the default). If you have installed with non-default ports, check `/opt/IBM/var/JazzSMPprofile_portDef.properties` for the value of `SOAP_CONNECTOR_ADDRESS` and use that.

9. Stop and restart the Dashboard Application Service Hub Server:

a) In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:

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

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

b) In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:

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

Example

In this example, the load balancing cluster is comprised of two Microsoft Windows nodes named `myserver1` and `myserver2`. The command entered on `myserver1`:

```
retrieveSigners.bat NodeDefaultTrustStore AnotherTrustStore -host myserver2
-port 16313
```

The command entered on `myserver2`:

```
retrieveSigners.bat NodeDefaultTrustStore AnotherTrustStore -host myserver1
-port 16313
```

Verifying a load balancing implementation

Use the information in this topic to verify that your Dashboard Application Service Hub load balancing setup is working correctly once you have added all nodes to the cluster and enabled server-to-server trust.

About this task

This task allows you to confirm the following functions are working correctly:

- The database used for your load balancing cluster is properly created and initialized.
- Every node in the cluster uses the database as its repository instead of its own local file system.
- Server-to-server trust is properly enabled between nodes in the cluster.

To verify your load balancing configuration:

Procedure

1. Ensure that each Dashboard Application Service Hub Server instance on every node in the cluster is running.
2. In a browser, log into one node, create a new View and save your changes.
3. Log into the remaining nodes and verify that the newly created view is available in each one.

Preparing the HTTP server for load balancing

Install the IBM HTTP Server and configure the Web server plug-in for passing requests to the Dashboard Application Service Hub Server that are part of the load balancing configuration.

Before you begin

The IBM HTTP Server uses a Web server plug-in to forward HTTP requests to the Dashboard Application Service Hub Server. You can configure the HTTP server and the Web server plug-in to act as the load balancing server, that is, pass requests (HTTP or HTTPS) to one of any number of nodes. The load balancing methods supported by the plug-in are *round robin* and *random*:

- With a round robin configuration, when a browser connects to the HTTP server, it is directed to one of the configured nodes. When another browser connects, it is directed to a different node.
- With the random setting, each browser is connected randomly to a node. Once a connection is established between a browser and a particular node, that connection remains until the user logs out or the browser is closed.

The HTTP server is necessary for directing traffic from browsers to the applications that run in the *Dashboard Application Service Hub* environment. The server is installed between the portal and the Dashboard Application Service Hub Server, and is outside the firewall.

The Web server plug-in uses the `plugin-cfg.xml` configuration file to determine whether a request is for the application server.

About this task

Complete this procedure to configure the Web server plug-in for load balancing for each node.

Procedure

1. If you do not already have the IBM HTTP Server installed, install it before proceeding. It should be installed where it can be accessed from the Internet or Intranet (or both). Select the link at the end of this topic for the installation procedure.
2. Install IBM HTTP Server ensuring that you include the IBM HTTP Server Plug-in for IBM WebSphere Application Server option.
For more information, see http://publib.boulder.ibm.com/infocenter/wasinfo/fep/topic/com.ibm.websphere.ihs.doc/info/ihs/ihs/tihs_installihs.html.
3. Create a new CMS-type key database.
For more information see http://publib.boulder.ibm.com/infocenter/wasinfo/fep/index.jsp?topic=com.ibm.websphere.ihs.doc/info/ihs/ihs/tihs_createkeydb.html.
4. Create a self-signed certificate to allow SSL connections between nodes.
For more information, see http://publib.boulder.ibm.com/infocenter/wasinfo/fep/index.jsp?topic=com.ibm.websphere.ihs.doc/info/ihs/ihs/tihs_certselfsigned.html.
5. To enable SSL communications for the IBM HTTP Server, in a text editor, open `HTTP_server_install_dir/conf/httpd.conf`. Locate the line `# End of example SSL configuration` and add the following lines, ensuring that the KeyFile line references the key database file created in step “3” on page 127 and save your changes.

```
LoadModule ibm_ssl_module modules/mod_ibm_ssl.so
<IfModule mod_ibm_ssl.c>
    Listen 443
```

```
<VirtualHost *:443>
  SSLEnable
</VirtualHost>
</IfModule>
SSLDisable
KeyFile "C:/Program Files/IBM/HTTPServer/bin/test.kdb"
```

For more information, refer to the first example at http://publib.boulder.ibm.com/infocenter/wasinfo/fep/index.jsp?topic=/com.ibm.websphere.ihs.doc/info/ihs/ihs/tihs_setupssl.html.

6. Restart the IBM HTTP Server.

For more information, see http://publib.boulder.ibm.com/infocenter/wasinfo/fep/topic/com.ibm.websphere.ihs.doc/info/ihs/ihs/tihs_startihs.html.

7. On the IBM HTTP Server computer, to verify that SSL is enabled ensure that you can access `https://localhost`.

8. Stop and restart the Dashboard Application Service Hub Server:

a) In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:

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

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

b) In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:

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

9. Start the HTTP server:

a) Change to the directory where it is installed.

b) Run this command: `bin/apachectl start`

Note you must restart the server after changing the `plugin-cfg.xml` file.

What to do next

Enter the URL for the HTTP Server in a browser `http://HTTP_server_host/HTTP_server_port` and it will be forwarded to one of the nodes.

Note: The default load balancing method is random, whereby each browser is connected randomly to a node.

Setting clone IDs for nodes

Assign a clone ID for all nodes in the cluster.

About this task

Complete this procedure to set clone IDs for all nodes in the cluster. You must carry out these steps on each node.

Procedure

1. In a text editor, open the `server.xml` file from the `./JazzSM/profile/installedApps/JazzSMNode01Cell/isc.ear/sla.war/otherfiles/` directory
2. In `server.xml`, locate the entry `<components xmi:type="applicationserver.webcontainer:WebContainer`.
3. Within the `components` element, add the following entry:

```
<properties xmi:id="WebContainer_1183077764084" name="HttpSessionCloneId"
value="12345" required="false"/>
```

Where:

value is the clone ID for the node, for example, value="12345". The clone ID must be unique to each node. An example of an updated components element is provided here:

```
<components xmi:type="applicationserver.webcontainer:WebContainer"
xmi:id="WebContainer_1183077764084" enableServletCaching="false"
disablePooling="false">
  <stateManagement xmi:id="StateManageable_1183077764087"
initialState="START"/>
  <services xmi:type="applicationserver.webcontainer:SessionManager"
xmi:id="SessionManager_1183077764084" enable="true" enableUrlRewriting="false"
enableCookies="true" enableSSLTracking="false"
enableProtocolSwitchRewriting="false"
sessionPersistenceMode="NONE" enableSecurityIntegration="false"
allowSerializedSessionAccess="false" maxWaitTime="5"
accessSessionOnTimeout="true">
    <defaultCookieSettings xmi:id="Cookie_1183077764084" domain=""
maximumAge="-1" secure="false"/>
    <sessionDatabasePersistence
xmi:id="SessionDatabasePersistence_1183077764084"
datasourceJNDIName="jdbc/Sessions" userId="db2admin" password="{xor}0z1tPjsyNjE="
db2RowSize="ROW_SIZE_4KB" tableSpaceName=""/>
    <tuningParams xmi:id="TuningParams_1183077764084"
usingMultiRowSchema="false" maxInMemorySessionCount="1000"
allowOverflow="true" scheduleInvalidation="false"
writeFrequency="TIME_BASED_WRITE" writeInterval="10"
writeContents="ONLY_UPDATED_ATTRIBUTES" invalidationTimeout="30">
      <invalidationSchedule xmi:id="InvalidationSchedule_1183077764084"
firstHour="14" secondHour="2"/>
    </tuningParams>
  </services>
  <properties xmi:id="WebContainer_1183077764084" name="HttpSessionCloneId"
value="12345" required="false"/>
</components>
```

4. Save the changes you made to server.xml.

Generating the plugin-cfg.xml file

Run GenPluginCfg.bat to generate the plugin-cfg.xml file and save it in /opt/IBM/JazzSM/profile/config/cells/JazzSMNode01Cell.

About this task

Complete this procedure to generate the plug-cfg.xml file. You must carry out these steps on each node.

Procedure

1. On a node, change to /opt/IBM/JazzSM/profile/bin and run the following command:

- **Windows** GenPluginCfg.bat
- **Linux** **UNIX** GenPluginCfg.sh

This command generates a file called plugin-cfg.xml and saves it to the /opt/IBM/JazzSM/profile/config/cells/JazzSMNode01Cell directory.

2. On the IBM HTTP Server, in the following directory, replace the existing plugin-cfg.xml with the version generated in step "1" on page 129:

```
HTTP_web_server_install_dir/plugins/config/webserver1
```

The following steps establish the new /ibm/* URI (Uniform Resource Identifier), which is where the plug-in will redirect requests:

- a) On the IBM HTTP Server, change to the directory where the Web server definition file is (such as `cd plugins/config/webserver1`).
- b) Open the `plugin-cfg.xml` file in a text editor, and in reference to the sample content extract provided below, edit the file to provide details of your IBM HTTP Server and all Dashboard Application Service Hub Server instances.

HTTP SERVER PATH is the path to where the HTTP server is installed.

HTTP SERVER PORT is the port for the HTTP server.

SERVER1 is the fully qualified name of the computer where the application server is installed and started.

SERVER2 is the fully qualified name of the computer where another application server is installed and started.

CLONE_ID is the is the unique clone ID assigned to a particular node (server) in the cluster.

- c) In the `ServerCluster` section, the values for the `keyring` and `stashfile` properties should be **HTTP SERVER PATH** `/plugins/etc/plugin-key.kdb` and **HTTP SERVER PATH** `/plugins/etc/plugin-key.sth` respectively.
- d) Continue to add `Server` entries for any other nodes, following the same pattern. Add a new entry under `PrimaryServers` for each additional server.
- e) Add `CloneID` and `LoadBalanceWeight` attributes for every `Server` entry.

Important: For more information on web server plug-in workload management policies and to help you determine the appropriate values for the elements `LoadBalance` and `LoadBalanceWeight`, refer to the following articles:

- <http://www.redbooks.ibm.com/abstracts/TIPS0235.html>
- <http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg21219567>



Attention: The HTTP and HTTPS port values for all nodes should be the same.

```
<Config ASDisableNagle="false" IISDisableNagle="false"
IgnoreDNSFailures="false" RefreshInterval="60"
ResponseChunkSize="64" AcceptAllContent="false"
IISPluginPriority="High" FIPSEnable="false"
AppServerPortPreference="HostHeader" VHostMatchingCompat="false"
ChunkedResponse="false">
  <Log LogLevel="Trace" Name="HTTP SERVER PATH/Plugins/logs/webserver1/
http_plugin.log"/>
  <Property Name="ESIEnable" Value="true" />
  <Property Name="ESIMaxCacheSize" Value="1024" />
  <Property Name="ESIInvalidationMonitor" Value="false" />
  <Property Name="ESIEnableToPassCookies" Value="false" />
  <Property Name="PluginInstallRoot" Value="HTTP SERVER PATH/Plugins" />
  <VirtualHostGroup Name="default_host">
    <VirtualHost Name="*:16310" />
    <VirtualHost Name="*:80" />
    <VirtualHost Name="*:16311" />
    <VirtualHost Name="*:5060" />
    <VirtualHost Name="*:5061" />
    <VirtualHost Name="*:443" />
  <VirtualHost Name="*:HTTP SERVER PORT" />
  </VirtualHostGroup>
  <ServerCluster CloneSeparatorChange="false" GetDWLMTable="false"
IgnoreAffinityRequests="true" LoadBalance="Round Robin"
Name="server1_Cluster" PostBufferSize="64" PostSizeLimit="-1"
RemoveSpecialHeaders="true" RetryInterval="60">
    <Server Name="TIPNode1_server1"
ConnectTimeout="0" CloneID="CLONE_ID" ExtendedHandshake="false"
ServerIOTimeout="0" LoadBalanceWeight="100" MaxConnections="-1"
WaitForContinue="false">
      <Transport Hostname="SERVER1" Port="16310"
Protocol="http"/>
      <Transport Hostname="SERVER1" Port="16311"
Protocol="https">
        <Property name="keyring" value="HTTP SERVER PATH\Plugins\config
\webserver1\plugin-key.kdb"/>
        <Property name="stashfile" value="HTTP SERVER PATH\Plugins\config
\webserver1\plugin-key.sth"/>
      </Transport>
    </Server>
  </ServerCluster>
</Config>
```



```

        </Server>
<Server Name="TIPNode1_server2"
ConnectTimeout="0" CloneID="CLONE_ID" ExtendedHandshake="false"
ServerIOTimeout="0" LoadBalanceWeight="100" MaxConnections="-1"
WaitForContinue="false">
    <Transport Hostname="SERVER2" Port="16310"
Protocol="http"/>
    <Transport Hostname="SERVER2" Port="16311"
Protocol="https">
        <Property name="keyring" value="HTTP_SERVER_PATH\Plugins\config
\webserver1\plugin-key.kdb"/>
        <Property name="stashfile" value="HTTP_SERVER_PATH\Plugins\config
\webserver1\plugin-key.sth"/>
    </Transport>
</Server>
</PrimaryServers>
<Server Name="TIPNode1_server1" />
    <Server Name="TIPNode1_server2" />
</PrimaryServers>
</ServerCluster>
<UriGroup Name="server1_Cluster_URIs">
    <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionId"
Name="/ivt/*" />
    <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionId"
Name="/IBM_WS_SYS_RESPONSESERVLET/*" />
    <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionId"
Name="/IBM_WS_SYS_RESPONSESERVLET/*.jsp" />
    <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionId"
Name="/IBM_WS_SYS_RESPONSESERVLET/*.jsw" />
    <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionId"
Name="/IBM_WS_SYS_RESPONSESERVLET/*.jsw" />
    <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionId"
Name="/IBM_WS_SYS_RESPONSESERVLET/j_security_check" />
    <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionId"
Name="/IBM_WS_SYS_RESPONSESERVLET/ibm_security_logout" />
    <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/ibm/console/*" />
    <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/ibm/help/*" />
    <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/ibm/action/*" />
    <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/ISCWire/*" />
    <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/isc/*" />
    <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/ISCHA/*" />
    <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/tip_ISCAdminPortlet/*" />
    <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/ISCAdminPortlets/*" />
    <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/mum/*" />
    <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/ibm/TIPChangePasswd/*" />
    <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/ibm/TIPEXportImport/*" />
    <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/ibm/tivoli/*" />
    <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/proxy/*" />
    <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/TIPWebWidget/*" />
    <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/ibm/dbfile/*" />
    <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/ibm/TIPChartPortlet/*" />
    <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/TIPUtilPortlets/*" />
    <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/WIMPortlet/*" />
    <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/SysMgmtCommonTaskGroups/*" />
</UriGroup>
<Route ServerCluster="server1_Cluster" UriGroup="server1_Cluster_URIs"
VirtualHostGroup="default_host" />
<RequestMetrics armEnabled="false" newBehavior="false" rmEnabled="false"
traceLevel="HOPS">
    <filters enable="false" type="URI">
        <filterValues enable="false" value="/snoop" />
        <filterValues enable="false" value="/hitcount" />
    </filters>

```

```

    <filters enable="false" type="SOURCE_IP">
      <filterValues enable="false" value="255.255.255.255" />
      <filterValues enable="false" value="254.254.254.254" />
    </filters>
    <filters enable="false" type="JMS">
      <filterValues enable="false" value="destination=aaa" />
    </filters>
    <filters enable="false" type="WEB_SERVICES">
      <filterValues enable="false" value="wsdlPort=aaa:op=bbb:nameSpace=ccc" />
    </filters>
  </RequestMetrics>
</Config>

```

Configuring SSL from each node to the IBM HTTP Server

For load balancing implementations, you must configure SSL between the IBM HTTP Server plug-in and each node in the cluster.

Before you begin

This task assumes that you have already installed and configured the IBM HTTP Server for load balancing.

About this task

For each node in the cluster, follow these instructions to configure the node to communicate over a secure (SSL) channel with the IBM HTTP Server.

Procedure

- Log in to the *Dashboard Application Service Hub*.
- In the navigation pane, click **Settings** > **Websphere Administrative Console** and click **Launch Websphere administrative console**.
- Follow these steps to extract signer certificate from the trust store:
 - In the WebSphere Application Server administrative console navigation pane, click **Security** > **SSL certificate and key management**.
 - In the Related Items area, click the **Key stores and certificates** link and in the table click the **NodeDefaultTrustStore** link.
 - In the Additional Properties area, click the **Signer certificates** link and in the table that is displayed, select the **root** entry check box.
 - Click **Extract** and in the page that is displayed, in the **File name** field, enter a certificate file name (*certificate.arm*), for example, *c:\tivpc064ha1.arm*.
 - From the **Data Type** list select the **Base64-encoded ASCII data** option and click **OK**.
 - Locate the extracted signer certificate and copy it to the computer running the IBM HTTP Server.

Note: These steps are particular to Dashboard Application Service Hub, for general WebSphere Application Server details and further information, see: http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.base.doc/info/aes/ae/tsec_sslextractsigncert.html
- On the computer running the IBM HTTP Server, follow these steps to import the extracted signer certificate into the key database:
 - Start the key management utility (iKeyman), if it is not already running, from *HTTP_SERVER_PATH/bin*:
 - Linux** | **UNIX** At the command line, enter `./ikeyman.sh`
 - Windows** At the command line, enter `ikeyman.exe`
 - Open the CMS key database file that is specified in *plugin-cfg.xml*, for example, *HTTP_SERVER_PATH/plugin-etc/plugin-key.kdb*.
 - Provide the password (default is WebAS) for the key database and click **OK**.

- d) From the **Key database content**, select **Signer Certificates**.
 - e) Click **Add** and select the signer certificate that you copied from the node to the computer running the IBM HTTP Server and click **OK**.
 - f) Select the **Stash password to a file** check box and click **OK** to save the key database file.

Note: For more information on certificates in WebSphere Application Server, see: http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.ihs.doc/info/ihs/ihs/tihs_ikeyscca.html
5. Repeat these steps for each node in the cluster.
 6. For the changes to take effect, stop and restart all nodes in the cluster and also restart the computer running the IBM HTTP Server.
 - a) In the /opt/IBM/JazzSM/profile/bin directory, depending on your operating system, enter one of the following commands:
 - **Windows** stopServer.bat server1
 - **Linux** | **UNIX** stopServer.sh server1

Note: On UNIX and Linux systems, you are prompted to provide an administrator username and password.
 - b) In the /opt/IBM/JazzSM/profile/bin directory, depending on your operating system, enter one of the following commands:
 - **Windows** startServer.bat server1
 - **Linux** | **UNIX** startServer.sh server1
 - c) Restart the IBM HTTP Server.

For more information, see http://publib.boulder.ibm.com/infocenter/wasinfo/fep/topic/com.ibm.websphere.ihs.doc/info/ihs/ihs/tihs_startihs.html.

What to do next

You should now be able to access the load balanced cluster through `https://http_server_hostname/ibm/console` (assuming that the default context root (/ibm/console) was defined in at the time of installation).

Importing stand-alone instance data to a cluster

If you created a cluster from a stand-alone application server instance, you can then import the data that you exported prior to configuring the stand-alone instance as a cluster node.

About this task

Import the previously exported data file to any node in the cluster.

Important: The instructions in this topic apply only to importing data that was exported when preparing to create a load balanced cluster from a stand-alone application server instance, as described in [“Exporting data from a stand-alone server to prepare for load balancing” on page 121](#).

Procedure

1. At the command line, change to the following directory:


```
/opt/IBM/JazzSM/profile/bin
```
2. On one of the nodes in the cluster (most likely the node that was previously set up as a stand-alone server instance), run the following command to import the data file:
 - **Linux** | **UNIX** restcli.sh import -username *tbsadmin* -password *tbsadmin_password* -source *data_file*

- **Windows** `restcli.bat import -username tbsmadmin -password tbsmadmin_password -source data_file`

Where:

tbsmadmin

Specifies the administrator user ID.

tbsmadmin_password

Specifies the password associated with the administrator user ID.

data_file

Specifies the path and file name to the data file that is to be imported, for example, `c:/tmp/data.zip`.

Results

The data from the initial application server is imported to the node and replicated across the other cluster nodes.

Removing a node

Follow these steps to remove a node from the load balancing cluster.

About this task

The following parameters are used on the `disjoin` option when a node is removed.

- **-Dusername** - specify the DB2 administrator's username
- **-Dpassword** - specify the DB2 administrator's password

Procedure

1. From a command prompt, change to the `/opt/IBM/WebSphere/AppServer/bin` directory and issue this command:

- **Windows** `..\ws_ant.bat -f uninstall.ant disjoin -Dusername=DB2_username -Dpassword=DB2password`
- **Linux** **UNIX** `../ws_ant.sh -f uninstall.ant disjoin -Dusername=DB2_username -Dpassword=DB2password`

2. Update the network dispatcher (for example, IBM HTTP Server) to remove the node from the configuration.

Removing a remote node

About this task

This command should be used only in the rare occasions where physical access to the node is not available or a serious hardware or software failure has occurred. If the node is remotely disjoined but continues to function, some problems with synchronization might arise that can lead to problems with data consistency and synchronization.

Procedure

1. From a command prompt, change to the `/opt/IBM/WebSphere/AppServer/bin` directory and issue this command:

- **Windows** `..\ws_ant.bat -f uninstall.ant remote-disjoin -DremoteHost=remote_host -DremotePort=9044 -Dusername=DB2_username -Dpassword=DB2_password`

- **Linux** **UNIX** `./ws_ant.sh -f uninstall.ant remote-disjoin -DremoteHost=remote_host -DremotePort=9044 -Dusername=DB2_username -Dpassword=DB2_password`
2. Update the network dispatcher (for example, IBM HTTP Server) to remove the node from the configuration.

Removing a load balancing cluster

Follow these steps to remove the last node from a cluster and thereby the cluster itself.

Before you begin

Make sure you have removed all other nodes from the cluster. This command should be issued from the last active node remaining in the cluster.

About this task

The following parameters are used on the uninstall option when the node is removed.

- **-Dusername** - specify the DB2 administrator's username
- **-Dpassword** - specify the DB2 administrator's password

Procedure

From a command prompt, change to the `/opt/IBM/WebSphere/AppServer/bin` directory and issue this command:

- **Windows** `..\ws_ant.bat -f uninstall.ant uninstall -Dusername=DB2_username -Dpassword=DB2_password`
- **Linux** **UNIX** `./ws_ant.sh -f uninstall.ant uninstall -Dusername=DB2_username -Dpassword=DB2_password`

Monitoring a load balancing cluster

If synchronized data fails to be committed to a node in the cluster, that node should be removed from the cluster for corrective action. Use the diagnosis tool to identify any unsynchronized nodes in the load balancing cluster.

To determine if changes to global data are not committed to any of the nodes, use the **HATool** command script to check the synchronization of modules and repositories on the nodes in a cluster. For the HATool, you must provide the DB2 administrator's credentials.

Query synchronization of modules

Use this command to determine if all nodes have identical sets of modules deployed.

```
HATool.bat/sh modules username password -byNodes -showAll
```

The following parameters are optional.

- **-byNodes**

Specifies that the results of the command are ordered by the node in the cluster. This parameter is optional. The default is to list the results by module.

- **-showAll**

Specifies that all modules and nodes in the cluster should be returned. This parameter is optional. The default is to return only modules for unsynchronized nodes.

Query the synchronization of global repositories

Use this command to determine if all repositories are synchronized on all nodes.

```
HATool.bat/sh repositories username password -byNodes -showAll
```

The following parameters are optional.

- **-byNodes**

Specifies that the results of the command are ordered by the node in the cluster. This parameter is optional. The default is to list the results by repository.

- **-showAll**

Specifies that all repositories and nodes in the cluster should be returned. This parameter is optional. The default is to return only repositories for unsynchronized nodes.

Release the global lock

Use this command to manually release the global lock placed on all of the console nodes when the cluster is in maintenance mode. This command is used when a node cannot commit a change during synchronization and has to be taken offline.

```
HATool.bat/sh release-lock username password
```

Configuring secure connections

These topics describe how you can configure secure key connections (via SSL) for IBM Tivoli Business Service Manager (TBSM) servers.

Before you begin

If you plan to configure TBSM failover, the failover configuration must be completed and verified before attempting to configure secure connections. Otherwise, Data server startup problems may occur.

If you plan to secure TBSM and the DASH Console component connections to Netcool/OMNIbus, OMNIbus must be configured for secure communications first. The OMNIbus certificate(s) should be added to the TBSM Dashboard server and Data server trust stores before securing the TBSM connections to OMNIbus as described below.

When securing OMNIbus, it is highly recommended that a non-SSL port on the OMNIbus server be left available for TBSM components to connect to until all certificates have been added to the TBSM server trust stores and all configurations have been completed as described in the following sections.

To set up secure connections:

1. Perform all necessary certificate exchanges.
 - Use the console for Dashboard servers
 - Use `wsadmin` commands for Data servers
2. Run the `secure_server_config` script on each TBSM server installed.
3. If you are securing connections between the Data server and Netcool/OMNIbus, take additional configuration steps.

Secure connections overview

This topic is an overview of secure connection configuration for TBSM.

You can secure key connections (via SSL) between TBSM servers and between the TBSM and DASH Console components and Netcool/OMNIbus. By default, the connections are not secured. These connections include:

- Connections between TBSM servers where the connections are used to exchange data, for configuration, for status updates, and for server synchronization.
- Connections between TBSM and DASH Console components and Netcool/OMNIbus where the connections are used for event processing, user authentication (installation option), and event management (Netcool/WebTop).

Process overview

To set up secure connections:

1. Configure Netcool/OMNIbus for secure communications if secure channels between TBSM and TIP components and Netcool/OMNIbus are required. For more information about the Netcool/OMNIbus ObjectServer and SSL ports see: [Netcool/OMNIbus documentation](#) .
2. Perform all needed certificate exchanges between Dash and the TBSM data server.

These include:

- “Retrieving signer certificate data for dashboard servers” on page 139
 - “Importing trusted JazzSM certificate into the TBSM Dataserver, GUI and server trust stores” on page 139
3. Run the `secure_server_config` script on each TBSM server installed
 4. If you are securing connections between the data server and Netcool/OMNIbus, you take additional configuration steps.
 5. Restart all servers.

Time Window Analyzer marker clients and secure connections

If the data server is configured for secure connections, there are implications to how Time Window Analyzer marker clients connect to it. For more information about HTTPS connections and Time Window Analyzer marker clients, see the *Netcool Impact Marker Provider Library* and the *Marker Command Line Client Utility* topics in the *Tivoli Marker Repository Service* section of the *TBSM Administrator's Guide* .

Secure connections in earlier TBSM versions

The connections that are or can be secured in TBSM version 4 release 2 are described in this table.

Client	Server	Function	Description
Browser	Dashboard server	Console interface	Secured by default with https.
Dashboard server or Data server	Central user repository (LDAP or OMNIbus)	User authentication & management	Secured with Websphere Application Server
Dashboard server	Dashboard server	Change notification	Part of Dashboard Application Service Hub Load Balancing
IBM Tivoli Monitoring data access client (data server)	Tivoli Monitoring web service (dashboard server)	Policy-based data fetchers to obtain Tivoli Monitoring data	Provides a secure connection for Tivoli Data Warehouse and requires interim fix 2.

Secure connections for TBSM 4.2.1 and later

In addition to the secure connections available in version 4.2, these connections can now be secured in TBSM version 4 release 2.1 and later as described in this table.

Client	Server	Function	Description
Dashboard server	Data server	Configuration	http(s)

Table 12. Secure connections for TBSM servers (continued)

Client	Server	Function	Description
Dashboard server	Data server	Retrieval of data model and other information	RMI port
Data server	Dashboard server	Status or config update notification	RMI port
Data server	Data server	Failover synchronization & health check	RMI port
Dashboard server – chart portlet	Data server – chart web service	Charting design and data retrieval	http(s)
radshell	Data server	Configuration	http(s)
Discovery Library Toolkit	Data server	Configuration	http(s)
Discovery Library Toolkit	TADDM	Data retrieval	RMI port
Data server	Netcool/OMNIbus ObjectServer	Event processing	JDBC

Certificate management for TBSM servers

This topic describes the certificates you need for TBSM servers.

Certificates for dashboard servers

The trust store of each Dashboard server must contain the following signer's certificates for secure connections:

- Data server (primary)
- Data server (backup) – when configured for failover
- OMNIbus (for WebTop and/or ObjectServer authentication)
 - Primary ObjectServer
 - Backup ObjectServer – when configured for failover
- LDAP – for external authentication

You retrieve certificate signer data from these servers using the DASH Console.

Certificates for data servers

The trust store of each Data server must contain the following signer's certificates for secure connections:

- Data server peer – when configured for failover
- Dashboard server – each for each, when there are multiple Dashboard servers
- Netcool/OMNIbus (for event processing and/or ObjectServer authentication - optional)
 - Primary ObjectServer
 - Backup ObjectServer – when configured for failover
- LDAP – for external authentication (optional)

You list and retrieve certificate signer data from these servers using the WebSphere Application Server **wsadmin** tool.

For more information about the `SignerCertificateCommands` command group, see: [SignerCertificateCommands command group for the AdminTask object](#)

For more information about the WebSphere Application Server **wsadmin** tool, see: [Wsadmin tool](#)

Importing trusted JazzSM certificate into the TBSM Dataserver, GUI and server trust stores

1. Export trusted JazzSM certificate from Websphere Administrative Console:
 - a. Login into DASH as smadmin user.
 - b. Go to **Console settings > WebSphere Administrative Console**.
 - c. Launch **WebSphere Administrative** console.
 - d. In WebSphere Administrative console, follow links: **Security > SSL certificate and key management > Key stores and certificates > NodeDefaultTrustStore > Signer certificates**
 - e. Select the root JazzSM certificate and select to extract this certificate.
 - f. Give a valid file name for the certificate and copy the generated certificate to the TBSM Data server.
2. Import the certificate (called `jazz.cer` in this example) to the TBSM Data server front and backend trust stores:

The graphical tool `ikeyman` or the `keytool` utility can be used to import the `jazz.cer`.

- Using `ikeyman` from `/opt/IBM/Tivoli/impact/sdk/jre/bin/ikeyman`, open the front and backend `trust.jks` files selecting signer certificates, and select **Add**.
- Using the `keytool` command:

```
$IMPACT_HOME/sdk/bin/keytool -importcert -alias JazzSMCert -file /tmp/jazz.cer -keystore /opt/IBM/tivoli/impact/wlp/usr/servers/TBSM/resources/security/trust.jks
$IMPACT_HOME/sdk/bin/keytool -importcert -alias JazzSMCert -file /tmp/jazz.cer -keystore /opt/IBM/tivoli/impact/wlp/usr/servers/ImpactUI/resources/security/trust.jks
```

Retrieving signer certificate data for dashboard servers

This topic describes how to retrieve SSL signer certificate data for Dashboard servers.

Before you begin

You need to log into the DASH Console as an administrator.

About this task

To retrieve certificate signer data:

Procedure

1. From the left navigation frame, select **Security > SSL certificate and key management**.
2. Click **SSL certificate and key management** on the page that opens.
3. Click **Key stores and certificates** on the page that opens.
4. Click **NodeDefaultTrustStore** on the page that opens.
5. Click **Signer certificates** on the page that opens.

Perform the **Retrieve from port** action twice, once for the TBSM UI certificate and once for the TBSM nameserver:

6. Click **Retrieve from port** on the page that opens.
7. Enter the **Host**, **Port**, and **alias** values for the server where you want to retrieve a certificate signer.

For example, for a TBSM UI certificate on the default port, enter:

- **Host:** `myhost.ibm.com`

- **Port:** 16311
- **Alias:** tbsmguiserver

For a TBSM nameserver certificate on the default port, enter:

- **Host:** myhost.ibm.com
- **Port:** 9081
- **Alias:** tbsmnameserver

8. Click **Retrieve signer information**.

The following retrieved signer information displays on the page:

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.

What to do next

Repeat this procedure for each server where you need to retrieve signer certificate information.

For more information, see the Page help and Command assistance for the page:

SSL certificate and key management > Key stores and certificates > NodeDefaultTrustStore > Signer certificates > Retrieve from port

Running the secure server command

This topic describes how to create secure connections for a TBSM server.

Before you begin

You need to list and retrieve the certificate signer data for the TBSM server.

About this task

To set up secure communications for a TBSM server:

Procedure

1. Create a secure connections template file with the following command:

```

Windows %TBSM_HOME%\bin\secure_server_config.bat template

UNIX $TBSM_HOME/bin/secure_server_config template

```

The command creates a file named `secure_server_config.date_time.props` in the `TBSM_HOME/bin` directory. This file includes instructions about how to use the `secure_server_config` command to configure secure communications for your server. Read these instructions for the latest information.

2. Change the default data server port from 17311, 17310 to 9081, 9081 in the TBSM SSL configuration ANT script and properties files (`secure_server_config.ant` and `secure_server_config.props`) as follows:

```
SECURE_PrimaryDataServerHTTPPort=9081
SECURE_BackupDataServerHTTPPort=9081
```

3. Rename the generated template properties file from `secure_server_config.date-time.props` to `secure_server_config.props`.
4. Run the script with the **secure-on** (or **secure-off**) parameter.

For example:

```
./secure_server_config secure-on
```

Example

This example shows some sample parameters in the properties files.

```
#####
# TBSM 6.2 Secure Server Configuration Properties file
# Template generated at 20180612104153
#####

# Invoke the script with template target to generate a configuration template
# in the local directory:
#
# Unix: $TBSM_HOME/bin/secure_server_config template
# Windows: %TBSM_HOME%\bin\secure_server_config template
#
# Invoke the script with secure-on target and specifying the configuration file
# to enable TBSM server(s) for secure communications:
#
# Unix: $TBSM_HOME/bin/secure_server_config secure-on
# -Dsecure_server_config.props=[config-file-path]
# Windows: %TBSM_HOME%\bin\secure_server_config secure-on
# -Dsecure_server_config.props=[config-file-path]
#
# Invoke the script with secure-off target and specifying the configuration file
# to restore TBSM server(s) to non-secure communications:
#
# Unix: $TBSM_HOME/bin/secure_server_config secure-off
# -Dsecure_server_config.props=[config-file-path]
# Windows: %TBSM_HOME%\bin\secure_server_config secure-off
# -Dsecure_server_config.props=[config-file-path]
#

#-----
# Informational - files/properties updated with given ports
#-----
# Primary Data Server HTTP Port
#-----
# The following files/properties are updated with the specified port:
# 1. The key REPLICANT.0.PORT in the
# $IMPACT_HOME/wlp/usr/servers/TBSM/apps/nameserver.war/WEB-INF/web.xml
# file on both the primary and the backup data servers.
# 2. The key impact.nameserver.0.port in the
# $IMPACT_HOME/etc/nameserver.props
# file on both the primary and the backup data servers.
# 3. The key impact.nameserver.0.port in the
# $JAZZ_HOME/profile/installedApps/JazzSMNode01Cell/isc.ear/sla.war/etc/nameserver.props
# file on the dashboard server.
# 4. The key impact.sla.serverhttpport in the
# $JAZZ_HOME/profile/installedApps/JazzSMNode01Cell/isc.ear/sla.war/etc/RAD_sla.props
# file on the dashboard server.
# 5. The key impact.server.http.port in the
# $JAZZ_HOME/profile/installedApps/JazzSMNode01Cell/isc.ear/sla.war/etc/RAD_server.props
# file on the dashboard server.
# 6. The key impact.server.http.port in the
# $IMPACT_HOME/etc/TBSM_server.props
# file on the primary data server.
#-----
# Backup Data Server HTTP Port
#-----
# The following files/properties are updated with the specified port:
```

```

# 1. The key REPLICANT.1.PORT in the
# $IMPACT_HOME/wlp/usr/servers/TBSM/apps/nameserver.war/WEB-INF/web.xml
# file on both the primary and the backup data servers.
# 2. The key impact.nameserver.1.port in the
# $IMPACT_HOME/etc/nameserver.props
# file on both the primary and the backup data servers.
# 3. The key impact.nameserver.1.port in the
# $JAZZ_HOME/profile/installedApps/JazzSMNode01Cell/isc.ear/sla.war/etc/nameserver.props
# file on the dashboard server.
# 4. The key impact.sla.serverhttpport.backup in the
# $JAZZ_HOME/profile/installedApps/JazzSMNode01Cell/isc.ear/sla.war/etc/RAD_sla.props
# file on the dashboard server.
# 5. The key impact.replication.replicationhttpport in the
# $JAZZ_HOME/profile/installedApps/JazzSMNode01Cell/isc.ear/sla.war/etc/RAD_server.props
# file on the dashboard server.
# 6. The key impact.replication.replicationhttpport in the
# $IMPACT_HOME/etc/TBSM_server.props
# file on the primary data server.
# 7. The key impact.server.http.port in the
# $IMPACT_HOME/etc/TBSM_server.props
# file on the backup data server.
#-----
# Dashboard Server HTTP Port
#-----
# The dashboard server HTTP port is used when configuring the
# TBSMChartService connection for the chart portlet.
#-----

#-----
# Credentials needed to complete configuration.
# TIPAdminUserid - the userid of the administrative user in DASH
# **Note: This userid must have the
# chartAdministrator role assigned.
#
# TIPAdminPassword - the password of the TIPAdminUserid user. The
# password will be removed from this properties file
# when the secure_server_config utility is run.
#-----
TIPAdminUserid=smadmin
TIPAdminPassword=smadminPassword

#-----
# Parameters used in secure server configuration
#-----
# Primary Data Server secure HTTP Port
# The default value is 17311.
# For the actual value in use, see the WC_defaulthost_secure key in the
# $JAZZ_HOME/profile/properties/portdef.props file on the
# primary data server.
#
SECURE_PrimaryDataServerHTTPPort=17311
#-----

# Backup Data Server secure HTTP Port
# The default value is 17311.
# For the actual value in use, see the WC_defaulthost_secure key in the
# $JAZZ_HOME/profile/properties/portdef.props file on the
# backup data server.
#
SECURE_BackupDataServerHTTPPort=17311
#-----

# Dashboard Server secure HTTP Port
# The default value is 16311.
# For the actual value in use, see the WC_defaulthost_secure key in the
# $JAZZ_HOME/profile/properties/portdef.props file on the
# dashboard server.
#
SECURE_DashboardServerHTTPPort=16311
#-----

#-----
# Parameters used in non-secure server configuration
#-----
# Primary Data Server non-secure HTTP Port
# The default value is 17310.
# For the actual value in use, see the WC_defaulthost key in the
# $JAZZ_HOME/profile/properties/portdef.props file on the
# primary data server.
#

```

```

OPEN_PrimaryDataServerHTTPPort=17310
#-----

# Backup Data Server non-secure HTTP Port
# The default value is 17310.
# For the actual value in use, see the WC_defaulthost key in the
# $JAZZ_HOME/profile/properties/portdef.props file on the
# backup data server.
#
OPEN_BackupDataServerHTTPPort=17310
#-----

# Dashboard Server non-secure HTTP Port
# The default value is 16310.
# For the actual value in use, see the WC_defaulthost key in the
# $JAZZ_HOME/profile/properties/portdef.props file on the
# dashboard server.
#
OPEN_DashboardServerHTTPPort=16310
#-----

```

What to do next

Repeat this procedure for every server in the environment. Secure the connections to Netcool/OMNIbus.

Configure secure communications to Netcool/OMNIBus

These tasks describe how to configure secure communications between the TBSM servers and Netcool/OMNIbus.

Limitation: rad_sendevent: The rad_sendevent utility does not currently support connecting to the Netcool/OMNIBus ObjectServer over SSL. As a result, if your ObjectServer is configured to only accept SSL connections, then rad_sendevent will fail. To work around this limitation, the ObjectServer needs to be configured to accept connections on an additional, non-SSL port.

1. Start the Netcool/OMNIBus Editor
2. Configure the NCOMS server to listen on an additional, non-SSL port.
3. After the configuration changes are completed, restart the ObjectServer.
4. Use the port number you specified in step 2 when you run rad_sendevent.

For more detailed information, see the Netcool/OMNIBus documentation at: http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/index.jsp?topic=/com.ibm.netcool_OMNIBus.doc_7.2.1/install/concept/omn_con_ssl_configuringserved.html

Securing connections to Netcool/OMNIBus

This topics describes how to secure connections to Netcool/OMNIBus.

Before you begin

Before you start this task, you need to:

1. Set up secure connections between the TBSM servers.
2. Create a key database and a self-signed certificate for Netcool/OMNIBus as described here:

[Using SSL for client and server communications](#)

About this task

To set up secure communications with Netcool/OMNIBus, perform the following tasks:

- Import ObjectServer signer's certificates into trust stores of TBSM servers
- Configure ObjectServer data sources on Data server
- Set up each server (primary and backup) in failover configuration

Procedure

1. Back up the following files on the Data server:
 - \$IMPACT_HOME/etc/TBSM_ObjectServer_DS.ds
 - \$IMPACT_HOME/etc/TBSM_OutputObjectServer_DS.ds
 - \$IMPACT_HOME/etc/TBSM_eventbroker.props
2. Update the following files with change the values of the following properties from FALSE to TRUE as appropriate: The variable to update is listed under each file as follows:

TBSM_ObjectServer_DS.ds

USESSLPRIMARY=TRUE

USESSLBACKUP=TRUE

TBSM_OutputObjectServer_DS.ds

USESSLPRIMARY=TRUE

USESSLBACKUP=TRUE

Note: If the USESSLPRIMARY and USESSLBACKUP properties do not exist in the TBSM_OutputObjectServer_DS.ds file, add them as follows:

OutputObjectServer_DS.ObjectServer.USESSLPRIMARY=TRUE

OutputObjectServer_DS.ObjectServer.USESSLBACKUP=TRUE

3. If you configured the secure ObjectServer channel over a different port than the typical 4100, change these port number properties accordingly.

TBSM_ObjectServer_DS.ds

ObjectServer_DS.ObjectServer.PRIMARYPORT

ObjectServer_DS.ObjectServer.BACKUPPORT

TBSM_OutputObjectServer_DS.ds

ObjectServer_DS.ObjectServer.PRIMARYPORT

ObjectServer_DS.ObjectServer.BACKUPPORT

4. Restart the Data server.

What to do next

Verify the configuration.

Note: Netcool/WebTop and the DASH Console support secure connections to Netcool/OMNIbus.

For Netcool/WebTop, see: [Creating secure connections](#)

Configuring dashboard server secure connection to Netcool/OMNIbus as user repository

This topic describes how to set up encrypted communications between the dashboard server and the Netcool/OMNIbus ObjectServer for an environment where the ObjectServer is the user registry.

Before you begin

It is assumed that Netcool/OMNIbus has already been configured for secure communications. If that is not the case, then see the Netcool/OMNIbus documentation to complete this configuration before you execute the procedures in this task. For more information see,; [Using SSL for client and server communications](#) on the Netcool/OMNIbus information center.

You need to ensure that the tbsmadmin user exists in the ObjectServer repository, and that the user is enabled in the ObjectServer. You can enable the user using the nco_config tool on Linux (**Administrator tool** on Windows), or by logging into the ObjectServer directly and using the following command:

```
./nco_sql script "ALTER USER 'tbsmadmin' SET ENABLED TRUE"
```

About this task

Follow these steps to establish a secure channel for communications between a TBSM dashboard server and the ObjectServer when the ObjectServer is being used as a user registry.

Procedure

1. Retrieve the ObjectServer certificate and save it into the trust store of the dashboard server as described in [“Retrieving signer certificate data for dashboard servers”](#) on page 139.
2. Enable SSL in the Impact server running within JazzSM by modifying the following property in this file on the DASH server `/opt/IBM/JazzSM/profile/installedApps/JazzSMNode01Cell/isc.ear/sla.war/etc/RAD_server.props`:

```
impact.server.ssl_enabled=true
```

This enables SSL for the RMI port.
3. Restart the dashboard server.

Configuring Web GUI connections to Netcool/OMNIBus

This topic describes how to configure secure communications between the dashboard server's Web GUI instance and Netcool/OMNIBus.

Before you begin

Netcool/OMNIBus needs to be configured to operate over SSL and the signer certificate for the ObjectServer (certificates for each server when configured for failover) has already been added to the trust store of the dashboard server.

About this task

To configure these connections:

Procedure

1. Make backups of the following files:

```
$NCHOME/omnibus_webgui/etc/server.init  
$NCHOME/omnibus_webgui/etc/datasources/ncwDataSourceDefinitions.xml
```

2. Edit the `server.init` file and set the trust store password for the property:

```
webtop.ssl.trustStorePassword
```

The default password for WebSphere trust stores is WebAS, so the property would look as follows after being edited:

```
webtop.ssl.trustStorePassword: WebAS
```

3. Save the file changes.
4. Edit `ncwDataSourceDefinitions.xml` file and change the value of the `ssl` attribute from `false` to `true` as follows:

```
<ncwPrimaryServer>  
  <ncwOSConnection host="myhost" port="4100" ssl="true"/>  
</ncwPrimaryServer>
```

Note: The values of the port attributes in the connection information above may also need to be changed if Netcool/OMNIBus was configured to use SSL on a different port.

5. Save the file changes.
6. Restart the dashboard server.

For more information about Web GUI secure connections see [Creating secure connections](#) on the Web GUI information center.

Securing connections to the Discovery Library Toolkit

The Discovery Library toolkit is capable of using Secure Socket Layer (SSL) for its communications with both the TBSM data server and the TADDM server.

Enablement of these is controlled through properties in the `$TBSM_HOME/XMLtoolkit/bin/xmltoolkitsvc.properties` file.

Note: The SSL connection secures the communications with the TADDM server and the TADDM API's, but does not secure the communications with the TADDM DB when using the JDBC connection type.

Properties for TADDM secure connections

These properties control the SSL connections to TADDM.

DL_TADDM_SSL

Set to true for SSL

If the `DL_TADDM_SSL` property is set to true, you must copy `jssecacerts.cert` file from the `$TADDMHOME/dist/etc` directory and place it in the `../XMLtoolkit/sdk/etc` directory.

DL_TADDM_SSL_Port

Set to the port that TADDM is listening on, the default is 9531. The `collation.properties` file on the TADDM server contains the SSL port value. The property is:

```
#Port for SSL API
com.collation.api.ssl.port=9531
```

Properties for TBSM Data Server secure connections

These properties configure the SSL connections to the TBSM data server.

Note: Since both the data server and the Discovery Library Toolkit are installed on the same machine this security option is normally not required.

DL_TBSM_SSL

Set to true for SSL

DL_TBSM_HTTP_Port

Specify the data server's SSL port.

Verifying secure connections

This topic describes how to verify secure connections.

To verify that your secure connections are working properly:

- Ensure all servers start successfully. Check `SystemOut.log`, `trace.log`, and so on.
- View the **Service Administration** and **Service Availability** pages, portlets, and functions within portlets in the Dashboard Application Service Hub console.
- Exercise failover function.
- Exercise event processing/status updates.

Typical errors and problems

Typical errors and problems include:

- Secure connections have been configured but trust has not been established because one or more certificates are missing from a client-side trust store – look for messages like “trust” could not be established

- One end of a connection is secured but the other isn't (for example, trying to use http when https is required) can happen if secure server script is not run on all servers
- Trying to establish SSL connection to nonsecure port; make sure that the correct port numbers were specified in the secure server properties file.

Example error message

Here is an error message indicating that no certificate was found:

```
Caused by: HTTP transport error: javax.net.ssl.SSLHandshakeException: co
m.ibm.jsse2.util.h:
  No trusted certificate found
    at com.sun.xml.rpc.client.http.HttpClientTransport.invoke(HttpClientTra
nsport.java:140)
    at com.sun.xml.rpc.client.StreamingSender._send(StreamingSender.java:92)
    at com.micromuse.sla.soap.RADSoapFacadeIfc_Stub.setUserForSession(RADSoa
pFacadeIfc_Stub.java:4379)
```

6.2.0.3 Enabling TLSv1.2 for the Dashboard server

This topic describes how to enable TLSv1.2 for the Dashboard server.

The Dashboard server supports the following protocol options:

- SSL
- SSLv2
- SSLv3
- SSL_TLS
- SSL_TLSv2
- TLS
- TLSv1
- TLSv1.2

To enable TLS 1.2, you should choose one of the following options:

- SSL_TLSv2: Supports TLS 1.0, TLS 1.1 and TLS 1.2
- TLSv1.2: Only supports TLS 1.2.

Ensure all remote outbound and inbound connections to the server support TLS 1.2 before proceeding.

Then perform the following steps:

1. Configure the Administrative Console to enable TLSv1.2:
 - a. Log in to the Dashboard server GUI and navigate to **Console Settings > WebSphere Administrative Console** then click the **Launch** button.
 - b. Click **Security > SSL certificate and key management**.
Under **Related Items**, click **SSL configurations > NodedDefaultSSLSettings**.
Under **Additional Properties**, click **Quality of protection (QoP) settings**.
 - c. Change the **Protocol** to either SSL_TLSv2 or TLSv1.2.
 - d. Click **OK** and **Save**.
2. Update the `ssl.client.props` configuration file for the Dashboard server:
 - a. Open the following configuration directory in a text editor file:
`/opt/IBM/JazzSM/profile/properties/ssl.client.props`
 - b. Change the value of the `com.ibm.ssl.protocol` property to `SSL_TLSv2` or `TLSv1.2`:

```
com.ibm.ssl.protocol=SSL_TLSv2
```

3. Update the Dashboard server profile:

a. Open the following configuration file in a text editor:

```
/opt/IBM/JazzSM/profile/config/cells/JazzSMNode01Cell/security.xml
```

b. Change the value of the `sslProtocol` attribute to `SSL_TLSv2` or `TLSv1.2`:

```
sslProtocol="SSL_TLSv2"
```

4. Restart the Dashboard server.

Replacing the default SSL certificates with certificates signed by a certificate authority

These topics describe how to switch from the default IBM supplied Secure Sockets Layer (SSL) certificate to a certificate signed by a certificate authority (CA).

After you change the certificate, you will be able to secure the client browser to server connection. These instructions do not cover securing the backend (server to server) connections.

Overview

To use a certificate signed by a CA, open the TBSM Dashboard server console and perform **all** of the following actions. The procedures are described in more detail in the topics which follow.

1. Create a personal certificate request to obtain a certificate that is signed by a CA. You can find the general Websphere Application Server instructions at: http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/tsec_sslcreateCArequest.html
2. Receive a certificate issued by a certificate authority. You can find the general Websphere Application Server instructions at: http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/tsec_sslreceiveCAcert.html

Note: This procedure allows you to continue using the console and access TBSM (with the exception of the Service Details portlet), which requires that the signer certificate be added to the Trust Store as described in the next step. Please note that TBSM cannot use this certificate until it is enabled.

Important: In some cases, such as when a certificate authority uses intermediate certificates, you may receive multiple certificates. In these cases install all the certificates issued by the certificate authority.

3. Adding a signer certificate to the Trust Store. You can find the general Websphere Application Server instructions at: http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/tsec_ssladdsignercert.html

Note: If you restart the TBSM Dashboard server without completing this step (adding the signer certificate to the Trust Store), the TBSM Dashboard will not be able to restart. And therefore, you'll have no access to TBSM whatsoever (not even the items previously accessible when step 2 was completed). However, please note that this step once completed, will allow the Dashboard server to connect to the Data server (without the need to restart any of the components). In this step, all we need to do, is add the CA's Intermediate certificate.

4. Enable the new SSL Certificate.
5. SSL Signer Exchange

Note: Failure to perform this action, could result in a failure of any future maintenance properly installing on the Dashboard server. Where the installer may need to first check on the status of the Dashboard server but, cannot log on, due to missing Signer trust information.

Creating a personal certificate request

This topic describes how to create a personal certificate request to obtain a certificate that is signed by a CA.

Before you begin

You need to log on to TBSM as a user with administrator permissions.

About this task

Complete the following steps in the administrative console:

Procedure

1. Click **Security > SSL certificate and key management >**
2. From the page that opens, under Related items, click **Key stores and certificates.**
3. From the page that opens, click **NodeDefaultKeyStore**
4. Under Additional Properties, click **Personal certificate requests.**
5. From the page that opens, click **New** and the Configuration: General Properties window opens
6. Type the full path of the **File for certificate request file.** The certificate request is created in this location.
7. Type an alias name in the **Key label** field. The alias identifies the certificate request in the keystore.
8. Type an organization value. This value is the O value in the certificate DN.
9. You can configure one or more of the following **optional** values:
 - a) Select a key size value. The default key size value is 1024 bits.
 - b) Type an organizational unit value. This organizational unit value is the OU value in the certificate DN.
 - c) Type a locality value. This locality value is the L value in the certificate DN.
 - d) Type a state or providence value. This value is the ST value in the certificate DN.
 - e) Type a zip code value. The zip code value is the POSTALCODE value in the certificate DN.
 - f) Select a country value from the list. This country value is the C= value in the certificate request DN.
10. Click **Apply** and the certificate request is saved displays in the panel.

Results

The certificate request is created in the specified file location in the keystore. The request functions as a temporary placeholder for the signed certificate until you manually receive the certificate in the keystore.

Note: Key store tools (such as iKeyman and keyTool) cannot receive signed certificates that are generated by certificate requests from WebSphere Application Server. Similarly, WebSphere Application Server cannot accept certificates that are generated by certificate requests from other keystore utilities.

What to do next

At this point the request can be sent to the Certificate Authority (CA). Once you receive the certificate, go to the next procedure for receiving the certificate.

Receiving the certificate

This topic describes how to receive a certificate issued by a certificate authority

Before you begin

You need to create a personal certificate request and the certificate authority needs to issue the certificate.

About this task

WebSphere Application Server can receive only those certificates that are generated by a WebSphere Application Server certificate request. It cannot receive certificates that are created with certificate requests from other keystore tools, such as iKeyman and keyTool.

Complete the following steps in the administrative console:

Procedure

1. Click **Security > SSL certificate and key management**
2. On the page that opens, under Configuration, click **Manage endpoint security configurations**.
3. On the page that opens, select either the **Inbound** or **Outbound** node SSL configuration, depending on the certificate you are receiving.
For example: to select the Inbound connection, click:
Inbound > JazzSMNode01Cell -> Nodes -> JazzSMNode01(NodeDefaultSSLSettings)
4. On the page that opens, under Related Items, click **Key store and certificates**, under 'Related Items'
5. On the page that opens, click **NodeDefaultKeyStore**,
6. On the page that opens, under Additional Properties, click **Personal certificates**.
7. On the page that opens, click **Receive a certificate** from a certificate authority.
8. On the page that opens:
 - a) Type the full path and name of the certificate file.
 - b) Select a data type from the list.
9. Click **Apply**
10. On the message box that opens, click **Save**.

Results

The personal certificate is added to the **NodeDefaultKeystore**. The keystore contains a new personal certificate that is issued by a CA. The original certificate request is changed to a personal certificate.

What to do next

Add a signer certificate to the new keystore.

Adding a signer certificate to a keystore

This topic describes how to add a signer certificate to the Trust Store.

Before you begin

You need to receive a certificate issued by a certificate authority.

About this task

Signer certificates establish the trust relationship in SSL communication. You can extract the signer part of a personal certificate from a keystore, and then you can add the signer certificate to other key stores.

Note: If you restart the TBSM Dashboard server without completing this step (adding the signer certificate to the Trust Store, the TBSM Dashboard will not be able to connect to the TBSM Data server at all. And therefore, you'll have no access to TBSM whatsoever (not even the items previously accessible when step 2 was completed). However, please note that this step once completed, will allow the Dashboard server to connect to the Data server (without the need to restart any of the components). In this step, all we need to do, is add the CA's Intermediate certificate.

Complete the following steps in the administrative console:

Procedure

1. Click **Security > SSL certificate and key management**.
2. From the page that opens, under Configuration settings, click **Manage endpoint security configurations**.
3. On the page that opens, select either the **Inbound** or **Outbound** node SSL configuration, depending on the certificate you are adding.
For example: to select the Inbound connection, click:
Inbound > JazzSMNode01Cell -> Nodes -> JazzSMNode01(NodeDefaultSSLSettings)
4. On the page that opens, under Related Items, click **Key store and certificates**, under 'Related Items'
5. On the page that opens, click **NodeDefaultTrustStore**.
6. On the page that opens, under Additional Properties, click **Signer certificates**.
7. On the page that opens, click **Add**.
8. From the page that opens:
 - a) Enter the alias for the signer certificate in the **Alias** field .
 - b) Enter the full path to the signer certificate file in the **File name** field
 - c) Select the data type from the list in the **Data Type** field
 - d) Click **Apply**.
9. In the message box that opens, click **Save**.

Results

When these steps are completed, the signer from the certificate file is stored in the keystore. You can see the signer in the keystore files list of signer certificates. Use the keystore to establish trust relationships for the SSL configurations.

What to do next

You need to enable the new SSL Certificate.

Enabling the new SSL certificate

This topic describes how to enable an SSL certificate.

Before you begin

You need to add a signer certificate to the Trust Store.

About this task

Complete the following configuration steps in the administrative console:

Procedure

1. Click **Security > SSL certificate and key management**
2. On the page that opens, under Configuration, click **Manage endpoint security configurations**.
3. On the page that opens, select either the **Inbound** or **Outbound** node SSL configuration, depending on the certificate you are receiving.
For example: to select the Inbound connection, click:
Inbound > JazzSMNode01Cell -> Nodes -> JazzSMNode01(NodeDefaultSSLSettings)
4. On the page that opens, select the **Certificate alias in keystore** you specified when you added the certificate.
5. Click **Apply**
6. On the message box that opens, click **Save**.

Results

The next time a user logs into the TBSM console, the Dashboard server will return the new, CA-signed certificate to the browser.

What to do next

Run SSL signer exchange command to add the signer to the trust store.

Adding SSL signer exchange

This topic describes how to add an Secure Socket Layer signer exchange for the new certificate.

Before you begin

The SSL certificate must be enabled.

About this task

Complete this task at a command prompt on the Dashboard server host.

Important: If you do not complete this task, fix packs and interim fixes may fail when they are applied to the Dashboard server.

Procedure

1. Enter the command:

```
$TIP_HOME/bin/serverStatus.sh -all
```

2. Enter 'y' when prompted to do so by the command.

Example

In this example the signer is added for a certificate from Verisign.

```
$JAZZSM_HOME/profile/bin/serverStatus.sh -all
ADMU0116I: Tool information is being logged in file
           $JAZZSM_HOME/profiles/logs/serverStatus.log
ADMU0128I: Starting tool with the TIPProfile profile
ADMU0503I: Retrieving server status for all servers
ADMU0505I: Servers found in configuration:
ADMU0506I: Server name: server1

*** SSL SIGNER EXCHANGE PROMPT ***
SSL signer from target host null is not found in trust store
$JAZZSM_HOME/profiles/etc/trust.p12.

Here is the signer information (verify the digest value matches what is displayed
at the server):

Subject DN:   CN=vmwlnx4l3j9d.tivlab.raleigh.ibm.com, OU=Terms of use at
www.verisign.com/cps/testca (c)05, OU=TBSM, O=IBM, L=RTP, ST=NORTH CAROLINA, C=US
Issuer DN:    CN=VeriSign Trial Secure Server CA - G2, OU=Terms of use at
https://www.verisign.com/cps/testca (c)09, OU="For Test Purposes Only.
No assurances.", O="VeriSign, Inc."...
Serial number: 13583817932481119925668512051223819622
Expires:      Fri Jul 17 19:59:59 EDT 2009
SHA-1 Digest: 1D:D1:ED:0C:BF:FE:F7:B1:80:0C:03:5D:75:FF:15:6B:F9:15:8E:F8
MD5 Digest:   C7:89:4D:79:2F:0F:F6:97:04:9E:B6:2D:CC:20:D9:53

Add signer to the trust store now? (y/n)
```

Enabling the Policy Editor from the TBSM Rules page

About this task

Before accessing the **Policy Editor** from the **TBSM Rule Editor**, you must first enable Single Sign On between the Impact servers and the Jazz SM server. For details of how to do this, see the following page on the Netcool/Impact Knowledge Center:

https://www.ibm.com/support/knowledgecenter/SSSHYH_7.1.0.15/com.ibm.netcoolimpact.doc/admin/imag_configure_single_signon.html

Once SSO is enabled, you should enter the following URL in a new tab:

https://<TBSM_Dashboard_Server>:9081/restui/policyui/policy/NumericAttributeFunctions/loadPolicyOrTemplate?policyName=NumericAttributeFunctions&template=null

Then accept the certificates in the browser and the Policy Editor should then be accessible.

Modifying the tivoli_eif.props file

About this task

The `tivoli_eif.props` file is self-documented in that it contains each supported property name along with its default value. All of these properties are contained in the section that begins with the following lines:

```
#####  
#  
# Property Name                Default  
#  
# Generic Properties
```

If not changed, the default value for any property is the value that appears in the **Default** column.

It is strongly recommended that any property value that is modified from its default value be added to this section at the bottom of the file:

```
#####  
#  
# Add your settings here  
#  
#####
```

Doing this ensures that the original default value and the changed value are available for debugging and problem support.

Ensure service templates were created

About this task

`$TBSM_HOME/install/BSM_Templates.radsh` is a radshell script that defines service templates for common TBSM services.

During a simple installation of the Data server, the TBSM installation program launches a tool to run the script, but does not wait for the tool to complete. The tool might not finish by the time you reboot because it must wait for the TBSM server to start completely. Additionally, if the tool tries to add the templates before TBSM initializes, you might see failure errors.

Look at the log files `templates.out` and `templates.err` in the directory:

```
$TBSM_HOME/logs/install/
```

If there are errors or exceptions in log files, run the tool manually to create the service templates. To run the tool:

Windows Type the following from a command prompt:

```
%TBSM_HOME%\bin\tbsmdefaultimport
```

Assuming that you have sourced `<installDir>/tivoli/tbsm/bin/setupTBSMDash.sh` or `<installDir>/tivoli/tbsm/bin/setupTBSMData.sh`, type the following from a command prompt:

```
$TBSM_HOME/bin/tbsmdefaultimport
```

Chapter 12. Troubleshooting installation issues

About this task

This section describes symptoms of and resolutions for common installation issues. Log files that the TBSM installation program uses are also listed.

Common installation issues

Problem: Error message when running `tbsm_db_update.sql` command:

If you are importing the TBSM schema into an existing Netcool/OMNIBus ObjectServer, you may receive error messages similar to the following when running `tbsm_db_update.sql` command:

```
ERROR=Object exists on line 83 of statement
'-----',
  at or near 'BSM_Identity' (0 rows affected) (0 rows affected)
(0 rows affected)
ERROR=Object not found on line 15 of statement
'-----'
...', at or near 'service_deps'
(0 rows affected)
(0 rows affected)
(0 rows affected)
(0 rows affected)
(0 rows affected)
(0 rows affected)
```

These messages can be ignored. The ObjectServer will return an error if a user tries to add a column that already exists, which explains the error on `BSM_Identity`. It will also return an error if a user drops a table that doesn't exist, which explains the second error.

Problem: Insufficient disk space message on Pre-Installation Summary panel

If you get this message indicating that you need more disk space, you need to free up disk space before you continue with the installation.

```
Insufficient disk space found for installation target
```

Do not proceed with the install until after you free up disk space. Even if the Install button is enabled, the installation may fail.

Do the following based on the type of installation:

1. Free up at least 3 GB of space on the drive on where you want to install TBSM
2. If you are running an upgrade installation, free up at least 3 GB of space on the backup destination drive
3. Click **Previous** on the Pre-Installation Summary panel to return to the WebSphere Information panel.
4. Click **Next** to proceed to the Pre-Installation Summary panel.

The Insufficient disk message should now be gone, as long as you have at least 3 GB available in both destinations. At this point, you can click **Install** to proceed with the installation.

Problem: Error seen when installing TBSM : Destination Host Name is unreachable

The TBSM installer uses the echo protocol/service to determine host availability when validating dependent applications like DASH, ObjectServer, DB2, and so forth. If the echo service is unable to determine host availability, the TBSM installer displays the following messages:

```
CRIMA1209E ERROR: Validation failed for property user.OmniBushostName. ERROR: Host Name is unreachable.
CRIMA1209E ERROR: Validation failed for property user.JazzSMHostName. ERROR: The Host Name is not reachable
```

```
CRIMA1209E ERROR: Validation failed for property user.DataServerDLBExportDashBoardServerI.  
ERROR: Destination Host Name is unreachable. WARNING: JazzSM is not available on specified  
Host and Port.  
CRIMA1210W WARNING: Warnings occurred during validation for property user.JazzSMPortNumber.  
WARNING: Port is not listening on host.
```

Solution

On UNIX operating systems, if a firewall separates the TBSM servers from the JazzSM or OMNibus servers, port 7 must be open on the firewall while the TBSM installer is running (port 7 is a standard port for the echo service). If port 7 is not open, you will see the errors shown above and the install will not proceed.

Database configuration utility issues

These topics describe issue with the Database configuration utility.

Database configuration installer fails

Symptoms

The database configuration installer fails.

Solution

The database name contains invalid characters. For example, you used Chinese characters in the name. If you specify an invalid name, the database configuration installer will fail, but the files will still be installed. You can modify the database name in the properties file and re-run `tbsm_db` script after installation to complete the database creation. Otherwise, uninstall the tool, and re-install specifying a valid database name. Valid characters for names:

- A through Z. When used in most names, characters A through Z are converted from lowercase to uppercase.
- 0 through 9.
- ! % () { } . - ^ ~ _ (underscore) @, #, \$, \ (backslash), and space.

For more information about valid characters for DB2 names, see the documentation for the version of DB2 you are using here:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=%2Fcom.ibm.db2.udb.doc%2Fdoc%2Ft0021844.htm>

TBSM database install fails when user id contains a hyphen

Symptoms

When you install the TBSM Database, the database user name you specify must not have a hyphen (-) in the name, otherwise some of the SQL is likely to fail.

Cause

The TBSM database does not recognize a user id containing a hyphen.

Solution

When you install the TBSM database, rename the Administrator user name and Administrators group and ensure they do not contain a hyphen.

Russian Windows TBSM and DB2 install fails

Russian Windows TBSM and DB2 install fails.

Symptom

You cannot install TBSM, DB2, or configure the DB2 databases for TBSM when you are logged in with a user id containing Russian characters.

Cause

DB2 does not recognize a user id with non-English characters. The Windows Administrator user id on a Russian operating system has Russian characters in the name. Typically, you do not have these problems in unix because the user names have English characters.

Resolution

To install DB2, rename the Russian Administrator user id and Administrators group to have english characters only. Log in with the renamed Administrator user id. Install DB2 while logged in with the renamed Administrator id. You also need to run the TBSM Database Configuration utility with the same renamed Administrator user id.

However, to install TBSM, you must log in with a user id with English characters that is a member of the Administrators group. The user id cannot be the renamed Russian Administrator id. You must log in with a user id that was originally created with English characters. One example is to use the db2admin user id that is normally created during a DB2 installation.

Cannot create database

The database configuration installer disables the database creation option.

Symptoms

The installer does not allow you to select **yes** to create the database during the installation

Solution

Windows On Windows, the installer must be run from a command window that has the DB2 command environment initialized. If the window being used was opened with db2cmd and the installer still does not allow you to select **yes**, then run the installer from a window opened with db2cadmin.

UNIX On UNIX, the user must have one of the following authorizations: SYSADM or SYSCTRL.

Database configuration error messages

Addressing errors when the database configuration completes.

Symptoms

The following message is displayed when the database configuration utility completes:

```
The schema was not created successfully.  
Check the log for SQL errors (such as DB21034E SQL errors).
```

Solution

The database and schema was created, but the log contains DB2 error messages that indicate not all of the schema processing was successful. The DB21034E error is followed by a SQL error message. The error messages may not be problem dependent on your DB2 environment. Review each message

following the DB21034E messages to determine if you need to correct your environment and re-execute the database configuration utility.

For example, the following error scenario is ignored by the database configuration utility. It prompts for the userid and password to connect to the database it created. A **GRANT** command is issued to grant DBADM authority to that userid. If the specified userid is the same as the login userid, then the **GRANT** command results in a DB2 error and the SQL error SQL0554N that authorization cannot grant a privilege or authority to itself. This is an acceptable error message and has no effect on the database schema creation process.

You should review the log file (db2_stdout.log) to determine if other DB21034E error scenarios are fatal. Similar to the **GRANT** error message, the other DB21034E error scenarios may not be a problem. Another acceptable error message if the tablespace already exists. This may occur if you previously executed the database configuration utility and did not drop the database before executing the utility again. An example of a problem is if the tablespace error message indicates that the system is full, and therefore could not create the tablespace.

Windows issues

Windows installation issues.

RAD shell issue connecting to the Data server

By default, you cannot use an ! (exclamation) character in the tbsmadmin password on Windows systems. It results in errors in the TBSM trace logs and the Rad Shell does not connect to the Data server. This topic outlines how to encrypt the tbsmadmin passwords that contain an ! character.

Symptoms

You cannot connect to the Data server using Rad Shell and no Rad Shell prompt is displayed. An error is created in the TBSM trace logs:

```
com.ibm.ws.wim.adapter.file.was.FileAdapter login
com.ibm.websphere.wim.exception.PasswordCheckFailedException:
CWWIM4512E The password match failed.
```

Note: This issue does not occur when you log in from the TBSM user interface.

Cause

This is a limitation in the use of the ! character in Windows batch programming.

Resolution

To ensure that the correct encryption is generated for the tbsmadmin password when it contains an ! (exclamation) character, use the nci_crypt command, located in the \$TBSM_HOME/bin directory, to generate the encrypted password and escape the ! character with a ^ character. For example, to encrypt a password, t!padm!m, execute the command:

```
nci_crypt "t^!padm^!n"
```

Add the output of this command as the value for property impact.server.vmm.admin.password in the property files in \$TBSM_HOME/etc..

Dashboard server

- \$TBSM_HOME/etc/server.props
- \$TBSM_DASHBOARD_SERVER_HOME/etc/RAD_sla.props

Data server

\$TBSM_HOME/etc/server.props

Dashboard server cannot connect to the Data server on Linux

On Linux systems, if the Dashboard server cannot connect to the Data server, the host name of the machine might be using the loopback address 127.0.0.1.

About this task

To correct this problem, perform the following steps:

Procedure

1. Do one of the following actions to each machine on which a Data server or Dashboard server is installed:

Ensure that the host name in the `/etc/hosts` file uses the IP address `9.x.x.x` instead of the loopback address `127.0.0.1`.

Modify the hosts line in the `/etc/nsswitch.conf` file so that DNS (dns is accessed before local files files).

2. Restart all TBSM components.

You must restart the Data server, the Dashboard server, and OMNIbus.

Clearing the Java plug-in cache on Windows

About this task

1. Open **Control Panel** and click **IBM Control Panel for Java** to launch the Java Control Panel program.
2. On the **General** tab, click **Delete Files** in the **Temporary Internet Files** section at the bottom of the panel.
3. On the **Delete Temporary Files** window that opens, ensure that **Downloaded Applets** is checked.
4. Click **OK**.
5. Click **Cancel** to close the Java Control Panel program.

Clearing the Java plug-in cache on UNIX

About this task

1. Launch the Java Control Panel program by running `$JAVA_HOME/jre/bin/ControlPanel` from a shell prompt.
2. On the **General** tab, click **Delete Files** in the **Temporary Internet Files** section at the bottom of the panel.
3. On the **Delete Temporary Files** window that opens, ensure that **Downloaded Applets** is checked.
4. Click **OK**.
5. Click **Cancel** to close the Java Control Panel program.

Default TBSM groups not created during the installation process

This topic describes how to manually create TBSM groups and assign roles.

Symptoms

Default TBSM groups are not created during installation.

Resolution

Completing the following procedure:

1. Go to `$TBSM_HOME/bin`.
2. Run the following commands to create TBSM groups:
 - a. **UNIX** `./vmm_create_entities.sh <WAS admin userid> <WAS admin password> <repository user suffix> <repository group suffix>`
 - b. **Windows** `./vmm_create_entities.bat <WAS admin userid> <WAS admin password> <repository user suffix> <repository group suffix>`
3. For file-based authentication, run the following command:

```
./vmm_create_entities.sh tbsmadmin password "o=defaultWIMFileBasedRealm" "o=defaultWIMFileBasedRealm"
```

4. For OMNIbus authentication, run the following command:

```
./vmm_create_entities.sh <WAS admin userid> <WAS admin password> "o=netcool10bjectServerRepository" "o=netcool10bjectServerRepository"
```

5. For LDAP, use the LDAP user and group suffixes, such as `ou=tivoli,dc=ibm,dc=com`.
6. Run the following command to assign roles to the created group:

- a. **UNIX** `./assign_group_roles.sh tbsmadmin password`
- b. **Windows** `assign_group_roles.bat tbsmadmin password`

Netcool/OMNIbus install fails on SuSE with ssh access

Using an SSH session to install TBSM on some versions of SUSE Linux can result in a problem during the Netcool/OMNIbus.

Symptoms

The install fails at Netcool/OMNIbus install step on SUSE Linux when ssh is used to access the machine.

The log `$HOME/IA-Netcool-Omnibus.log` shows:

```
data/time STDERR:  
com.ibm.ac.si.tpreg.TpregRegistrationFailedException: ACUTRI0024E An error occur red  
registering the Managed Resource Properties for the resource type OSRT:PalMOS.
```

Cause

Caused by Deployment Engine and Java issues:

1. When you run locally, the hostname will be tried to pick from the cache first and in this case hostname is displayed from the cache. Platform like Solaris does not want to give us a fully qualified domain name. Even it is the same case happening in SUSE linux. That is the reason why you are seeing only short name without `in.ibm.com` when you run locally. But java always try to give fully qualified domain name by doing reverse lookup. This will happen only when the hostname is resolved through DNS. Resolution of hostname through DNS is not happened when you ran locally because hostname is already picked from cache.
2. But when you run from remote machine through ssh the name resolution for hostname will happen through DNS. So Java tries to return fully qualified domain name even if platform like Solaris, SUSE linux, etc does not want to give fully qualified domain name. Java does a reverse lookup and tries to pick the hostname from `/etc/hosts` file inorder to give fully qualified domain name. You can edit `/etc/hosts` file and change `xx.in.ibm.com` to `xx`. Then through ssh ,only short name is found. However, it is not recommended to change the default setting of `/etc/hosts` file.

Resolution

Place short name of the host in `/etc/hosts` file.

TBSM server fails after Netcool/Impact install/migration

If you install and migrate another Netcool/Impact server on a TBSM server host, the TBSM server can fail.

Symptoms

The TBSM Data or Dashboard server fails after you install and migrate another Netcool/Impact server.

Cause

TBSM and Netcool/Impact share a common keystore configuration file and the TBSM file was over written by Netcool/Impact. You need to follow the guidelines in the Planning section of this guide before you attempt to install Netcool/Impact on a TBSM server host.

Resolution

Copy the keystore file that was backed up during Netcool/Impact migration to the `IMPACT_HOME/etc` directory and update property `impact.keystore.location` in `IMPACT_HOME/etc/TBSM_server.props` to point to the new location.

Separating the Data server and Dashboard server with a firewall

If the Data server and Dashboard server are separated by a firewall, the connection between the servers is completed on a random port. If the firewall does not allow the connection. The Dashboard server does not initialize correctly and status and configuration changes are not sent to the Dashboard server.

Symptoms

An error message is displayed:

```
Connection refused to host: <name or IP address>
```

In addition, a message similar to this is added to the Data server logs:

```
updatepublish 1 com.micromuse.sla.updatepublisher.ClientUpdate
HandlerThread run ENTER^ERROR WRITING to client
<dashboard host>:17543 we will remove this client updater.
Connection refused to host: <dashboard host>; nested exception is:
java.net.ConnectException: Connection timed out.
```

The error message displays the port of the RMI registry, even if the communication fails when it is running RMI stubs using a different port. This can be misleading as `netstat` might display an established connection to port 17543. However, TBSM fails to run RMI stubs on a random port.

Causes

The RMI registry port is defined by the parameter: `impact.server.rmiport`. On the Data server this has a default value of 17542 and is stored in `$IMPACT_HOME/etc/TBSM_server.props`. On the Dashboard server this has a default value of 17543 and is stored in `$JAZZSM_HOME/profile/installedApps/JazzSMNode01Cell/isc.ear/sla.war/etc/RAD_server.props`. By default, a random port is used when the running RMI stubs on a remote server. This can cause issues when a firewall is present between the servers and the random port is blocked.

Resolution

If a firewall exists between the Data server and the Dashboard server, you might have to open and specify the server port used to run RMI stubs. To specify this port:

1. On the Data server, locate the file: `$IMPACT_HOME/etc/TBSM_server.props`.

- Using a text editor, open and edit the file to add the lines:

```
impact.rmiPortRangeStart=17544  
impact.rmiPortRangeEnd=17544
```

- On the Dashboard server, edit the `$JAZZSM_HOME/profile/installedApps/JazzSMNode01Cell/isc.ear/sla.war/etc/RAD_server.props` to add the lines:

```
impact.rmiPortRangeStart=17544  
impact.rmiPortRangeEnd=17544
```

Note: 17544 is a suggested port, use a different free port value where required. You can also use a range of more than 1 port if required for your Netcool/Impact configuration. The port you choose must be different to the `impact.server.rmiport` which is used for the RMI registry.

Chapter 13. Reference

Reference information is organized to help you locate particular facts quickly.

Log files that TBSM uses

On both UNIX and Windows systems, TBSM log files are located in the *installDir/impact/logs* and *installDir/tbsm/logs* directory.

In TBSM 6.2, you can refer to the following log directories:

- Install logs :

```
/var/ibm/InstallationManager/logs
```

- TBSM Data server logs :

```
/opt/IBM/tivoli/impact/logs  
/opt/IBM/tivoli/impact/wlp/usr/servers/TBSM/logs  
/opt/IBM/tivoli/tbsm/logs
```

- TBSM Dashserver logs :

```
/opt/IBM/JazzSM/profile/logs  
/opt/IBM/tivoli/impact/wlp/usr/servers/ImpactUI/logs
```

- OMNIbus Server logs :

```
/opt/IBM/tivoli/netcool/omnibus/logs
```

- OMNIbus WebGUI logs :

```
/opt/IBM/tivoli/ui/logs
```

Log files generated by the installation of Discovery Library Toolkit

The XMLToolkit is installed as part of TBSM Data Server.

The logs related to toolkit are in the following directories:

```
UNIX /opt/IBM/tivoli/tbsm/XMLtoolkit/log
```

```
Linux /opt/IBM/tivoli/tbsm/XMLtoolkit/log
```

```
Windows C:\Program Files\IBM\tivoli\tbsm\XMLToolkit\logs
```

Sample response file

The TBSM product DVD also includes these pre-filled response files in the TBSM directory:

TBSM Data server response file

```
%IMAGE_HOME%\scripts\sampleResponseFiles\DATA\Win  
$IMAGE_HOME/scripts/sampleResponseFiles/DATA/Linux
```

TBSM Dashboard server response file

```
%IMAGE_HOME%\scripts\sampleResponseFiles\DASH\Win  
$IMAGE_HOME/scripts/sampleResponseFiles/DASH/Linux
```

dbconfig-installer.properties

The sample response files for the database configuration utility are located in the following directory:

%IMAGE_HOME%\scripts\sampleResponseFiles\DBConfig\Win
\$IMAGE_HOME/scripts/sampleResponseFiles/DBConfig/Linux

Dashboard Application Service Hub overview

Web-based products built on the Dashboard Application Service Hub framework share a common user interface where you can launch applications and share information.

Dashboard Application Service Hub helps the interaction and secure passing of data between Tivoli products through a common portal. You can launch from one application to another and within the same dashboard view research different aspects of your managed enterprise.

Dashboard Application Service Hub is installed automatically with the first Tivoli product using the Dashboard Application Service Hub framework. Subsequent products may install updated versions of Dashboard Application Service Hub.

Dashboard Application Service Hub provides the following features:

- A Web based user interface for individual products and for integrating multiple products.
- A single, task-based navigation panel for multiple products. Users select actions based around the task that they want to complete, not by the product that supports that task.
- Single sign-on (SSO), consolidated user management, and a single point of access for different Tivoli applications.
- Aggregated views that span server instances, such as the Tivoli Netcool/OMNIbus ObjectServer.
- Inter-view messaging between products to support contextual linkage between applications.
- The ability to create customized pages and administer access to content by user, role, or group.

Accepting the security certificate

When logging in, you might see a security alert with a message that says there is a problem with the security certificate. This indicates that the browser application is verifying the security certificate of the application server.

Self-signed or CA-signed certificate

The application server uses a self-signed security certificate. You might see a Security Alert when you first connect to the portal that alerts you to a problem with the security certificate. You might be warned of a possible invalid certificate and be recommended to not log in.

Although this warning appears, the certificate is valid and you can accept it. Or, if you prefer, you can install your own CA-signed certificate. For information on creating your own CA-signed certificate, go to: http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tsec_sslcreateCArequest.html

For more information about certificates, go to the IBM WebSphere Application Server Community Edition Documentation Project at <http://publib.boulder.ibm.com/wasce/V2.1.1/en/overview.html>, and search for *Managing trust* and *Managing SSL certificates*.

Logging in

Log in to the portal whenever you want to start a work session.

Before you begin

The Dashboard Application Service Hub Server must be running before you can connect to it from your browser.

About this task

Complete these steps to log in:

Procedure

1. In a Web browser, enter the URL of the Dashboard Application Service Hub Server: `http://host.domain:16310/ibm/console` or `https://host.domain:16311/ibm/console` if it is configured for secure access.
 - `host.domain` is the fully qualified host name or IP address of the Dashboard Application Service Hub Server (such as `MyServer.MySubdomain.MyDomain.com` or `9.51.111.121`, or `localhost` if you are running the Dashboard Application Service Hub Server locally).
 - 16310 is the default nonsecure port number for the portal and 16311 is the default secure port number. If your environment was configured with a port number other than the default, enter that number instead. If you are not sure of the port number, read the application server profile to get the correct number.
 - `ibm/console` is the default path to the Dashboard Application Service Hub Server, however this path is configurable and might differ from the default in your environment.
2. In the login page, enter your user ID and password and click **Log in**.



Attention: After authentication, the web container used by the Dashboard Application Service Hub Server redirects to the last URL requested. This is usually `https://<host>:<port>/ibm/console`, but if you manually change the page URL, after being initially directed to the login page, or if you make a separate request to the server in a discrete browser window before logging in, you may be redirected unexpectedly.

Note: If you have more than one instance of the Dashboard Application Service Hub Server installed on your computer, you should not run more than one instance in a browser session, that is, do not log in to different instances on separate browser tabs.

Results

After your user credentials have been verified, the Welcome page is displayed. If you entered the `localhost` or port number incorrectly, the URL will not resolve. View the application server profile to check the settings for `localhost`, port, and user ID.

What to do next

Select any of the items in the navigation tree to begin working with the console.

While you are logged into the Dashboard Application Service Hub Server, avoid clicking the browser **Back** button because you will be logged out automatically. Click **Forward** and you will see that you are logged out and must resubmit your credentials to log in again.

Note: If you want to use single sign-on (SSO) then you must use the fully qualified domain name of the Dashboard Application Service Hub host.

Port assignments

The application server requires a set of sequentially numbered ports.

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

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 `/opt/IBM/JazzSM/profile/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 `/opt/IBM/JazzSM/profile/logs\AboutThisProfile.txt`:

```
Application server environment to create: Application server
Location: C:\Program Files\IBM\JazzSM\profile
Disk space required: 200 MB
Profile name: DASHProfile
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 `/opt/IBM/JazzSM/var/JazzSMProfile_portDef.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
```

Configuring

Once you have installed Dashboard Application Service Hub, you can configure it to operate in a variety of ways, for example, you can enable load balancing and employ a central user repository.

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 *Dashboard Application Service Hub*.

About this task

To add a new LDAP repository:

Procedure

1. Log in to the *Dashboard Application Service Hub*.
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 Dashboard Application Service Hub Server to communicate with an external LDAP repository.

Configuring an external LDAP repository

You can configure the Dashboard Application Service Hub Server to communicate with an external LDAP repository.

About this task

In a load balanced environment, all Dashboard Application Service Hub 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 *Dashboard Application Service Hub*.
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 Dashboard Application Service Hub Server:
 - a) In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:
 - **Windows** `stopServer.bat server1`
 - **Linux** | **UNIX** `stopServer.sh server1`**Note:** On UNIX and Linux systems, you are prompted to provide an administrator username and password.
 - b) In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:
 - **Windows** `startServer.bat server1`
 - **Linux** | **UNIX** `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 Dashboard Application Service Hub 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 *Dashboard Application Service Hub*.
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 Dashboard Application Service Hub Server.
In a load balanced environment, you must stop and restart each Dashboard Application Service Hub Server instance.
12. Stop and restart the Dashboard Application Service Hub Server:
 - a) In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:
 - **Windows** `stopServer.bat server1`
 - **Linux** **UNIX** `stopServer.sh server1`

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

b) In the /opt/IBM/JazzSM/profile/bin directory, depending on your operating system, enter one of the following commands:

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

Results

You can now manage your LDAP repository users in the portal through the **Settings > 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 **Settings > Manage Users** portlet that have been created in Microsoft Active Directory repositories.

Configuring an SSL connection to an LDAP server

If your implementation of *Dashboard Application Service Hub* 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 Dashboard Application Service Hub Server.

About this task

Follow these instructions to configure the Dashboard Application Service Hub 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 Dashboard Application Service Hub 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 Dashboard Application Service Hub Server.

About this task

Follow these steps to establish a secure channel for communications between the Dashboard Application Service Hub 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 `/opt/IBM/JazzSM/profile/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 Dashboard Application Service Hub Server:
 - a) In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:
 - `Windows` `stopServer.bat server1`
 - `Linux` | `UNIX` `stopServer.sh server1`

Note: On UNIX and Linux systems, you are prompted to provide an administrator username and password.
 - b) In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:
 - `Windows` `startServer.bat server1`
 - `Linux` | `UNIX` `startServer.sh server1`

Configuring VMM for the ObjectServer

When your Tivoli Netcool/OMNIbus ObjectServer is in a federated repository, use the script provided with Dashboard Application Service Hub to configure the Virtual Member Manager adapter for the ObjectServer.

Before you begin

Have the following ObjectServer information at hand: administrator name and password, IP address, and port number. If you have a second ObjectServer for failover support, you need the IP address and port number. The ObjectServer must be running at the time of installing *Dashboard Application Service Hub*, as the installation process attempts to connect to the ObjectServer.

About this task

The script assumes that the DASH installation directory is the parent directory and that the cell name is `JazzSMNode01Cell`. Run the VMM configuration script on every computer where the application server is installed.

Procedure

1. Change to the `install_dir\bin` directory.
The directory contains a script to run:
 - `Windows` `confvmm4ncos.bat`
 - `Linux` `confvmm4ncos.sh`
 - `UNIX` `confvmm4ncos.sh`
2. Enter the following command at the command line: `confvmm4ncos user password address port [address2 port2]where`
 - a) `user` is the ID of a user with administrative privileges for this ObjectServer
 - b) `password` is the password for the user ID
 - c) `address` is the IP address of the ObjectServer
 - d) `port` is the port number used by the ObjectServer

- e) Optional: `address2` and `port2`, if there is a failover server, is the IP address and port number of the failover ObjectServer

Results

The VMM adapter is configured for the ObjectServer. Thereafter, whenever the user registry needs to be accessed, the VMM adapter is called for this information.

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 applications deployed to Dashboard Application Service Hub 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](#).

Configuring single sign-on

Use these instructions to establish single sign-on support and configure a federated repository.

Before you begin

Configuring SSO is a prerequisite to integrating products that are deployed on multiple servers. All Dashboard Application Service Hub Server instances must point to the central user registry (such as a Lightweight Directory Access Protocol server).



Attention: ITM single sign on (SSO) support is only available with ITM Version 6.2 Fix Pack 1 or higher.

About this task

To configure the WebSphere federated repositories functionality for LDAP:

Procedure

1. Log in to the *Dashboard Application Service Hub*.
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. Click **Requires SSL** if all of the requests are expected to use HTTPS.
7. Enter the fully-qualified domain names in the Domain name field where SSO is effective.

If the domain name is not fully qualified, the Dashboard Application Service Hub Server does not set a domain name value for the **LtpaToken** cookie and SSO 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 (use the same domain name).

8. Optional: Enable the **Interoperability Mode** option if you want to support SSO connections in WebSphere Application Server version 5.1.1 or later to interoperate with previous versions of the application server.
9. Optional: Enable the **Web inbound security attribute propagation** option if you want information added during the login at a specific Tivoli Enterprise Portal Server to propagate to other application server instances.
10. After clicking **OK** to save your changes, stop and restart all the Dashboard Application Service Hub Server instances.

What to do next

Note: When you launch *Dashboard Application Service Hub*, you must use a URL in the format `protocol://host.domain:port/*`. If you do not use a fully-qualified domain name, *Dashboard Application Service Hub* cannot use SSO between Tivoli products.

Load balancing

You can setup a load balancing cluster of portal nodes with identical configurations to evenly distribute user sessions.

Load balancing is ideal for *Dashboard Application Service Hub* installations with a large user population. When a node within a cluster fails, new user sessions are directed to other active nodes.

You can create a load balanced cluster from an existing stand-alone application server instance, but must export its data before you configure it for load balancing. The exported data is subsequently imported to one of the nodes in the cluster so that it is replicated across the other nodes in the cluster.

Work load is distributed by session, not by request. If a node in the cluster fails, users who are in session with that node must log back in to access the *Dashboard Application Service Hub*. Any unsaved work is not recovered.

Synchronized data

After load balancing is set up, changes in the console that are stored in global repositories are synchronized to all of the nodes in the cluster using a common database. The following actions cause changes to the global repositories used by the console. Most of these changes are caused by actions in the **Settings** folder in the console navigation.

- Creating, restoring, editing, or deleting a page.
- Creating, restoring, editing, or deleting a view.
- Creating, editing, or deleting a preference profile or deploying preference profiles from the command line.
- Copying a portlet entity or deleting a portlet copy.
- Changing access to a portlet entity, page, external URL, or view.
- Creating, editing, or deleting a role.
- Changes to portlet preferences or defaults.
- Changes from the **Settings** applications, including assigning users and groups to roles.

Note: Global repositories should never be updated manually.

During normal operation within a cluster, updates that require synchronization are first committed to the database. At the same time, the node that submits the update for the global repositories notifies all other nodes in the cluster about the change. As the nodes are notified, they get the updates from the database and commit the change to the local configuration.

If data fails to be committed on any given node, a warning message is logged into the log file. The node is prevented from making its own updates to the database. Restarting the Dashboard Application Service Hub Server instance on the node rectifies most synchronization issues, if not, the node should be removed from the cluster for corrective action. See [“Monitoring a load balancing cluster”](#) on page 135 for more information.

Note: If the database server restarts, all connections from it to the cluster are lost. It may take up to five minutes for connections to be restored, so that users can again perform update operations, for example, modifying or creating views or pages.

Manual synchronization and maintenance mode

Updates to deploy, redeploy, or remove console modules are not automatically synchronized within the cluster. These changes must be performed manually at each node. For deploy and redeploy operations, the console module package must be identical at each node.

When one of the deployment commands is started on the first node, the system enters *maintenance mode* and changes to the global repositories are locked. After you finish the deployment changes on each of the nodes, the system returns to an unlocked state. There is not any restriction to the order that modules are deployed, removed, or redeployed on each of the nodes.

While in maintenance mode, any attempts to make changes in the portal that affect the global repositories are prevented and an error message is returned. The only changes to global repositories that are allowed are changes to a user's personal portlet preferences. Any changes outside the control of the portal, for example, a form submission in a portlet to a remote application, are processed normally.

The following operations are also not synchronized within the cluster and must be performed manually at each node. These updates do not place the cluster in maintenance mode.

- Deploying, redeploying, and removing wires and transformations
- Customization changes to the console user interface (for example, custom images or style sheets) using `consoleProperties.xml`.

To reduce the chance that users could establish sessions with nodes that have different wire and transformation definitions or user interface customizations, schedule these changes to coincide with console module deployments.

Requirements

The following requirements must be met before load balancing can be enabled:

- If you are creating a cluster from a stand-alone instance of Dashboard Application Service Hub, you must export its data before you configure it for load balancing. Once you have configured the cluster, you can import the data to one of the nodes for it to be replicated across the other nodes.
- Lightweight Directory Access Protocol (LDAP) or OMNIbus ObjectServer must be installed and configured as the user repository for each node in the cluster. See [Configuring LDAP user registries](#) for instructions on how to enable LDAP for each node.
- A front-end network dispatcher (for example, IBM HTTP Server) must be setup to handle and distribute all incoming session requests. See [Setting up intermediary services](#) for more information about this task.
- DB2 Version 9.7 must be installed within the network to synchronize the global repositories for the console cluster.
- Each node in the cluster must be enabled to use the same LDAP using the same user and group configuration.
- All console nodes in load balancing cluster must be installed in the same cell name. After console installation on each node, use the `-cellName` parameter on the `manageprofiles` command.
- All console nodes in load balancing cluster must have synchronized clocks.

- The Websphere application server and Dashboard Application Service Hub Server versions must have the same release level, including any fix packs. Fixes and upgrades for the runtime must be applied manually at each node.
- Before joining nodes to a cluster, in each case make sure the node uses the same file-based repository user ID, which has been assigned the role of *iscadmins*.

Exporting data from a stand-alone server to prepare for load balancing

You can export data from an existing stand-alone application server instance to create a data file that can be imported to a load balanced cluster.

About this task

When you are creating a new load balanced cluster from a stand-alone instance, you must first export all data from the stand-alone instance and subsequently import the previously exported data once the cluster is set up.

Note: If you are joining the server to an existing cluster, the other nodes in the cluster should not contain custom data, that is, each node in the cluster should be clean installations. When you import data from the stand-alone server it is replicated across all other nodes.

Procedure

1. At the command line, change to the following directory:

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

2. Run the following command to export the stand-alone server's data:

- **Linux** **UNIX** `restcli.sh export -username tbsmadmin -password tbsmadmin_password -destination data_file`
- **Windows** `restcli.bat export -username tbsmadmin -password tbsmadmin_password -destination data_file`

Where:

tbsmadmin

Specifies the administrator user ID.

tbsmadmin_password

Specifies the password associated with the administrator user ID.

data_file

Specifies the path and file name for the exported data, for example, `c:/tmp/data.zip`.

3. Create a new load balanced cluster using the stand-alone server, or join it to an existing cluster.

4. Import the previously exported data to any node in the cluster.

- a) At the command line, if necessary, change to the following directory:

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

- b) On one of the nodes in the cluster, run the following command to import the stand-alone server's data:

```
restcli.sh import -username tbsmadmin -password tbsmadmin_password -source data_file
```

Where:

tbsmadmin

Specifies the administrator user ID.

tbsmadmin_password

Specifies the password associated with the administrator user ID.

data_file

Specifies the path and file name for the data to be imported, for example, `c:/tmp/data.zip`.

Results

Create a new load balanced cluster using the stand-alone application server, or join it to an existing cluster. Once the cluster is configured, you can import the data file to one of the nodes in the cluster.

What to do next

Setting up a load balancing cluster

You can configure a Dashboard Application Service Hub Server instance to use a database as a file repository instead of a local directory.

Before you begin

If you are creating a cluster from an existing Dashboard Application Service Hub Server instance that contains custom data, ensure that you have exported its data before you begin to configure it for load balancing. Once it is configured, you can import the data to one of the nodes in the new cluster.

Dashboard Application Service Hub is installed on a machine using the cell name designated for all console nodes within the cluster. You have installed and setup a network dispatcher (for example, IBM HTTP Server), DB2, and an LDAP as explained in [“Requirements” on page 120](#).

Procedure

1. On the machine where DB2 is installed, create a DB2 database (see [Creating databases](#)).
2. Check that you have the JDBC driver for DB2 on the computer where Dashboard Application Service Hub is installed. The JDBC driver should be available at: `/opt/IBM/WebSphere/AppServer/universalDriver/lib`.
3. Configure load balancing, see https://www.ibm.com/support/knowledgecenter/SSSHTQ_8.1.0/com.ibm.netcool_OMNIBus.doc_8.1.0/webtop/wip/task/web_con_lb_configure.html.
4. In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:

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

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

5. Make sure your database is empty and the server is not started.
Problems may occur if you try to setup load balancing on a non-empty database or active server.
6. From a command prompt, change to the `/opt/IBM/WebSphere/AppServer/bin` directory and issue this command:

- **Windows** `..\ws_ant.bat -f install.ant configHA -Dusername=DB2_username -Dpassword=DB2_password`
- **Linux** **UNIX** `../ws_ant.sh -f install.ant configHA -Dusername=DB2_username -Dpassword=DB2_password`

7. In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:

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

Results

The load balancing cluster is created and the console node is joined to the cluster as the first node.

What to do next

Add (or join) additional nodes to the cluster.

Joining a node to a load balancing cluster

You can configure a Dashboard Application Service Hub Server to join an existing load balancing cluster.

Before you begin

1. If you are joining a stand-alone Dashboard Application Service Hub Server instance to a cluster, ensure that you first export all of its data. Once you have joined it to the cluster, you can then import the previously exported data. Other nodes in the cluster should not contain any custom data and should effectively be new installed instances.
2. Make sure you have successfully enabled load balancing following the steps in [“Setting up a load balancing cluster”](#) on page 122.
3. Dashboard Application Service Hub should be installed to the node using the same cell name that is designated for the cluster.
4. All console modules deployed to the cluster must be already deployed to the node that you intend to join.
5. You should deploy any wires or transformations used by the nodes in the cluster.
6. If the cluster is using any customization changes in `consoleProperties.xml` you must copy these changes and this file to the same location on the node that you intend to join.
7. The node must be configured to the same LDAP with the same user and group definitions as all other nodes in the cluster.

About this task

The following parameters are used on the `join` option when a node is added:

- **-Dusername** - specify the DB2 administrator's username
- **-Dpassword** - specify the DB2 administrator's password

Procedure

1. Check that you have the JDBC driver for DB2 on the computer where Dashboard Application Service Hub is installed. The JDBC driver should be available at: `/opt/IBM/WebSphere/AppServer/universalDriver/lib`.
2. Configure load balancing, see https://www.ibm.com/support/knowledgecenter/SSSHTQ_8.1.0/com.ibm.netcool_OMNibus.doc_8.1.0/webtop/wip/task/web_con_lb_configure.html.
3. In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:

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

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

4. Make sure the Dashboard Application Service Hub Server is not started.
5. At a command prompt, change to the `/opt/IBM/WebSphere/AppServer/bin` directory and issue this command
 - **Windows** `..\ws_ant.bat -f install.ant configHA -Dusername=DB2_username -Dpassword=DB2_password`
 - **Linux** **UNIX** `../ws_ant.sh -f install.ant configHA -Dusername=DB2_username -Dpassword=DB2_password`

6. In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:

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

Results

The console node is joined to the cluster.

What to do next

Add another node to the cluster, or if you have completed adding nodes, enable server to server trust for each node to every other node in the cluster.

Depending on the network dispatcher (for example, IBM HTTP Server) that you use, you might have further updates to get session requests routed to the new node. Refer to the documentation applicable to your network dispatcher for more information.

Enabling server-to-server trust

Use this procedure to enable load balanced nodes to connect to each other and send notifications.

About this task

These steps are required to enable load balancing between the participating nodes. Complete these steps on each node.

Procedure

1. In a text editor, open the `ssl.client.props` file from the `/opt/IBM//WebSphere/AppServer/profileTemplates/management/documents/properties/` directory.
2. Uncomment the section that starts with **`com.ibm.ssl.alias=AnotherSSLSettings`** so that it looks like this:

```
com.ibm.ssl.alias=AnotherSSLSettings
com.ibm.ssl.protocol=SSL_TLS
com.ibm.ssl.securityLevel=HIGH
com.ibm.ssl.trustManager=IbmX509
com.ibm.ssl.keyManager=IbmX509
com.ibm.ssl.contextProvider=IBMJSSE2
com.ibm.ssl.enableSignerExchangePrompt=true
#com.ibm.ssl.keyStoreClientAlias=default
#com.ibm.ssl.customTrustManagers=
#com.ibm.ssl.customKeyManager=
#com.ibm.ssl.dynamicSelectionInfo=
#com.ibm.ssl.enabledCipherSuites=
```

3. Uncomment the section that starts with **`com.ibm.ssl.trustStoreName=AnotherTrustStore`** so that it looks like this:

```
# TrustStore information
com.ibm.ssl.trustStoreName=AnotherTrustStore
com.ibm.ssl.trustStore=${user.root}/config/cells/JazzSMNode01Cell/nodes/
JazzSMNode01/trust.p12
com.ibm.ssl.trustStorePassword={xor}CDo9Hgw=
com.ibm.ssl.trustStoreType=PKCS12
com.ibm.ssl.trustStoreProvider=IBMJCE
com.ibm.ssl.trustStoreFileBased=true
com.ibm.ssl.trustStoreReadOnly=false
```

4. Update the location of the trust store that the signer should be added to in the `com.ibm.ssl.trustStore` property of `AnotherTrustStore` by replacing the default value **`com.ibm.ssl.trustStore=${user.root}/etc/trust.p12`** with the correct path for your trust store. Example:

```
com.ibm.ssl.trustStore=${user.root}/config/cells/JazzSMNode01Cell/nodes/  
JazzSMNode01/trust.p12
```

After the update, the section must look like this:

```
com.ibm.ssl.trustStoreName=AnotherTrustStore  
com.ibm.ssl.trustStore=${user.root}/config/cells/TIPCell/nodes/TIPNode/trust.p12  
com.ibm.ssl.trustStorePassword={xor}CDo9Hgw=  
com.ibm.ssl.trustStoreType=PKCS12  
com.ibm.ssl.trustStoreProvider=IBMJCE  
com.ibm.ssl.trustStoreFileBased=true
```

5. Save your changes to `ssl.client.props`.

6. Stop and restart the Dashboard Application Service Hub Server:

a) In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:

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

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

b) In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:

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

7. Complete all of the steps so far on each node before you continue with the rest of the steps.

8. Run the following command on each node for each *myremotehost* (that is, for every node that you want to enable trust with) in the cluster:

```
Windows C:\Program Files\IBM\JazzSM\profile\bin\retrieveSigners.bat  
NodeDefaultTrustStore AnotherTrustStore -host myremotehost -port  
remote_SOAP_port
```

```
Linux | UNIX /opt/IBM/JazzSM/profile/bin/retrieveSigners.sh  
NodeDefaultTrustStore AnotherTrustStore -host myremotehost -port  
remote_SOAP_port
```

where *myremotehost* is the name of the computer to enable trust with; *remote_SOAP_port* is the SOAP connector port number (16313 is the default). If you have installed with non-default ports, check `/opt/IBM/var/JazzSMPprofile_portDef.properties` for the value of `SOAP_CONNECTOR_ADDRESS` and use that.

9. Stop and restart the Dashboard Application Service Hub Server:

a) In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:

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

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

b) In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:

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

Example

In this example, the load balancing cluster is comprised of two Microsoft Windows nodes named *myserver1* and *myserver2*. The command entered on *myserver1*:

```
retrieveSigners.bat NodeDefaultTrustStore AnotherTrustStore -host myserver2  
-port 16313
```

The command entered on *myserver2*:

```
retrieveSigners.bat NodeDefaultTrustStore AnotherTrustStore -host myserver1  
-port 16313
```

Verifying a load balancing implementation

Use the information in this topic to verify that your Dashboard Application Service Hub load balancing setup is working correctly once you have added all nodes to the cluster and enabled server-to-server trust.

About this task

This task allows you to confirm the following functions are working correctly:

- The database used for your load balancing cluster is properly created and initialized.
- Every node in the cluster uses the database as its repository instead of its own local file system.
- Server-to-server trust is properly enabled between nodes in the cluster.

To verify your load balancing configuration:

Procedure

1. Ensure that each Dashboard Application Service Hub Server instance on every node in the cluster is running.
2. In a browser, log into one node, create a new View and save your changes.
3. Log into the remaining nodes and verify that the newly created view is available in each one.

Preparing the HTTP server for load balancing

Install the IBM HTTP Server and configure the Web server plug-in for passing requests to the Dashboard Application Service Hub Server that are part of the load balancing configuration.

Before you begin

The IBM HTTP Server uses a Web server plug-in to forward HTTP requests to the Dashboard Application Service Hub Server. You can configure the HTTP server and the Web server plug-in to act as the load balancing server, that is, pass requests (HTTP or HTTPS) to one of any number of nodes. The load balancing methods supported by the plug-in are *round robin* and *random*:

- With a round robin configuration, when a browser connects to the HTTP server, it is directed to one of the configured nodes. When another browser connects, it is directed to a different node.
- With the random setting, each browser is connected randomly to a node. Once a connection is established between a browser and a particular node, that connection remains until the user logs out or the browser is closed.

The HTTP server is necessary for directing traffic from browsers to the applications that run in the *Dashboard Application Service Hub* environment. The server is installed between the portal and the Dashboard Application Service Hub Server, and is outside the firewall.

The Web server plug-in uses the `plugin-cfg.xml` configuration file to determine whether a request is for the application server.

About this task

Complete this procedure to configure the Web server plug-in for load balancing for each node.

Procedure

1. If you do not already have the IBM HTTP Server installed, install it before proceeding. It should be installed where it can be accessed from the Internet or Intranet (or both). Select the link at the end of this topic for the installation procedure.
2. Install IBM HTTP Server ensuring that you include the IBM HTTP Server Plug-in for IBM WebSphere Application Server option.
For more information, see http://publib.boulder.ibm.com/infocenter/wasinfo/fep/topic/com.ibm.websphere.ihs.doc/info/ihs/ihs/tihs_installihs.html.
3. Create a new CMS-type key database.
For more information see http://publib.boulder.ibm.com/infocenter/wasinfo/fep/index.jsp?topic=/com.ibm.websphere.ihs.doc/info/ihs/ihs/tihs_createkeydb.html.
4. Create a self-signed certificate to allow SSL connections between nodes.
For more information, see http://publib.boulder.ibm.com/infocenter/wasinfo/fep/index.jsp?topic=/com.ibm.websphere.ihs.doc/info/ihs/ihs/tihs_certselfsigned.html.
5. To enable SSL communications for the IBM HTTP Server, in a text editor, open *HTTP_server_install_dir/conf/httpd.conf*. Locate the line # End of example SSL configuration and add the following lines, ensuring that the KeyFile line references the key database file created in step "3" on page 182 and save your changes.

```
LoadModule ibm_ssl_module modules/mod_ibm_ssl.so
<IfModule mod_ibm_ssl.c>
  Listen 443
  <VirtualHost *:443>
    SSLEnable
  </VirtualHost>
</IfModule>
SSLDisable
KeyFile "C:/Program Files/IBM/HTTPServer/bin/test.kdb"
```

For more information, refer to the first example at http://publib.boulder.ibm.com/infocenter/wasinfo/fep/index.jsp?topic=/com.ibm.websphere.ihs.doc/info/ihs/ihs/tihs_setupssl.html.

6. Restart the IBM HTTP Server.
For more information, see http://publib.boulder.ibm.com/infocenter/wasinfo/fep/topic/com.ibm.websphere.ihs.doc/info/ihs/ihs/tihs_startihs.html.
7. On the IBM HTTP Server computer, to verify that SSL is enabled ensure that you can access `https://localhost`.
8. Stop and restart the Dashboard Application Service Hub Server:
 - a) In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:
 - **Windows** `stopServer.bat server1`
 - **Linux** | **UNIX** `stopServer.sh server1`**Note:** On UNIX and Linux systems, you are prompted to provide an administrator username and password.
 - b) In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:
 - **Windows** `startServer.bat server1`
 - **Linux** | **UNIX** `startServer.sh server1`
9. Start the HTTP server:
 - a) Change to the directory where it is installed.
 - b) Run this command: `bin/apachectl start`
Note you must restart the server after changing the `plugin-cfg.xml` file.

What to do next

Enter the URL for the HTTP Server in a browser `http://HTTP_server_host/HTTP_server_port` and it will be forwarded to one of the nodes.

Note: The default load balancing method is random, whereby each browser is connected randomly to a node.

Setting clone IDs for nodes

Assign a clone ID for all nodes in the cluster.

About this task

Complete this procedure to set clone IDs for all nodes in the cluster. You must carry out these steps on each node.

Procedure

1. In a text editor, open the `server.xml` file from the `./JazzSM/profile/installedApps/JazzSMNode01Cell/isc.ear/sla.war/otherfiles/` directory
2. In `server.xml`, locate the entry `<components xmi:type="applicationserver.webcontainer:WebContainer.`
3. Within the `components` element, add the following entry:

```
<properties xmi:id="WebContainer_1183077764084" name="HttpSessionCloneId" value="12345" required="false"/>
```

Where:

`value` is the clone ID for the node, for example, `value="12345"`. The clone ID must be unique to each node. An example of an updated `components` element is provided here:

```
<components xmi:type="applicationserver.webcontainer:WebContainer" xmi:id="WebContainer_1183077764084" enableServletCaching="false" disablePooling="false">
  <stateManagement xmi:id="StateManageable_1183077764087" initialState="START"/>
  <services xmi:type="applicationserver.webcontainer:SessionManager" xmi:id="SessionManager_1183077764084" enable="true" enableUrlRewriting="false" enableCookies="true" enableSSLTracking="false" enableProtocolSwitchRewriting="false" sessionPersistenceMode="NONE" enableSecurityIntegration="false" allowSerializedSessionAccess="false" maxWaitTime="5" accessSessionOnTimeout="true">
    <defaultCookieSettings xmi:id="Cookie_1183077764084" domain="" maximumAge="-1" secure="false"/>
    <sessionDatabasePersistence xmi:id="SessionDatabasePersistence_1183077764084" datasourceJNDIName="jdbc/Sessions" userId="db2admin" password="{xor}0z1tPjsyNjE=" db2RowSize="ROW_SIZE_4KB" tableSpaceName=""/>
    <tuningParams xmi:id="TuningParams_1183077764084" usingMultiRowSchema="false" maxInMemorySessionCount="1000" allowOverflow="true" scheduleInvalidation="false" writeFrequency="TIME_BASED_WRITE" writeInterval="10" writeContents="ONLY_UPDATED_ATTRIBUTES" invalidationTimeout="30">
      <invalidationSchedule xmi:id="InvalidationSchedule_1183077764084" firstHour="14" secondHour="2"/>
    </tuningParams>
  </services>
  <properties xmi:id="WebContainer_1183077764084" name="HttpSessionCloneId" value="12345" required="false"/>
</components>
```

4. Save the changes you made to `server.xml`.

Generating the plugin-cfg.xml file

Run GenPluginCfg.bat to generate the plugin-cfg.xml file and save it in /opt/IBM/JazzSM/profile/config/cells/JazzSMNode01Cell.

About this task

Complete this procedure to generate the plugin-cfg.xml file. You must carry out these steps on each node.

Procedure

1. On a node, change to /opt/IBM/JazzSM/profile/bin and run the following command:

- **Windows** GenPluginCfg.bat
- **Linux** **UNIX** GenPluginCfg.sh

This command generates a file called plugin-cfg.xml and saves it to the /opt/IBM/JazzSM/profile/config/cells/JazzSMNode01Cell directory.

2. On the IBM HTTP Server, in the following directory, replace the existing plugin-cfg.xml with the version generated in step “1” on page 184:

`HTTP_web_server_install_dir/plugins/config/webserver1`

The following steps establish the new /ibm/* URI (Uniform Resource Identifier), which is where the plug-in will redirect requests:

- a) On the IBM HTTP Server, change to the directory where the Web server definition file is (such as `cd plugins/config/webserver1`).
- b) Open the plugin-cfg.xml file in a text editor, and in reference to the sample content extract provided below, edit the file to provide details of your IBM HTTP Server and all Dashboard Application Service Hub Server instances.

HTTP SERVER PATH is the path to where the HTTP server is installed.

HTTP SERVER PORT is the port for the HTTP server.

SERVER1 is the fully qualified name of the computer where the application server is installed and started.

SERVER2 is the fully qualified name of the computer where another application server is installed and started.

CLONE_ID is the is the unique clone ID assigned to a particular node (server) in the cluster.

- c) In the ServerCluster section, the values for the keyring and stashfile properties should be **HTTP SERVER PATH** /plug-ins/etc/plug-in-key.kdb and **HTTP SERVER PATH** /plug-ins/etc/plug-in-key.sth respectively.
- d) Continue to add Server entries for any other nodes, following the same pattern. Add a new entry under PrimaryServers for each additional server.
- e) Add CloneID and LoadBalanceWeight attributes for every Server entry.

Important: For more information on web server plug-in workload management policies and to help you determine the appropriate values for the elements LoadBalance and LoadBalanceWeight, refer to the following articles:

- <http://www.redbooks.ibm.com/abstracts/TIPS0235.html>
- <http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg21219567>



Attention: The HTTP and HTTPS port values for all nodes should be the same.

```
<Config ASDisableNagle="false" IISDisableNagle="false"
IgnoreDNSFailures="false" RefreshInterval="60"
ResponseChunkSize="64" AcceptAllContent="false"
IISPluginPriority="High" FIPSEnable="false"
```

```

AppServerPortPreference="HostHeader" VHostMatchingCompat="false"
ChunkedResponse="false">
  <Log LogLevel="Trace" Name="HTTP_SERVER_PATH/Plugins/logs/webserver1/
http_plugin.log"/>
  <Property Name="ESIEnable" Value="true" />
  <Property Name="ESIMaxCacheSize" Value="1024" />
  <Property Name="ESIInvalidationMonitor" Value="false" />
  <Property Name="ESIEnableToPassCookies" Value="false" />
  <Property Name="PluginInstallRoot" Value="HTTP_SERVER_PATH/Plugins" />
  <VirtualHostGroup Name="default_host">
    <VirtualHost Name="*:16310" />
    <VirtualHost Name="*:80" />
    <VirtualHost Name="*:16311" />
    <VirtualHost Name="*:5060" />
    <VirtualHost Name="*:5061" />
    <VirtualHost Name="*:443" />
  <VirtualHost Name="*:HTTP_SERVER_PORT"/>
  </VirtualHostGroup>
  <ServerCluster CloneSeparatorChange="false" GetDWLMTable="false"
IgnoreAffinityRequests="true" LoadBalance="Round Robin"
Name="server1_Cluster" PostBufferSize="64" PostSizeLimit="-1"
RemoveSpecialHeaders="true" RetryInterval="60">
  <Server Name="TIPNode1_server1"
ConnectTimeout="0" CloneID="CLONE_ID" ExtendedHandshake="false"
ServerIOTimeout="0" LoadBalanceWeight="100" MaxConnections="-1"
WaitForContinue="false">
    <Transport Hostname="SERVER1" Port="16310"
Protocol="http"/>
    <Transport Hostname="SERVER1" Port="16311"
Protocol="https">
      <Property name="keyring" value="HTTP_SERVER_PATH\Plugins\config
\webserver1\plugin-key.kdb"/>
      <Property name="stashfile" value="HTTP_SERVER_PATH\Plugins\config
\webserver1\plugin-key.sth"/>
    </Transport>
  </Server>
  <Server Name="TIPNode1_server2"
ConnectTimeout="0" CloneID="CLONE_ID" ExtendedHandshake="false"
ServerIOTimeout="0" LoadBalanceWeight="100" MaxConnections="-1"
WaitForContinue="false">
    <Transport Hostname="SERVER2" Port="16310"
Protocol="http"/>
    <Transport Hostname="SERVER2" Port="16311"
Protocol="https">
      <Property name="keyring" value="HTTP_SERVER_PATH\Plugins\config
\webserver1\plugin-key.kdb"/>
      <Property name="stashfile" value="HTTP_SERVER_PATH\Plugins\config
\webserver1\plugin-key.sth"/>
    </Transport>
  </Server>
  <PrimaryServers>
    <Server Name="TIPNode1_server1" />
    <Server Name="TIPNode1_server2" />
  </PrimaryServers>
</ServerCluster>
  <UriGroup Name="server1_Cluster_URIs">
    <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionId"
Name="/ivt/*" />
    <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionId"
Name="/IBM_WS_SYS_RESPONSESERVLET/*" />
    <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionId"
Name="/IBM_WS_SYS_RESPONSESERVLET/*.jsp" />
    <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionId"
Name="/IBM_WS_SYS_RESPONSESERVLET/*.jsw" />
    <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionId"
Name="/IBM_WS_SYS_RESPONSESERVLET/*.jsw" />
    <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionId"
Name="/IBM_WS_SYS_RESPONSESERVLET/j_security_check" />
    <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionId"
Name="/IBM_WS_SYS_RESPONSESERVLET/ibm_security_logout" />
    <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/ibm/console/*" />
    <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/ibm/help/*" />
    <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/ibm/action/*" />
    <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/ISCWire/*" />
    <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/isc/*" />
    <Uri AffinityCookie="JSESSIONID_ibm_console_16310"

```

```

AffinityURLIdentifier="jsessionId" Name="/ISCHA/*" />
  <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/tip_ISCAAdminPortlet/*" />
  <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/ISCAAdminPortlets/*" />
  <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/mum/*" />
  <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/ibm/TIPChangePasswd/*" />
  <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/ibm/TIPEXportImport/*" />
  <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/ibm/tivoli/*" />
  <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/proxy/*" />
  <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/TIPWebWidget/*" />
  <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/ibm/dbfile/*" />
  <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/ibm/TIPChartPortlet/*" />
  <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/TIPUtilPortlets/*" />
  <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/WIMPortlet/*" />
  <Uri AffinityCookie="JSESSIONID_ibm_console_16310"
AffinityURLIdentifier="jsessionId" Name="/SysMgmtCommonTaskGroups/*" />
</UriGroup>
<Route ServerCluster="server1_Cluster" UriGroup="server1_Cluster_URIs"
VirtualHostGroup="default_host" />
<RequestMetrics armEnabled="false" newBehavior="false" rmEnabled="false"
traceLevel="HOPS">
  <filters enable="false" type="URI">
    <filterValues enable="false" value="/snoop" />
    <filterValues enable="false" value="/hitcount" />
  </filters>
  <filters enable="false" type="SOURCE_IP">
    <filterValues enable="false" value="255.255.255.255" />
    <filterValues enable="false" value="254.254.254.254" />
  </filters>
  <filters enable="false" type="JMS">
    <filterValues enable="false" value="destination=aaa" />
  </filters>
  <filters enable="false" type="WEB_SERVICES">
    <filterValues enable="false" value="wsdlPort=aaa:op=bbb:nameSpace=ccc" />
  </filters>
</RequestMetrics>
</Config>

```

Configuring SSL from each node to the IBM HTTP Server

For load balancing implementations, you must configure SSL between the IBM HTTP Server plug-in and each node in the cluster.

Before you begin

This task assumes that you have already installed and configured the IBM HTTP Server for load balancing.

About this task

For each node in the cluster, follow these instructions to configure the node to communicate over a secure (SSL) channel with the IBM HTTP Server.

Procedure

1. Log in to the *Dashboard Application Service Hub*.
2. In the navigation pane, click **Settings** > **Websphere Administrative Console** and click **Launch Websphere administrative console**.
3. Follow these steps to extract signer certificate from the trust store:
 - a) In the WebSphere Application Server administrative console navigation pane, click **Security** > **SSL certificate and key management**.

- b) In the Related Items area, click the **Key stores and certificates** link and in the table click the **NodeDefaultTrustStore** link.
 - c) In the Additional Properties area, click the **Signer certificates** link and in the table that is displayed, select the `root` entry check box.
 - d) Click **Extract** and in the page that is displayed, in the **File name** field, enter a certificate file name (*certificate.arm*), for example, `c:\tivpc064ha1.arm`.
 - e) From the **Data Type** list select the **Base64-encoded ASCII data** option and click **OK**.
 - f) Locate the extracted signer certificate and copy it to the computer running the IBM HTTP Server.

Note: These steps are particular to Dashboard Application Service Hub, for general WebSphere Application Server details and further information, see: http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.base.doc/info/aes/ae/tsec_sslextractsigncert.html
4. On the computer running the IBM HTTP Server, follow these steps to import the extracted signer certificate into the key database:
 - a) Start the key management utility (iKeyman), if it is not already running, from `HTTP_SERVER_PATH/bin`:
 - **Linux** | **UNIX** At the command line, enter `./ikeyman.sh`
 - **Windows** At the command line, enter `ikeyman.exe`
 - b) Open the CMS key database file that is specified in `plugin-cfg.xml`, for example, `HTTP_SERVER_PATH/plugin-ins/etc/plugin-in-key.kdb`.
 - c) Provide the password (default is WebAS) for the key database and click **OK**.
 - d) From the **Key database content**, select **Signer Certificates**.
 - e) Click **Add** and select the signer certificate that you copied from the node to the computer running the IBM HTTP Server and click **OK**.
 - f) Select the **Stash password to a file** check box and click **OK** to save the key database file.

Note: For more information on certificates in WebSphere Application Server, see: http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.ihs.doc/info/ihs/ihs/tihs_ikeyscca.html
 5. Repeat these steps for each node in the cluster.
 6. For the changes to take effect, stop and restart all nodes in the cluster and also restart the computer running the IBM HTTP Server.
 - a) In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:
 - **Windows** `stopServer.bat server1`
 - **Linux** | **UNIX** `stopServer.sh server1`

Note: On UNIX and Linux systems, you are prompted to provide an administrator username and password.
 - b) In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:
 - **Windows** `startServer.bat server1`
 - **Linux** | **UNIX** `startServer.sh server1`
 - c) Restart the IBM HTTP Server.

For more information, see http://publib.boulder.ibm.com/infocenter/wasinfo/fep/topic/com.ibm.websphere.ihs.doc/info/ihs/ihs/tihs_startihs.html.

What to do next

You should now be able to access the load balanced cluster through `https://http_server_hostname/ibm/console` (assuming that the default context root (`/ibm/console`) was defined in at the time of installation).

Importing stand-alone instance data to a cluster

If you created a cluster from a stand-alone application server instance, you can then import the data that you exported prior to configuring the stand-alone instance as a cluster node.

About this task

Import the previously exported data file to any node in the cluster.

Important: The instructions in this topic apply only to importing data that was exported when preparing to create a load balanced cluster from a stand-alone application server instance, as described in [“Exporting data from a stand-alone server to prepare for load balancing” on page 121.](#)

Procedure

1. At the command line, change to the following directory:

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

2. On one of the nodes in the cluster (most likely the node that was previously set up as a stand-alone server instance), run the following command to import the data file:

- **Linux** **UNIX** `restcli.sh import -username tbsmadmin -password tbsmadmin_password -source data_file`
- **Windows** `restcli.bat import -username tbsmadmin -password tbsmadmin_password -source data_file`

Where:

tbsmadmin

Specifies the administrator user ID.

tbsmadmin_password

Specifies the password associated with the administrator user ID.

data_file

Specifies the path and file name to the data file that is to be imported, for example, `c:/tmp/data.zip`.

Results

The data from the initial application server is imported to the node and replicated across the other cluster nodes.

Removing a node

Follow these steps to remove a node from the load balancing cluster.

About this task

The following parameters are used on the `disjoin` option when a node is removed.

- **-Dusername** - specify the DB2 administrator's username
- **-Dpassword** - specify the DB2 administrator's password

Procedure

1. From a command prompt, change to the `/opt/IBM/WebSphere/AppServer/bin` directory and issue this command:

- **Windows** `..\ws_ant.bat -f uninstall.ant disjoin -Dusername=DB2_username -Dpassword=DB2password`
 - **Linux** **UNIX** `../ws_ant.sh -f uninstall.ant disjoin -Dusername=DB2_username -Dpassword=DB2password`
2. Update the network dispatcher (for example, IBM HTTP Server) to remove the node from the configuration.

Removing a remote node

About this task

This command should be used only in the rare occasions where physical access to the node is not available or a serious hardware or software failure has occurred. If the node is remotely disjoined but continues to function, some problems with synchronization might arise that can lead to problems with data consistency and synchronization.

Procedure

1. From a command prompt, change to the `/opt/IBM/WebSphere/AppServer/bin` directory and issue this command:
 - **Windows** `..\ws_ant.bat -f uninstall.ant remote-disjoin -DremoteHost=remote_host -DremotePort=9044 -Dusername=DB2_username -Dpassword=DB2_password`
 - **Linux** **UNIX** `../ws_ant.sh -f uninstall.ant remote-disjoin -DremoteHost=remote_host -DremotePort=9044 -Dusername=DB2_username -Dpassword=DB2_password`
2. Update the network dispatcher (for example, IBM HTTP Server) to remove the node from the configuration.

Removing a load balancing cluster

Follow these steps to remove the last node from a cluster and thereby the cluster itself.

Before you begin

Make sure you have removed all other nodes from the cluster. This command should be issued from the last active node remaining in the cluster.

About this task

The following parameters are used on the `uninstall` option when the node is removed.

- **-Dusername** - specify the DB2 administrator's username
- **-Dpassword** - specify the DB2 administrator's password

Procedure

From a command prompt, change to the `/opt/IBM/WebSphere/AppServer/bin` directory and issue this command:

- **Windows** `..\ws_ant.bat -f uninstall.ant uninstall -Dusername=DB2_username -Dpassword=DB2_password`
- **Linux** **UNIX** `../ws_ant.sh -f uninstall.ant uninstall -Dusername=DB2_username -Dpassword=DB2_password`

Monitoring a load balancing cluster

If synchronized data fails to be committed to a node in the cluster, that node should be removed from the cluster for corrective action. Use the diagnosis tool to identify any unsynchronized nodes in the load balancing cluster.

To determine if changes to global data are not committed to any of the nodes, use the **HATool** command script to check the synchronization of modules and repositories on the nodes in a cluster. For the HATool, you must provide the DB2 administrator's credentials.

Query synchronization of modules

Use this command to determine if all nodes have identical sets of modules deployed.

```
HATool.bat/sh modules username password -byNodes -showAll
```

The following parameters are optional.

- **-byNodes**

Specifies that the results of the command are ordered by the node in the cluster. This parameter is optional. The default is to list the results by module.

- **-showAll**

Specifies that all modules and nodes in the cluster should be returned. This parameter is optional. The default is to return only modules for unsynchronized nodes.

Query the synchronization of global repositories

Use this command to determine if all repositories are synchronized on all nodes.

```
HATool.bat/sh repositories username password -byNodes -showAll
```

The following parameters are optional.

- **-byNodes**

Specifies that the results of the command are ordered by the node in the cluster. This parameter is optional. The default is to list the results by repository.

- **-showAll**

Specifies that all repositories and nodes in the cluster should be returned. This parameter is optional. The default is to return only repositories for unsynchronized nodes.

Release the global lock

Use this command to manually release the global lock placed on all of the console nodes when the cluster is in maintenance mode. This command is used when a node cannot commit a change during synchronization and has to be taken offline.

```
HATool.bat/sh release-lock username password
```

Configuring Tivoli Access Manager in Dashboard Application Service Hub

You can configure Dashboard Application Service Hub to use Tivoli Access Manager WebSEAL Version 6.1 to manage authentication.

You must install and configure Tivoli Access Manager WebSEAL Version 6.1. To set up and configure Tivoli Access Manager WebSEAL, see http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itame.doc/am611_install196.htm#webseal.

For more information on administering Tivoli Access Manager WebSEAL, see http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itame.doc/am611_webseal_admin.htm.



Attention: The IBM Tivoli Netcool Impact user interface contained within IBM Tivoli Business Service Manager may not function within the Tivoli Access Manager WebSEAL environment. If you need to access the Netcool/ Impact user interface (for example, to edit Impact policies or access Operator Views), you must do so outside of the Tivoli Access Manager WebSEAL environment.

Configuring single sign-on using ETai

In a WebSphere Application Server (WAS) environment, Tivoli Access Manager WebSEAL can be used as a reverse proxy to intercept incoming http or https requests to ensure that users are authenticated and authorized and are passed to the relevant Dashboard Application Service Hub Server .

ETai is the component that implements the WebSphere Application Server trust association interceptor interface to achieve single sign on from WebSEAL to the Dashboard Application Service Hub Server.

Dashboard Application Service Hub supports single sign-on (SSO) with perimeter authentication services such as reverse proxies through trust associations. When trust associations are enabled, the WebSphere Application Server is not required to authenticate a user if a request arrives from a trusted source that has already performed authentication.

Once a trust association is configured between WebSEAL and the Dashboard Application Service Hub Server, a user can login into Tivoli Access Manager and then access the Dashboard Application Service Hub Server without having to re-authenticate. The ETai must be configured in Dashboard Application Service Hub Server server and is responsible for establishing trust against the WebSEAL server. ETai simplifies the use of Tivoli Access Manager and the configuration required to achieve SSO. One advantage is that Tivoli Access Manager and Dashboard Application Service Hub can use different user registries and still be able to perform SSO. It also provides the mapping between different registry formats.

Installing ETai

Use these instructions, to install the Tivoli Access Manager Extended Trust Association Interceptor in a Dashboard Application Service Hub environment.

Before you begin

Source a copy of `com.ibm.sec.authn.tai.etai_6.0.jar` from your installation media.

About this task

To install ETai:

Procedure

1. Copy `com.ibm.sec.authn.tai.etai_6.0.jar` to the `plugins` directory.
2. At the command line, depending on your operating system, run the relevant command:
 - **Windows** `C:\Program Files\IBM\JazzSM\profile\bin\Osgicfginit.bat`
 - **Linux** **UNIX** `/opt/IBM/JazzSM/profile/bin/Osgicfginit.sh`
3. Copy `pd.jar` to `/opt/IBM/WebSphere/AppServer/java/jre/lib/ext`

What to do next

Configure ETai in a Dashboard Application Service Hub environment.

Enabling a trust association for ETai

You must enable a trust association between the Tivoli Access Manager Extended Trust Association Interceptor in the Dashboard Application Service Hub environment.

About this task

To configure a trust association for ETai:

Procedure

1. Log in to the portal and click **Settings > WebSphere Administrative Console**.
2. In the **WebSphere Administrative Console** page, click **Launch WebSphere administrative console**.
3. In the **WebSphere Administrative Console** navigation pane, click **Global security**.

4. In the **Global security** page, expand **Web security** and click **Trust association**.
5. In the General Properties area, click the **Enable trust association** option if it is disabled and click **Apply**.
Your update is saved and you are returned to the **Global security** page.
6. In the **Global security** page, expand **Web security** and click **Trust association** to display the **Trust association** page.
7. In the Additional properties area, click the **Interceptors** link to display the **Interceptors** page.
8. If `com.ibm.sec.authn.tai.TAMETai` is not listed on the page, click **New**.
9. In the **Interceptor class name** field enter the string `com.ibm.sec.authn.tai.TAMETai` and click **Apply**.
10. In the Messages area, click the **Save** link to commit your change.

What to do next

Configure ETai in the a Dashboard Application Service Hub environment.

Configuring custom properties for ETai

Once you have enabled a trust association for the Tivoli Access Manager Extended Trust Association Interceptor in the Dashboard Application Service Hub environment, you must configure its custom properties.

About this task

To configure custom properties for the ETai:

Procedure

1. Log in to the portal and click **Settings > WebSphere Administrative Console**.
2. In the **WebSphere Administrative Console** page, click **Launch WebSphere administrative console**.
3. In the **WebSphere Administrative Console** navigation pane, click **Global security**.
4. In the **Global security** page, expand **Web security** and click **Trust association** to display the **Trust association** page.
5. In the Additional properties area, click the **Interceptors** link to display the **Interceptors** page.
6. From the list of interceptor classes, select the `com.ibm.sec.authn.tai.TAMETai` entry.
7. In the Additional properties area, click the **Custom properties** link to display the **Custom properties** page.
8. Review the details for the custom properties listed in the following table:

<i>Table 13. ETai custom properties</i>	
Property details	Notes
Property name: <code>com.ibm.websphere</code> <code>.security.webseal</code> <code>.useWebSphereUserRegistry</code> Type: string Required: Yes Values: true or false Default value: true	ETai authenticates the trusted user against the WebSphere Application Server user registry or the Tivoli Access Manager Authorization Server. If this property is set to true, the resulting Subject will not contain a PDPrincipal as the Tivoli Access Manager Authorization Server is required to build the PDPrincipal. Any other value for this property will result in a PDPrincipal being added to the Subject.

Table 13. ETai custom properties (continued)

Property details	Notes
<p>Property name: com.ibm.websphere .security.webseal .tamUserDnMapping</p> <p>Required: Yes</p> <p>Value: WAS</p> <p>Default value: TAM</p>	<p>The ETai adds users' credential information into the JAAS Subject. This information includes the users dn. Maps this dn to the WebSphere Application Server dn, or (Value = WAS). If a mapping is attempted for a user that does not exist in the WebSphere Application Server user registry, it is ignored and not added to the JAAS Subject.</p>
<p>Property name: com.ibm.websphere .security.webseal .tamGroupDnMapping</p> <p>Required: Yes</p> <p>Value: WAS</p> <p>Default value: TAM</p>	<p>The ETai adds users' credential information into the JAAS Subject. This information includes the group dn's. The ETai can be configured to either:</p> <p>Map these dn's to the WebSphere Application Server dn's, or (Value = WAS).</p> <p>If a mapping is attempted for a group that does not exist in the WebSphere Application Server user registry, it is ignored and not added to the JAAS Subject.</p>
<p>Property name: com.ibm.websphere .security.webseal .loginId</p> <p>Type: String</p> <p>Required: Yes</p> <p>Value: websealSSOID</p> <p>Default value: None</p>	<p>The value of this property must exist as a valid user in the user registry.</p> <p>If necessary, create a new user in the Dashboard Application Service Hub registry called websealSSOID.</p> <p>The ETai must be configured with the username of the WebSEAL trusted user. This is the single sign-on user that is authenticated using the password in the Basic Authentication header inserted by WebSEAL in the request. The format of the username is the short name representation.</p> <p>This property interacts with the following property:</p> <p>com.ibm.websphere.security .webseal.useWebSphereUserRegistry</p> <p>If com.ibm.websphere.security .webseal.useWebSphereUserRegistry is set to true then the specified user must exist in either the WebSphere Application Server user registry or the Tivoli Access Manager user registry.</p>

<i>Table 13. ETai custom properties (continued)</i>	
Property details	Notes
<p>Property name: com.ibm.websphere .security.webseal .checkViaHeader</p> <p>Type: String</p> <p>Required: Yes</p> <p>Value: true</p> <p>Default value: false</p>	<p>The ETai can be configured so that the Via header can be ignored when validating trust for a request. This property is required, if WebSEAL is to allow requests into the Dashboard Application Service Hub only from particular hosts.</p> <p>This property interacts with the following properties:</p> <ul style="list-style-type: none"> • com.ibm.websphere.security.webseal.hostnames • com.ibm.websphere.security.webseal.ports <p>If com.ibm.websphere.security.webseal.checkViaHeader is set to false then the values set for the two associated properties are not used.</p>
<p>Property name: com.ibm.websphere .security.webseal.id</p> <p>Required: Yes</p> <p>Value: iv-creds</p> <p>Default value: iv-creds</p>	<p>Iv-creds carries end user credentials, which is used by Dashboard Application Service Hub for authorization.</p> <p>Note: Any additional values set for this property are added to a list along with Iv-creds, that is, Iv-creds is a required header for the ETai.</p>

Table 13. ETai custom properties (continued)

Property details	Notes
<p>Property name: <code>com.ibm.websphere</code> <code>.security.webseal</code> <code>.hostnames</code></p> <p>Required: Yes</p> <p>Value: A comma separated list of strings.</p> <p>Default value: There is no default value for this property.</p>	<p>The ETai can be configured so that the request must arrive from a list of expected hosts. If any of the hosts in the <code>Via</code> header of the HTTP request are not listed in the values set for this property, the request is ignored by the ETai.</p> <p>This property interacts with the following property: <code>com.ibm.websphere.security.webseal.ports</code></p> <p>All of the values listed for <code>com.ibm.websphere.security.webseal.hostname</code>s are used with the ports listed for <code>com.ibm.websphere.security.webseal.ports</code> to indicate a trusted host.</p> <p>For example, if: <code>com.ibm.websphere.security.webseal.hostname</code>s is set to <code>abc,xyz</code> <code>com.ibm.websphere.security.webseal.ports</code> is set to <code>80,443</code></p> <p>Then, the <code>Via</code> header is checked for these hostname/port combinations: <code>abc:80</code>; <code>abc:443</code>; <code>xyz:80</code>; <code>xyz:443</code>.</p> <p>If <code>com.ibm.websphere.security.webseal.checkViaHeader</code> is set to <code>false</code> then the values set for <code>com.ibm.websphere.security.webseal.hostname</code>s are not used.</p>
<p>Property name: <code>com.ibm.websphere</code> <code>.security.webseal</code> <code>.ports</code></p> <p>Required: Yes</p> <p>Value: 443</p> <p>Default value: There is no default value for this property.</p>	<p>This property interacts with the following property: <code>com.ibm.websphere.security.webseal.hostname</code>s</p> <p>All of the values listed for <code>com.ibm.websphere.security.webseal.hostname</code>s are used with the ports listed for <code>com.ibm.websphere.security.webseal.ports</code> to indicate a trusted host.</p> <p>For more information, see the notes for <code>com.ibm.websphere.security.webseal.hostname</code>s.</p>

Table 13. ETai custom properties (continued)	
Property details	Notes
<p>Property name: com.ibm.websphere .security.webseal .ssoPwdExpiry</p> <p>Required: No</p> <p>Value: A positive integer.</p> <p>Default value: 600</p>	<p>Once trust has been established for a request, the password for the Single sign-on user is cached for subsequent trust validation of requests. This saves the ETai from having to re-authenticate the single sign-on user with the user registry for every request, therefore increasing performance. The cache timeout period can be modified by setting this property to the required time in seconds. If the password expiry property is set to 0, the cached password does not expire.</p>
<p>Property name: com.ibm.websphere .security.webseal .groupRealmPrefix</p> <p>Required: Yes</p> <p>Value: "group:"</p> <p>Default value: "group:"</p>	<p>This property is needed to map the group realm prefix from Tivoli Access Manager to group realm prefix in WebSphere Application Server registry.</p>
<p>Property name: com.ibm.websphere .security.webseal .userRealmPrefix</p> <p>Required: Yes</p> <p>Value: "user:"</p> <p>Default value: "user:"</p>	<p>This property is needed to map the user realm prefix from Tivoli Access Manager to user realm prefix in WebSphere Application Server registry.</p>

9. If a custom property does not exist, click **New** to configure a custom property and provide a name, value, and optional description and click **Apply** to add the custom property.
10. If the custom property exists, but is not in line with the details provided in the table above, click on the custom property entry, update its details and click **Apply** to modify the custom property.
11. Stop and restart the Dashboard Application Service Hub Server:
 - a) In the /opt/IBM/JazzSM/profile/bin directory, depending on your operating system, enter one of the following commands:
 - **Windows** stopServer.bat server1
 - **Linux** | **UNIX** stopServer.sh server1

Note: On UNIX and Linux systems, you are prompted to provide an administrator username and password.
 - b) In the /opt/IBM/JazzSM/profile/bin directory, depending on your operating system, enter one of the following commands:
 - **Windows** startServer.bat server1

- `startServer.sh server1`

What to do next

Configure the Tivoli Access Manager WebSEAL by creating a WebSEAL junction and creating a junction mapping table.

Checking your Tivoli Access Manager configuration

To ensure that your Tivoli Access Manager configuration is valid, you can carry out a number of checks.

Before you begin

Ensure that you have the following software versions installed:

- Tivoli Access Manager version 6.1
- Dashboard Application Service Hub Server, version 1.1 fix pack 11 or later

About this task

This topic describes how to check the following items:

- The status of the Tivoli Access Manager server.
- Connecting to the Dashboard Application Service Hub Server.

Procedure

1. To check the status of the Tivoli Access Manager server, at the command line, enter `pd start status`.

The following output indicates that the Tivoli Access Manager server is running:

```
pdmgrd      yes      yes
pdacl      yes      no (sometimes yes)
pdmgrproxyd no      no
webseald-ip1 yes      yes
```

2. To check if the Lightweight Directory Access Protocol (LDAP) user registry is active:
 - a) At the command line, enter `pdadmin -a sec_master -p sec_master_password`.

Note: This command assumes that `pdadmin` is in the path.

Expected output:

```
pdadmin -a sec_master -p sec_master_password
```

- b) At the command line, enter `user list * 10`.

Example output:

```
sec_master
ivmgrd/master
ivacl/ip1
ip1-webseald/ip1
```

- c) To quit, at the command line, enter `quit`.
3. If the Tivoli Access Manager processes are not started, at the command line enter `pd start start`.
If the processes are already started, the following output can be expected:

```
Starting the: Access Manager authorization server
Could not start the server
```

4. To check that you can connect from the Dashboard Application Service Hub Server to the Tivoli Access Manager computer:

- a) On the Dashboard Application Service Hub Server use a Web browser to connect to `http://tam_server_hostname`. A security message may be displayed, confirm the Tivoli Access Manager self-signed certificate to display an authorization dialog.
- b) Enter a username and password to display the Tivoli Access Manager WebSEAL splash screen (username = `sec_master`, password = `sec_master_password`).

What to do next

Configure the WebSEAL keystore.

Configuring the WebSEAL keystore

To allow the application server to use Tivoli Access Manager WebSEAL, you must import Dashboard Application Service Hub Server security certificate to the WebSEAL keystore.

About this task

To export the Dashboard Application Service Hub Server security certificate and import it into the WebSEAL keystore:

Procedure

1. Log in to the Dashboard Application Service Hub console.
2. Export the Dashboard Application Service Hub X.509 certificate.

The process for exporting varies depending on your browser. Refer to your browser documentation for assistance.

For example, the following substeps describe how you can export the certificate using a Firefox browser:

 - a) Double-click on lock icon that appears in the browser window to display the **Security** dialog for the Web page.
 - b) Click **View Certificate** and in the **Certificate Viewer** dialog and then click the **Details** tab.
 - c) Click **Export** and in the **Save Certificate To File** dialog and select a directory to export the Dashboard Application Service Hub X.509 certificate.
3. Copy the exported certificate file to the Tivoli Access Manager computer.
4. On the Tivoli Access Manager computer, at the command line, change to the directory that hosts the IKeyman utility.

For example, the following directories reflect typical locations for the IKeyman utility, but it may vary depending on your environment:

 - **Linux** **UNIX** `WAS_home/profiles/profile_name/bin/`
 - **Windows** `WAS_home\java\jre\bin\`
5. Start the IKeyman utility and complete the substeps:
 - **Linux** **UNIX** At the command line, enter `./ikeyman.sh`
 - **Windows** At the command line, enter `ikeyman.exe`
 - a) On the toolbar, click **Open** to display the **Open** window.
 - b) Select CMS as the key database type.
 - c) Click **Browse** and from `/var/pdweb/www-ip1/certs`, select `pdsrv.kdb` to display the **Password Prompt** dialog.

The default password reflects the file name, that is, `pdsrv`.
 - d) In the Key database content section, select **Signer Certificates** and click **Add**.
 - e) In the **Add CA's Certificate from a File** dialog, for the **Data type**, select the Base64-encoded ASCII data option and click **Browse**.

- f) Locate the Dashboard Application Service Hub X.509 certificate and enter a label for the certificate (for example, `tipmachine`).
 - g) Click **Save** to add the certificate to the WebSEAL keystore (do not change the certificate's file name).
6. To restart Tivoli Access Manager WebSEAL, at the command line, enter `pdweb restart`.

The following is the expected output:

```
Stopping the: webseald-ip1
Starting the: webseald-ip1
```

What to do next

Create a WebSEAL junction.

Creating a WebSEAL junction

A WebSEAL junction is an HTTP or HTTPS connection between a front-end WebSEAL server and a back-end Web application server, for example the Dashboard Application Service Hub Server.

About this task

Junctions logically combine the Web space of the back-end server with the Web space of the WebSEAL server, resulting in a unified view of the entire Web object space. To create a junction:

Procedure

1. On the Tivoli Access Manager computer, at the command line, enter `pdadmin -a sec_master_account -p sec_master_password`.
2. At the command line, enter `s l`.

The following is the expected output:

```
ivaclld-ip1
ip1-webseald-ip1
```

Note: Where `ip1` is the hostname of the Tivoli Access Manager computer.

3. Enter `s t ip1-webseald-ip1 list`.

The following is the expected output:

```
/
```

4. Enter `s t ip1-webseald-ip1 create -t ssl -c iv-creds -b supply -h tbsm_hostname/ip -p tbsmadmin_console_secure_port /tip`.

Where:

```
s t = server task
ip1-webseald-ip1 = WebSEAL instance name
-t ssl = transport type is SSL
-c iv-creds = needed for single sign on (SSO) to work, carry credential
of user
-b supply = basic authorization header needed for SSO to work
```

The following is the expected output:

```
Created junction at /tip
```

Note: If you want to delete a junction, enter `s t ip1-webseald-ip1 delete /tip`.

Note: If you want to show details for a junction, enter `s t ip1-webseald-ip1 show /tip`.

What to do next

Create a WebSEAL junction mapping table.

Creating a WebSEAL junction mapping table

A junction mapping table maps specific target resources to junction names. Junction mapping is an alternative to a cookie-based solution for filtering dynamically generated server-relative URLs.

About this task

To create a WebSEAL junction mapping table:

Procedure

1. On the Tivoli Access Manager computer, in a text editor open the WebSEAL configuration file, `/opt/pdweb/etc/webseald-ip1.conf`.
2. In the `[junction]` section, edit the `jmt-map` path so that it reads `jmt-map = lib/jmt.conf`.

Note: This path is relative to the server root path. Check the server root path in the `[server]` section of the file and take a note of the full `jmt-map` path. For example, `/opt/pdweb/www-ip1/lib/jmt.conf`.

3. In a text editor create or edit open the `jmt.conf` file and add or modify the following:

- `/tip /ibm/console/*`

Note: The `/ibm/console/` element of the path shown assumes that the Dashboard Application Service Hub root context path was not reconfigured at installation time.

- `/tip /ibm/sla/*`

4. To load the `jmt.conf` file into WebSEAL, enter `s t ip1-webseald-ip1 jmt load`.

The following is the expected output:

```
DPWWM1462I    JMT Table successfully loaded
```

5. To restart the WebSEAL server, enter `pdweb restart`.

The following is the expected output:

```
Stopping the: webseald-ip1
Starting the: webseald-ip1
```

What to do next

Test the WebSEAL junction.

Testing the WebSEAL junction

Once you have created a WebSEAL junction, you can test it.

About this task

To test a WebSEAL junction:

Procedure

1. In your Web browser's address bar, enter `https://tam_server_hostname/tip/ibm/console`, where `tip` is the name of the WebSEAL junction.
The Dashboard Application Service Hub login page is displayed.
2. To test if Tivoli Access Manager challenges you when you try to access the Dashboard Application Service Hub:
 - a) Close all instances of your Web browser.
 - b) Start your Web browser and go to `https://tam_server_hostname/tip/ibm/console/`.

Note: The `/ibm/console/` element of the URL shown assumes that the Dashboard Application Service Hub root context path was not reconfigured at installation time.

If the WebSEAL junction is working as expected, an **Authentication Required** dialog is displayed and you have to provide Tivoli Access Manager account (`sec_master`) details to proceed.

What to do next

Edit `customizationProperties.xml` to ensure that when you log out of Dashboard Application Service Hub that you also log out from Tivoli Access Manager.

Configuring single sign off for Tivoli Access Manager and Dashboard Application Service Hub

To ensure that you when you log out from the Dashboard Application Service Hub that you also log out from Tivoli Access Manager, you must edit `customizationProperties.xml`.

About this task

To configure single sign off for the Dashboard Application Service Hub Server and the Tivoli Access Manager computer:

Procedure

1. In a text editor, open `/opt/IBM/JazzSM/profile/config/cells/JazzSMNode01Cell/applications/isc.ear/deployments/isc/isclite.war/WEB-INF/customizationProperties.xml`.

Windows

For example: `C:\IBM\tivoli\tipv2C:\program Files\IBM\JazzSM\profile\config\cells\JazzSMNode01Cell\applications\isc.ear\deployments\isc\isclite.war\WEB-INF\customizationProperties.xml`

2. Edit the `TAMJunctionName` property, as follows:

```
<consoleproperties:console-property id="TAMJunctionName" value="tip"/>
<consoleproperties:console-property id="WebSealServerName" value=""/>
```

Where:

- `TAMJunctionName` is the junction name in Tivoli Access Manager that is configured to point at the Dashboard Application Service Hub Server.
- `WebSealServerName` is a Tivoli Access Manager WebSEAL server instance name. This property allows the Dashboard Application Service Hub Server process requests from declared WebSEAL hosts.

Results

When you log out from the Dashboard Application Service Hub, a `Successful Logout` message is displayed in your browser. This indicates that you logged out from both the Dashboard Application Service Hub and Tivoli Access Manager.

Setting form-based authentication for WebSEAL

Tivoli Access Manager provides form-based authentication as an optional alternative to the standard Basic Authentication mechanism.

About this task

For information on WebSEAL authentication and changing from basic mode to the form-based mode refer to Tivoli Access Manager documentation at http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itame.doc_6.1/am61_webseal_admin74.htm#chpt4_amwebpi_authent:

Configuring access for HTTP and HTTPS

By default, the application server requires HTTPS (Hypertext Transfer Protocol Secure) access. If you want some users to be able to log in and use the console with no encryption of transferred data, including user ID and password, configure the environment to support both HTTP and HTTPS modes.

Before you begin

After installing *Dashboard Application Service Hub* and before beginning this procedure, log in to the portal to ensure that it has connectivity and can start successfully.

About this task

Configuring for HTTP and HTTPS console access involves editing the `web.xml` file of Web components. Use this procedure to identify and edit the appropriate Web XML files.

Procedure

1. Change to the following directory: `/opt/IBM/JazzSM/profile/installedApps/JazzSMNode01Cell/`.
2. From this location, locate the `web.xml` files in the following directories:
 - For the Integrated Solutions Console web application archive: `isc.ear/deployments/isc/isc-lite.war/WEB-INF`
 - For the Dashboard Application Service Hub Charts web application archive: `isc.ear/deployments/isc/tip.charts.war/WEB-INF`
 - For the Dashboard Application Service Hub Change Password web application archive: `isc.ear/deployments/isc/TIPChangePasswd.war/WEB-INF`
 - Additionally for IBM Tivoli Business Service Manager, the `web.xml` files in the following directories must be updated to allow the use of HTTP:
 - `isc.ear/deployments/isc/sla.war/WEB-INF`
 - `isc.ear/deployments/isc/twa.war/WEB-INF`
 - `isc.ear/deployments/isc/impactAdmin.war/WEB-INF`
3. Open one of the `web.xml` files using a text editor.
4. Find the `<transport-guarantee>` element.

The initial value of all `<transport-guarantee>` elements is `CONFIDENTIAL`, meaning that secure access is always required.
5. Change the setting to `NONE` to enable both HTTP and HTTPS requests.

The element now reads: `<transport-guarantee>NONE</transport-guarantee>`.
6. Save the file, and then repeat these steps for the other `web.xml` deployment files.
7. Log in to *Dashboard Application Service Hub*.
8. In the navigation pane, click **Settings > Websphere Administrative Console** and click **Launch Websphere Administrative Console**.
9. In the WebSphere Application Server administrative console, select **Security > Global security** and click the **External authorization providers** link.
10. In the **External authorization providers** page, select the **Update with application names listed** option.
11. In the text pane, type `isc` and click **Apply**.
12. In the messages area at the top of the page, click the **Save** link to commit your changes to the master configuration.
13. Stop and restart the Dashboard Application Service Hub Server:
 - a) In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:

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

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

b) In the /opt/IBM/JazzSM/profile/bin directory, depending on your operating system, enter one of the following commands:

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

Example

The following example is a section of the web.xml file for TIPChangePasswd where the transport-guarantee parameter is set to NONE:

```
<security-constraint>
  <display-name>
    ChangePasswdControllerServletConstraint</display-name>
  <web-resource-collection>
    <web-resource-name>ChangePasswdControllerServlet</web-resource-name>
    <url-pattern>*/*/url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <description>Roles</description>
    <role-name>administrator</role-name>
    <role-name>operator</role-name>
    <role-name>configurator</role-name>
    <role-name>monitor</role-name>
    <role-name>iscadmins</role-name>
  </auth-constraint>
  <user-data-constraint>
    <transport-guarantee>NONE</transport-guarantee>
  </user-data-constraint>
</security-constraint>
```

What to do next

Users must now specify a different port, depending on the mode of access. The default port numbers are as follows:

http://<host_name>:16310/ibm/console

Use the HTTP port for logging in to the Dashboard Application Service Hub on the HTTP port .

https://<host_name>:16311/ibm/console

Use the HTTPS secure port for logging in to the Dashboard Application Service Hub.

Note: If you want to use single sign-on (SSO) then you must use the fully qualified domain name of the Dashboard Application Service Hub host.

Configuring the LPTA token timeout value

You can configure the Lightweight Third Party Authentication (LTPA) token timeout value for Dashboard Application Service Hub in the WebSphere Application Server console.

Before you begin

Dashboard Application Service Hub is enabled for single sign-on.

About this task

The default timeout for an LTPA token is 120 minutes. An LTPA timeout causes you to be logged out from Dashboard Application Service Hub and can also cause an authentication popup message, if the first request after the timeout is an AJAX request from a portlet. To configure the LTPA token timeout:

Procedure

1. In the Dashboard Application Service Hub navigation pane, click **Settings > WebSphere Admin Console**.
2. Click **Launch WebSphere Admin Console** to start the WebSphere Application Server console.
3. In the WebSphere Application Server console navigation pane, click **Security > Global security**.
4. In the Authentication area of the **Global security** page, click the **LTPA** link.
5. In the LTPA timeout area of the **LTPA** page, edit the value for the LTPA timeout and click **OK**.
6. 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

In a load balanced environment, you must set the LTPA token timeout value on each of the Dashboard Application Service Hub Server instances.

Deleting a data source definition

Before you create a CMS data source, in some circumstance you may want to delete an existing data source definition.

About this task

As part of the Data Integration Services (DIS) database creation, the DBConfig installer also creates an external CMS database. Dashboard Application Service Hub applications use an external CMS database to both publish their CMS launch definitions as well as to obtain the launch definitions from other products. Tivoli Business Service Manager creates a data source definition in WebSphere Application Server for the Data Integration Services (DIS) database, CMS infers the CMS external database location from this since the CMS tables are created in the DIS database. If the CMS external database tables reside in the DIS database, then there may not be an existing CMS data source and the DIS datasource is used instead. If this is the case then the data source does not need to be removed.

To delete a data source:

Procedure

1. Run the following command to list existing data sources:

```
$AdminConfig list DataSource
```

2. Run the following command to remove the data source:

```
$AdminConfig remove ds_name_string
```

Where *ds_name_string* is the name of the data source (which was displayed after you completed Step 1) that you want to remove.

3. Save your changes:

```
$save
```

Configuring the CMS database

The Context Menu Service (CMS) is a component of Dashboard Application Service Hub which can be used by TBSM to share information outside of the Dashboard Application Service Hub environment.

CMS facilitates *launch-in-context* capability between products. The term *launch-in-context* is used to describe the ability for one application to invoke a function or launch a user interface provided by another application while also passing in data that the function or user interface may immediately process. CMS enables *launch-in-context* by allowing a product to register launch points for itself and locate launch points for other products. Launch points provide information to allow an application to invoke a function or user interface from another application.

If you want to change the location of the CMS database after installation, you must update the Dashboard server.

Creating a database for CMS

Copy CMS scripts from your Dashboard Application Service Hub installation to your remote computer and create a database.

About this task

To create a remote database for CMS:

Procedure

1. On the computer running Dashboard Application Service Hub, at the command line, change to the following directory:

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

The CMS directory contains a number of scripts that are provided by Dashboard Application Service Hub. The script that you use depends on the type of database and the operating system of the database computer:

- **Linux** **UNIX** db2_scripts.tar for a DB2 database
- **Linux** **UNIX** MsSql_scripts.tar for a Microsoft SQL Server database
- **Linux** **UNIX** Oracle_scripts.tar for an Oracle database
- **Windows** db2_scripts.zip for a DB2 database
- **Windows** MsSql_scripts.zip for a Microsoft SQL Server database
- **Windows** Oracle_scripts.zip for an Oracle database

The steps described here reflect setting up a DB2 database on a on a Microsoft Windows system.

2. Transfer a copy of the relevant script file from the CMS directory to your remote database computer and take note of the location in which you save the file.

For example, for a DB2 database running on a Microsoft Windows system, you need to transfer a copy of db2_scripts.zip to the remote computer.

3. On the remote database system, extract the file that you copied to a known location and at the command line change to that directory.

For example, for a DB2 database: `cd C:\demo\db2scripts\db2`

4. Open the CMS_database_type_Readme.txt file, in this case CMS_DB2_ReadMe.txt, in a text editor.

This file provides instructions and samples on how to use the scripts provided.

5. Open a database command window, so that you can execute database commands.

For example, for a DB2 database running on a Windows system, click **Start > IBM DB2 > DB2COPY1 (default) > Command Line Tools > Command Window**.

6. In the command window, change to the directory that contain your extracted script files.

For example, `cd demo\db2_scripts\db2`

7. Run the database **setup** command providing the relevant arguments to the parameters outlined in the CMS_database_type_Readme.txt file for the database **setup** command.

For example, run `CMS_DB2Setup.bat -d database_name -u database_user_name -p database_user_password`.

Where:

database_name

The name of the database that you want to create. You can also provide the name of an existing database.

database_user_name

The user name for the database.

database_user_password

The user password associated with the specified user name.

The database is now ready to communicate with a Dashboard Application Service Hub data source.

What to do next

When you have set up a remote database, you can configure a data source in Dashboard Application Service Hub that CMS can use.

Configuring a hostname to be used by CMS

Configure a hostname to be used by CMS.

About this task

You need to set a hostname that CMS can use. For example, in a load balanced environment, it may not be obvious which hostname CMS should use. To specify a hostname to CMS:

Procedure

1. On the computer running Dashboard Application Service Hub, at the command line, change to the following directory:

```
/opt/IBM/JazzSM_bkup/ui/bin/cms
```

2. Run the **cmssetconf** command to view details of the different options that are available to you in setting up CMS to use the remote database.

```
Linux ./cmssetconf.sh
```

```
Windows cmssetconf.bat
```

One of the settings that you apply using the **cmssetconf** command, is the hostname.

3. Run the following command to specify the hostname that you want to use:

```
Linux ./cmssetconf.sh -hostname hostname -port DASH_port_number
```

```
Windows cmssetconf.bat -hostname hostname -port DASH_port_number
```

The hostname is now configured.

4. Run the following command to review your CMS configuration and verify that you have correctly specified the hostname:

```
Linux ./cmsshowconf.sh -hostname hostname -port DASH_port_number
```

```
Windows cmsshowconf.bat -hostname hostname -port DASH_port_number
```

5. Stop and restart the Dashboard Application Service Hub Server:

- a) In the /opt/IBM/JazzSM/profile/bin directory, depending on your operating system, enter one of the following commands:

- **Windows** stopServer.bat server1

- **Linux** | **UNIX** stopServer.sh server1

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

b) In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:

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

What to do next

When you have configured the hostname, you can set up logging for CMS.

Administering

The administrator tasks involve configuring and customizing the environment and controlling access to it.

In a single installation the Dashboard Application Service Hub provides a product design environment and customization, with services that enable multiple-product integration.

Logging in

Log in to the portal whenever you want to start a work session.

Before you begin

The Dashboard Application Service Hub Server must be running before you can connect to it from your browser.

About this task

Complete these steps to log in:

Procedure

1. In a Web browser, enter the URL of the Dashboard Application Service Hub Server: `http://host.domain:16310/ibm/console` or `https://host.domain:16311/ibm/console` if it is configured for secure access.
 - `host.domain` is the fully qualified host name or IP address of the Dashboard Application Service Hub Server (such as `MyServer.MySubdomain.MyDomain.com` or `9.51.111.121`, or `localhost` if you are running the Dashboard Application Service Hub Server locally).
 - `16310` is the default nonsecure port number for the portal and `16311` is the default secure port number. If your environment was configured with a port number other than the default, enter that number instead. If you are not sure of the port number, read the application server profile to get the correct number.
 - `ibm/console` is the default path to the Dashboard Application Service Hub Server, however this path is configurable and might differ from the default in your environment.
2. In the login page, enter your user ID and password and click **Log in**.



Attention: After authentication, the web container used by the Dashboard Application Service Hub Server redirects to the last URL requested. This is usually `https://<host>:<port>/ibm/console`, but if you manually change the page URL, after being initially directed to the login page, or if you make a separate request to the server in a discrete browser window before logging in, you may be redirected unexpectedly.

Note: If you have more than one instance of the Dashboard Application Service Hub Server installed on your computer, you should not run more than one instance in a browser session, that is, do not log in to different instances on separate browser tabs.

Results

After your user credentials have been verified, the Welcome page is displayed. If you entered the localhost or port number incorrectly, the URL will not resolve. View the application server profile to check the settings for localhost, port, and user ID.

What to do next

Select any of the items in the navigation tree to begin working with the console.

While you are logged into the Dashboard Application Service Hub Server, avoid clicking the browser **Back** button because you will be logged out automatically. Click **Forward** and you will see that you are logged out and must resubmit your credentials to log in again.

Note: If you want to use single sign-on (SSO) then you must use the fully qualified domain name of the Dashboard Application Service Hub host.

Stopping and starting the application server

The Dashboard Application Service Hub Server starts automatically after it has been installed, and on systems running Windows, whenever the computer is started.

About this task

You can manually stop the Dashboard Application Service Hub Server before beginning certain configuration tasks or as needed.

Note: For environments using a central user repository, for example LDAP, a user must be given the *Administrator* role in the WebSphere Application Server administrative console before they can stop the Dashboard Application Service Hub Server. For information on assigning WebSphere Application Server roles, see: http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/tsec_tselugadro.html

Procedure

1. In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:

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

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

2. In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:

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

Port assignments

The application server requires a set of sequentially numbered ports.

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

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 /opt/IBM/JazzSM/profile/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 /opt/IBM/JazzSM/profile/logs/AboutThisProfile.txt:

```
Application server environment to create: Application server
Location: C:\Program Files\IBM\JazzSM\profile
Disk space required: 200 MB
Profile name: DASHProfile
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 /opt/IBM/JazzSM/var/JazzSMPprofile_portDef.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
```

Changing passwords

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

About this task

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



Attention: If you are an administrator and you want to change the password for the tbsmadmin administrator and the Tivoli Netcool/OMNIbus ObjectServer root user, you must use the **Settings > WAS Admin Console** portlet to change their password. Do not use the **Users and Groups > Manage Users** portlet.

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

To change passwords:

Procedure

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



Attention:

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

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

Changing password on nameserver

The Dashboard Application Service Hub server uses the properties in the `$JAZZSM_HOME/profile/installedApps/JazzSMNode01Cell/isc.ear/sla.war/etc/nameserver.props` file to determine the Name Server connection details.

If for any reason the `impactadmin` password changes (for example, if you switch from file-based user registry to LDAP), you must update the password details for the `impactadmin` user in the `nameserver.props` file. To do so, you must manually encrypt the password and then edit the `nameserver.props` file.

For example, to retrieve the encrypted password for `impactadmin`, use the following command:

```
/opt/IBM/WebSphere/AppServer/bin/crypto.sh {impactadmin user password}
```

Then edit the `nameserver.props` file with the encrypted password:

```
impact.nameserver.userid=impactadmin
impact.nameserver.password={AES}A1FD4E0DE2389E727873104C811FB744
```

Exporting and importing

You can export customized configuration data from an existing *Dashboard Application Service Hub* installation to another by exporting the data and subsequently importing the exported data.

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

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

You can export and import the following elements:

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

Note: Copies of a portlet entity are not exported; either through the console Export Wizard or through the **tipcli.bat|.sh Export** command.
 - View profiles.
 - Events and wires.
 - Access permissions.
 - Navigation structure.
- Custom views (or customized system views).

Note: You can also export pages associated with a view if the `exportpageinview` parameter is set to `true`.
- Custom roles, including:
 - Role name, creation date, and update date.
 - Role mapping information in relation to users and groups.
 - Associated role preference, that is, the relevant console preference profile.
- Console properties and customization properties, including:
 - Transformations.
 - Themes and images.
 - Bundles.

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

- All the required applications (WAR files) must be deployed on all computers in the pool.
- The load balanced pool configuration must be locked during the import operation.
- The import operation must be ran on one of the nodes in the pool.

Restriction: In a load balanced environment that includes charting, the `ListRestore` command only runs successfully on the node that is used for the import operation because backup files are stored locally on that node and are not synchronized across other nodes in the cluster.

- You must provide the load balancing manager an updated file list to update the load balancing scope. The migration tool plugin provides the file list.
- The load balanced pool configuration, can then be unlocked.
- The import of transformations in a load balanced environment is not supported. Transformations must be imported to each node independently.

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

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

Basic export commands

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

Exporting pages in simplified mode

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

Before you begin

Ensure that the Dashboard Application Service Hub Server is running.

About this task

To export specific pages in simplified mode for an instance of Dashboard Application Service Hub:

Procedure

1. At the command line change to: `/opt/IBM/JazzSM/profile/bin`.
2. To return a list of customized pages that can be exported, run the following command:

- **Windows** `C:\program Files\IBM\JazzSM\ui\bin\tipcli.bat ListPages --customizePages true`
- **Linux** **UNIX** `tip_home_dir/opt/IBM/JazzSM/ui/bin/tipcli.sh ListPages --customizePages true`

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

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

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

- **Windows** `tip_home_dirC:\program Files\IBM\JazzSM\ui\bin\tipcli.bat ExportPage --uniqueName pageID_1,pageID_2,pageID_3 --username tbsmadmin_user_name --password tbsmadmin_password`
- **Linux** **UNIX** `tip_home_dir/opt/IBM/JazzSM/ui/bin/tipcli.sh ExportPage --uniqueName pageID_1,pageID_2,pageID_3 --username tbsmadmin_user_name --password tbsmadmin_password`

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

Results

When the command completes, a `Data.zip` file is created in `/opt/IBM/JazzSM/ui/output`.

What to do next

Locate `/opt/IBM/JazzSM/ui/output/Data.zip` and copy it to the computer where you intend to apply the exported customization data.

Exporting views in simplified mode

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

Before you begin

Ensure that the Dashboard Application Service Hub Server is running.

About this task

To export specific views in simplified mode for an instance of Dashboard Application Service Hub:

Procedure

1. At the command line change to: `/opt/IBM/JazzSM/profile/bin`.
2. Optional: To return a list of customized views that can be exported, run the following command:

- **Windows** `C:\Program Files\IBM\JazzSM\ui\bin\tipcli.bat ListViews`

- **Linux** **UNIX** `/opt/IBM/JazzSM/ui/bin/tipcli.sh ListViews`

3. Review the list of returned view records and take note of the view IDs for the views that you want to export.

4. To export specific views, run the following command:

- **Windows** `C:\Program Files\IBM\JazzSM\ui\bin\tipcli.bat ExportView --uniqueName viewID_1, viewID_2, viewID_3`

- **Linux** **UNIX** `/opt/IBM/JazzSM/ui/bin/tipcli.sh ExportView --uniqueName viewID_1, viewID_2, viewID_3`

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

Results

When the command completes, a `Data.zip` file is created in `/opt/IBM/JazzSM/ui/output`.

What to do next

Locate `/opt/IBM/JazzSM/ui/output/Data.zip` and copy it to the computer where you intend to apply the exported customization data.

Exporting console preference profiles in simplified mode

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

Before you begin

Ensure that the Dashboard Application Service Hub Server is running.

About this task

To export console preference profiles in simplified mode:

Procedure

1. At the command line change to: `/opt/IBM/JazzSM/profile/bin`.
2. Optional: To return a list of console preference profiles that can be exported:

- **Windows** C:\Program Files\IBM\JazzSM\ui\bin\tipcli.bat
ListPreferenceProfiles
 - **Linux** **UNIX** /opt/IBM/JazzSM/ui/bin/tipcli.sh
ListPreferenceProfiles
3. Review the list of returned records and take note of the unique names for the console preference profiles that you want to export.
 4. To export specific console preference profiles, run the following command:
 - **Windows** C:\Program Files\IBM\JazzSM\ui\bin\tipcli.bat ExportProfile --uniqueName profile_ID1,profile_ID2,profile_ID3
 - **Linux** **UNIX** /opt/IBM/JazzSM/ui/bin/tipcli.sh ExportProfile --uniqueName profile_ID1,profile_ID2,profile_ID3
- Note:** The file portletEntities.xml is always exported, even if you specify NONE as an argument to the uniqueName parameter.

Results

When the command completes, a Data.zip file is created in /opt/IBM/JazzSM/ui/output/.

What to do next

Locate /opt/IBM/JazzSM/ui/output/Data.zip and copy it to the computer where you intend to apply the exported customization data.

Advanced export commands

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

Exporting all customization data

You can export all customization data for an instance of Dashboard Application Service Hub in one command.

Before you begin

Ensure that the Dashboard Application Service Hub Server is running.

About this task

To export all customization data for an instance of Dashboard Application Service Hub:

Procedure

1. At the command line change to: /opt/IBM/JazzSM/ui/bin.
2. Optional: To return a list of plugins that will be run during the export operation, run the following command:
 - **Linux** **UNIX** /opt/IBM/JazzSM/ui/bin/tipcli.sh ListExportPlugins
 - **Windows** C:\Program Files\IBM\JazzSM\ui\bin\tipcli.bat ListExportPlugins
3. To export all customization data, run the following command:
 - **Linux** **UNIX** /opt/IBM/JazzSM/ui/bin/tipcli.sh Export --username tbsmadmin_user_name --password tbsmadmin_password
 - **Windows** C:\Program Files\IBM\JazzSM\ui\bin\tipcli.bat Export --username tbsmadmin_user_name --password tbsmadmin_password

Results

When the Export command completes, a Data.zip file is created in /opt/IBM/JazzSM/ui/output/.

Note:

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

What to do next

Locate /opt/IBM/JazzSM/ui/output/Data.zip and copy it to the computer where you intend to apply the exported customization data.

Exporting using a properties file

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

Before you begin

By default, the tipcli command uses the /opt/IBM/JazzSM/ui/etc/tipcli.properties file unless this behavior is overridden by the specifying a discrete settings file using the settingFile parameter.

Ensure that the Dashboard Application Service Hub Server is running.

About this task

To export customization data using a properties file:

Procedure

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

Below is example content for an export properties file:

```
import.includePlugins=ImportPagePlugin
export.includePlugins=ExportPagePlugin
import.backupDir=c:/tmp/bkups
export.exportFile=c:/tmp/extest.zip
import.importFile=c:/tmp/extest.zip
username=tbsmadmin
password=tbsmadmin_password
import.haSupport=true
```

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

2. At the command line change to: /opt/IBM/JazzSM/ui/bin.
3. To export customization data based on the contents of a specific properties file, run the following command:

- **Linux** **UNIX** /opt/IBM/JazzSM/ui/bin/tipcli.sh Export --username *tbsmadmin_user_name* --password *tbsmadmin_password* --settingFile *export_properties_file*
- **Windows** C:\Program Files\IBM\JazzSM\ui\bin\tipcli.bat Export --username *tbsmadmin_user_name* --password *tbsmadmin_password* --settingFile *export_properties_file*

Where:

export_properties_file

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

Note: **Windows** You must use double backslashes characters (\\) when specifying the path to your settings file.

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

Results

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

What to do next

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

Exporting specific pages

When exporting Dashboard Application Service Hub data, you can specify that you want to export particular pages.

Before you begin

Ensure that the Dashboard Application Service Hub Server is running.

About this task

To export specific pages for an instance of Dashboard Application Service Hub:

Procedure

1. At the command line change to: `/opt/IBM/JazzSM/ui/bin/`.
2. To return a list of customized pages that can be exported, run the following command:

- **Windows** `C:\Program Files\IBM\JazzSM\ui\bin\tipcli.bat ListPages -- customizePages true`
- **Linux** **UNIX** `/opt/IBM/JazzSM/ui/bin//tipcli.sh ListPages -- customizePages true`

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

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

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

- **Linux** **UNIX** `/opt/IBM/JazzSM/ui/bin/tipcli.sh Export --username tbsmadmin_user_name --password tbsmadmin_password --pages pageID_1, pageID_2, pageID_3`

- **Windows** `C:\Program Files\IBM\JazzSM\ui\bin\tipcli.bat Export --username tbsmadmin_user_name --password tbsmadmin_password --pages pageID_1, pageID_2, pageID_3`

Results

When the command completes, a `Data.zip` file is created in `/opt/IBM/JazzSM/ui/bin/output/`.

What to do next

Locate `/opt/IBM/JazzSM/ui/bin/output/Data.zip` and copy it to the computer where you intend to apply the exported customization data.

Exporting specific views

When exporting Dashboard Application Service Hub data, you can specify that you want to export particular views.

Before you begin

Ensure that the Dashboard Application Service Hub Server is running.

About this task

To export specific views for an instance of Dashboard Application Service Hub:

Procedure

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

Where:

exportpageinviews

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

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

Results

When the command completes, a `Data.zip` file is created in `/opt/IBM/JazzSM/ui/bin/output/`.

What to do next

Locate `/opt/IBM/JazzSM/ui/bin/output/Data.zip` and copy it to the computer where you intend to apply the exported customization data.

Rules for exporting

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

The following rules apply when exporting customized configuration data from a Dashboard Application Service Hub environment:

Rules and options for pages

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

Rules and options for views

1. You can export a particular view by view ID or choose to export all views.
2. You can optionally export all views that contains a specified page.
3. The default export scope is `All`.
4. You can optionally export all pages associated with the views that you want to export.
5. If an view has a default node in the navigation pane associated with it, then that page is automatically exported with the view.
6. Views that match the following conditions should not be exported as the subsequent import of that view will fail:
 - An empty view, that is, a view that contains no pages or roles.
 - A view that contains roles, but no pages.
 - A view that contains empty pages, that is, the page exists but it does not contain portlets.

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

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

Rules and options for user preferences

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

Rules and options for console properties and customization properties

All console properties and customization properties are exported.

Rules and options for transformations

All transformations are exported.

Import commands

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

Importing previously exported data

You can import data that was exported from another instance of Dashboard Application Service Hub.

Before you begin

Ensure that the Dashboard Application Service Hub Server is running.

Ensure that you have run the export operation on an originating instance of the Dashboard Application Service Hub Server and that you have copy the output file (data.zip) to the following directory on the other instance:

```
/opt/IBM/JazzSM/ui/bin/output
```

About this task

To import data from a data.zip file that was exported from another instance Dashboard Application Service Hub Server:

Procedure

1. At the command line change to: \$JAZZSM_HOME/profile/bin.
2. Optional: To return a list of plugins that will be run during the import operation, run the following command:

- **Windows** C:\Program Files\IBM\JazzSM\ui\bin\tipcli.bat ListImportPlugins

- **Linux** **UNIX** /opt/IBM/JazzSM/ui/bin/tipcli.sh ListImportPlugins

3. To import the customization data, run the following command:

- **Windows** C:\Program Files\IBM\JazzSM\ui\bin\tipcli.bat Import --username *tbsmadmin_user_name* --password *tbsmadmin_password*

- **Linux** **UNIX** /opt/IBM/JazzSM/ui/bin/tipcli.sh Import --username *tbsmadmin_user_name* --password *tbsmadmin_password*

Results

When the **Import** command completes, the imported data is merged with the existing Dashboard Application Service Hub environment.

Rolling back imports

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

Before you begin

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

Ensure that the Dashboard Application Service Hub Server is running.

About this task

To roll back imports for a Dashboard Application Service Hub environment:

Procedure

1. At the command line change to: /opt/IBM/JazzSM/ui/bin/.
2. To rollback an import, run the following command:

- **Windows** C:\Program Files\IBM\JazzSM\ui\bin\tipcli.bat Import --rollback ALL
- **Linux** **UNIX** /opt/IBM/JazzSM/ui/bin/tipcli.sh Import --rollback ALL

When the command completes successfully, the Dashboard Application Service Hub environment is restored to the state that prevailed before the latest import operation was performed.

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

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

Rules for importing

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

The following rules apply when importing customized configuration data for a Dashboard Application Service Hub environment:

Rules and options for pages

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

Rules and options for views

1. All views included in an exported package are imported.
2. Views that match the following conditions should not be imported as the import operation for the view fails:
 - An empty view, that is, a view that contains no pages or roles.
 - A view that contains roles, but no pages.
 - A view that contains empty pages, that is, the page exists but it does not contain portlets.

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

All roles included in an exported package are imported.

Rules and options for user preferences

All user preferences included in an exported package are imported.

Rules and options for console properties and customization properties

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

Rules and options for transformations

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

[Table 1](#) provides details how various elements are processed during import:

Table 14. Rules for overwriting and merging during import

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

Changing the default security registry

The default security registry can be set at install time. Use this procedure to change the default registry after installation.

Before you begin

These steps require that your user ID has the Administrator role and that you know the base entry value of your repository. For LDAP or Microsoft Active Directory, this is usually a string like `ou=company,dc=country,dc=region`. For the ObjectServer, the base entry is `o=netcoolobjectServerRepository`.

About this task

If you want to change the default to a different registry, complete these steps:

Procedure

1. Log into the Dashboard Application Service Hub.
Your ID must have the Administrator role.
2. In the navigation pane, click **Settings** > **WebSphere Admin Console** and click **Launch WebSphere Admin Console**.
3. In the WebSphere Application Server administrative console navigation pane, click **Security** > **Secure administration, applications, and infrastructure**.
4. In the User account repositories area, select **Federated repositories** from the Available realm definitions, then click **Configure**.
5. Click **Supported entity types** under **Additional Properties**.
6. Click the entity type, then edit the **Base entry for the default parent** and **Relative Distinguished Name properties**.
7. After you click **OK** to save your changes, repeat the previous step to configure the other entity types.
For Microsoft Active Directory, the entity types (PersonAccount, Group, and OrgContainer) must be configured with a base DN and the RDN for PersonAccount should be `cn` instead of `uid`.
8. Stop and restart the Dashboard Application Service Hub Server:

a) In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:

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

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

b) In the `/opt/IBM/JazzSM/profile/bin` directory, depending on your operating system, enter one of the following commands:

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

CGI support

Use the initialization parameters to control the behavior of CGIServlet.

CGIServlet

CGI scripts run on a Web server and use the Common Gateway Interface (CGI) to perform tasks. The support for CGI in Dashboard Application Service Hub is provided by *CGIServlet*, extracted from Apache Tomcat. The Tomcat CGI support is largely compatible with the Apache HTTP Server but there are some limitations (such as only one `cgi-bin` directory). To change the configuration, edit `web.xml` in the directory where the CGI application is installed.

Servlet initialization parameters

Several initialization parameters are available for configuring the behavior of the CGIServlet.

cgiPathPrefix

The CGI search path will start at the Web application root directory + File.separator + this prefix.
Default setting: `cgiPathPrefix` is `Web-INF/cgi`.

debug

Determines the level of debugging detail for messages that are logged by the servlet. Default setting: `0`.

executable

This is type of the program to be used to run the script. Default setting: `perl`.

parameterEncoding

Names the parameter encoding to be used with the CGI servlet. Default setting: `System.getProperty("file.encoding", "UTF-8")`.

passShellEnvironment

Determines whether shell environment variables, if there are any, shall be passed to the CGI script.
Default setting: `false`.

Setting Java Virtual Machine memory for DASHProfile

You can increase the amount of memory available to the Dashboard Application Service Hub.

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 `/opt/IBM/JazzSM/profile/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 Dashboard Application Service Hub Server.

The changes take effect when the Dashboard Application Service Hub Server is restarted.



Attention: If you attempt to start the Dashboard Application Service Hub Server with a maximum heap size that is too large, error messages that are similar to the following are generated in the `/opt/IBM/JazzSM/profile/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.
```

Checking hostname settings

The value of the Hostname property in the `/opt/IBM/JazzSM/profile/properties` file is used by Dashboard Application Service Hub to convert incoming browser requests (for example, `http://<SystemName>:16310`) to the appropriate Dashboard Application Service Hub non-secure access (for example, `http://<HostnameValue>:16315/ibm/console`), which is then converted to the Dashboard Application Service Hub secure access (for example, `https://<HostnameValue>:16316/ibm/console/login.jsp`).

About this task

The Hostname property should contain the fully qualified hostname. This is required if the web browser being used to access Dashboard Application Service Hub is running on a machine in a different DNS domain to the Dashboard Application Service Hub Server (application server).

The value of the `/opt/IBM/JazzSM/profile/properties/tip.properties` file's Hostname entry is set during installation by a routine built into Java that checks the `/etc/hosts` (or `%WinDir%\system32\drivers\etc\hosts`) entry for the system; if the fully qualified domain name (FQDN) is not set in `/etc/hosts`, the Java routine returns either the short name or the IP address of the machine, depending on the type of operating system (all but AIX).

Therefore, before the *Dashboard Application Service Hub* installer is run, ensure that a line exists in `/etc/hosts` of the following form:

IP address FQDN shortname

For example: `9.10.11.12 yourserver.domainname.com yourserver`

This line ensures that the FQDN is set as the Hostname entry at install time in `/opt/IBM/JazzSM/profile/properties/tip.properties`.

If you try to connect to the application server and the URL conversion to the non-secure access appears to be working incorrectly, you should check Hostname property entry in `tip.properties`.

Procedure

1. Open the `/opt/IBM/JazzSM/profile/properties/tip.properties` file in a text editor.
2. Check the Hostname property and make sure the value can be correctly resolved by the web browser being used to access the application server.

3. Edit the Hostname entry to the FQDN of the application server and save the changes.
4. Stop and restart the application server.

The changes take effect when the application server is restarted.

Accessing Context Menu Service features

To access Context Menu Service features from within *Dashboard Application Service Hub*, you must be assigned the Monitor role in Dashboard Application Service Hub.

About this task

The Context Menu Service, a component of Dashboard Application Service Hub, facilitates launch-in-context capability between products. This capability enables one application to invoke a function or launch a user interface that is provided by another application while also passing data that the function or user interface can immediately process. To access Context Menu Service features, for example, CMS command line functions, you must be assigned the *Monitor* role in Dashboard Application Service Hub.

To assign the *Monitor* role to a user in Dashboard Application Service Hub:

Procedure

- You can assign roles to users in the portal or by using the `tipcli` command:
 - To assign the *Monitor* role to a user in the portal, from the navigation pane, click **Users and Groups** > **User Roles**. Search for the user, assign the *Monitor* role and save your changes.
 - To assign the *Monitor* role to a user using the `tipcli` command, at the command line change to `/opt/IBM/JazzSM/ui/bin/` and enter the following command:

```
Windows tipcli.bat MapUsersToRole --username DASH_username --password
DASH_user_password --roleName monitor --usersList user_ID
```

```
Linux UNIX tipcli.sh MapUsersToRole --username DASH_username
--password DASH_user_password --roleName monitor --usersList user_ID
```

Command reference

Use the Dashboard Application Service Hub command line interface *tipcli* commands for writing scripts for passing information between applications.

The `tipcli` commands are entered in the `/opt/IBM/JazzSM/ui/bin` directory, for example, `C:\Program Files\IBM\JazzSM\ui\bin\tipcli.bat` on Windows or `/opt/IBM/JazzSM/ui/bin/tipcli.sh` on Linux or UNIX.

The `tipcli` component provides help for its various commands:

Help [`--command command_name`]

Access help for all commands or optionally you can use the command argument to return detailed help for a specific command.

The following returns help for the `AddUpdatePreferenceProfile` command:

```
tipcli.bat Help --command AddUpdatePreferenceProfile
```

```
Help
----
AddUpdatePreferenceProfile --username <TIPusername> --password <passwordForUser>
--profileName <profileName> [--newProfileName <newProfileName>] [--themeDir <th
emeDir>] [--showNavTree <true|false>] [--componentDir <default|ltr|rtl>] [--text
Dir <default|contextual|ltr|rtl>] [--views <viewList>] [--roles <roleList>] [--d
efaultView <defaultView>]
where
<TIPusername> is the username on TIP that has iscadmins role.
<passwordForUser> is the password for the user.
<profileName> is profile name which will be created or updated.
<newProfileName> is the new name for the existing preference profile.
```

```
<themeDir> is the directory name of the installed theme. Example: TIPLight
<showNavTree> specify if show navigation tree by default after login the console.
<componentDir> specify component direction for the console.
<textDir> specify text direction for the console.
<viewList> is views assignment for the preference profile.
<roleList> is roles assignment for the preference profile.
<defaultView> specify which view is displayed by default after login the console.

CTGWA4017I The command completed successfully.
```

Working with roles

Use these tipcli commands for to manipulate roles.

ListRoles

Use the **ListRoles** command to list all roles configured for a portal instance.

Syntax

This command has the following syntax:

- **Linux** | **UNIX** tipcli.sh ListRoles
- **Windows** tipcli.bat ListRoles

Example

Linux | **UNIX** For example, in a UNIX or Linux environment, use the following command:

```
/opt/IBM/JazzSM/ui/bin/tipcli.sh ListRoles
```

Where */opt/IBM/JazzSM/* is location of the Dashboard Application Service Hub instance that you want to query.

AddRole

Use the **AddRole** command to add a specified role to the portal instance. Portal users are granted access to resources based on the role to which they are assigned. All roles created with this command have a resource type of Custom.

Syntax

This command has the following syntax:

- **Linux** | **UNIX** tipcli.sh AddRole --username *DASH_username* --password *DASH_user_password* --roleName *role_name*
- **Windows** tipcli.bat AddRole --username *DASH_username* --password *DASH_user_password* --roleName *role_name*

Where:

DASH_username is the portal administrator user ID.

DASH_user_password is the password associated with the portal administrator user ID.

role_name is the name of the role to be added.

Note: Arguments to the **rolesList** parameter must not include spaces.

Example

Linux | **UNIX** For example, in a UNIX or Linux environment, use the following command:

```
/opt/IBM/JazzSM/ui/bin/tipcli.sh AddRole --username DASH_username --password DASH_user_password --roleName role_name
```

Where */opt/IBM/JazzSM/* is location of the Dashboard Application Service Hub instance involved.

UpdateRole

Use the **UpdateRole** command to change the name of a custom role.

Syntax

This command has the following syntax:

- **Linux** | **UNIX** `tipcli.sh UpdateRole --username DASH_username --password DASH_user_password --roleName role_name --newRoleName new_role_name`
- **Windows** `tipcli.bat UpdateRole --username DASH_username --password DASH_user_password --roleName role_name --newRoleName new_role_name`

Where:

DASH_username is the portal administrator user ID.

DASH_user_password is the password associated with the portal administrator user ID.

role_name is the name of the role to be modified.

new_role_name is the new name you want for the specified role.

Note: Arguments to the **role_name** and **newRoleName** parameters must not include spaces.

Example

Linux | **UNIX** For example, in a UNIX or Linux environment, use the following command:

```
/opt/IBM/JazzSM/ui/bin/tipcli.sh UpdateRole --username DASH_username --password DASH_user_password --roleName role_name --newRoleName new_role_name
```

Where */opt/IBM/JazzSM/* is location of the Dashboard Application Service Hub instance involved.

DelRole

Use the **DelRole** command to delete a custom role.

Syntax

This command has the following syntax:

- **Linux** | **UNIX** `tipcli.sh DelRole --username DASH_username --password DASH_user_password --roleName role_name`
- **Windows** `tipcli.bat DelRole --username DASH_username --password DASH_user_password --roleName role_name`

Where:

DASH_username is the portal administrator user ID.

DASH_user_password is the password associated with the portal administrator user ID.

role_name is the name of the role to be modified.

Example

Linux | **UNIX** For example, in a UNIX or Linux environment, use the following command:

```
/opt/IBM/JazzSM/ui/bin/tipcli.sh DelRole --username DASH_username --password DASH_user_password --roleName role_name
```


Where `/opt/IBM/JazzSM/` is location of the Dashboard Application Service Hub instance involved.

ListRolesFromGroup

Use the **ListRolesFromGroup** command to list all roles associated with a specified user group.

Syntax

This command has the following syntax:

- **Linux** | **UNIX** `tipcli.sh ListRolesFromGroup --username DASH_username --password DASH_user_password --groupID group_ID`
- **Windows** `tipcli.bat ListRolesFromGroup --username DASH_username --password DASH_user_password --groupID group_ID`

Where:

DASH_username is the portal administrator user ID.

DASH_user_password is the password associated with the portal administrator user ID.

group_ID is the name of the user group associated with the roles that you want to list.

Example

Linux | **UNIX** For example, in a UNIX or Linux environment, use the following command:

```
/opt/IBM/JazzSM/ui/bin/tipcli.sh ListRolesFromGroup --username DASH_username --password DASH_user_password --groupID group_ID
```

Where `/opt/IBM/JazzSM/` is location of the Dashboard Application Service Hub instance involved.

MapRolesToGroup

Use the **MapRolesToGroup** command to associate a comma-separated list of roles to a specified user group.

Syntax

This command has the following syntax:

- **Linux** | **UNIX** `tipcli.sh MapRolesToGroup --username DASH_username --password DASH_user_password --groupID group_ID --rolesList role_name1, role__name2`
- **Windows** `tipcli.bat MapRolesToGroup --username DASH_username --password DASH_user_password --groupID group_ID --rolesList role_name1, role__name2`

Where:

DASH_username is the portal administrator user ID.

DASH_user_password is the password associated with the portal administrator user ID.

group_ID is the name of the user group associated with the roles that you want to map.

role_name1, role__name2 is a comma-separated list of roles that are to be associated with the specified user group.

Note: Individual role name arguments to the **rolesList** parameter must not include spaces.

Example

Linux | **UNIX** For example, in a UNIX or Linux environment, use the following command:

```
/opt/IBM/JazzSM/ui/bin/tipcli.sh MapRolesToGroup --username DASH_username --password DASH_user_password --groupID group_ID --rolesList role_name1, role_name2
```

Where */opt/IBM/JazzSM/* is location of the Dashboard Application Service Hub instance.

RemoveRolesFromGroup

Use the **RemoveRolesFromGroup** command to disassociate a comma-separated list of roles from a specified user group.

Syntax

This command has the following syntax:

- **Linux** **UNIX** `tipcli.sh RemoveRolesFromGroup --username DASH_username --password DASH_user_password --groupID group_ID --rolesList role_name1, role_name2`
- **Windows** `tipcli.bat RemoveRolesFromGroup --username DASH_username --password DASH_user_password --groupID group_ID --rolesList role_name1, role_name2`

Where:

DASH_username is the portal administrator user ID.

DASH_user_password is the password associated with the portal administrator user ID.

group_ID is the name of the user group from which roles are to be removed.

role_name1, role_name2 is a comma-separated list of roles to be removed from the user group.

Note: Individual role name arguments to the **rolesList** parameter must not include spaces.

Example

Linux **UNIX** For example, in a UNIX or Linux environment, use the following command:

```
/opt/IBM/JazzSM/ui/bin/tipcli.sh RemoveRolesFromGroup --username DASH_username --password DASH_user_password --groupID group_ID --rolesList role_name1, role_name2
```

Where */opt/IBM/JazzSM/* is location of the Dashboard Application Service Hub instance involved.

ListRolesForPage

Use the **ListRolesForPage** command to list all roles associated with a specified page.

Syntax

This command has the following syntax:

- **Linux** **UNIX** `tipcli.sh ListRolesForPage --pageUniqueName page_unique_name`
- **Windows** `tipcli.bat ListRolesForPage --pageUniqueName page_unique_name`

Where:

page_unique_name is the unique ID for the page.

Example

Linux **UNIX** For example, in a UNIX or Linux environment, use the following command:

```
/opt/IBM/JazzSM/ui/bin/tipcli.sh ListRolesForPage --pageUniqueName  
page_unique_name
```

Where */opt/IBM/JazzSM/* is location of the Dashboard Application Service Hub instance.

MapRolesToPage

Use the **MapRolesToPage** command to associate a comma-separated list of roles with a specified page and set an access level for each role.

Syntax

This command has the following syntax:

- **Linux** | **UNIX** `tipcli.sh MapRolesToPage --username DASH_username --password DASH_user_password --pageUniqueName page_unique_name --rolesList role_name1, role__name2 --accessLevelList level1, level2`
- **Windows** `tipcli.bat MapRolesToPage --username DASH_username --password DASH_user_password --pageUniqueName page_unique_name --rolesList role_name1, role__name2 --accessLevelList level1, level2`

Where:

DASH_username is the portal administrator user ID.

DASH_user_password is the password associated with the portal administrator user ID.

page_unique_name is the page ID with which to associate with the list of roles.

role_name1, role__name2 is a comma-separated list of roles that are to be associated with the page.

level1, level2 is a comma-separated list of page access levels that relate to the list of specified roles.

Each of the listed roles is assigned the access level that corresponds to its position in each list. For example, the second argument in the list associated with **rolesList** is assigned to the second argument associated **accessLevelList**.

Note: Individual role name arguments to the **rolesList** parameter must not include spaces.

Example

Linux | **UNIX** For example, in a UNIX or Linux environment, use the following command:

```
/opt/IBM/JazzSM/ui/bin/tipcli.sh MapRolesToPage --username DASH_username --  
password DASH_user_password --pageUniqueName page_unique_name --rolesList  
role_name1, role__name2 --accessLevelList level1, level2
```

Where */opt/IBM/JazzSM/* is location of the Dashboard Application Service Hub instance.

RemoveRolesFromPage

Use the **RemoveRolesFromPage** command to disassociate a comma-separated list of roles with a specified page.

Syntax

This command has the following syntax:

- **Linux** | **UNIX** `tipcli.sh RemoveRolesFromPage --username DASH_username --password DASH_user_password --pageUniqueName page_unique_name --rolesList role_name1, role__name2`
- **Windows** `tipcli.bat RemoveRolesFromPage --username DASH_username --password DASH_user_password --pageUniqueName page_unique_name --rolesList role_name1, role__name2`

Where:

DASH_username is the portal administrator user ID.

DASH_user_password is the password associated with the portal administrator user ID.

page_unique_name is the page ID associated with the roles that you want to remove.

role_name1, role__name2 is a comma-separated list of roles that are to be disassociated with the page.

Note: Individual role name arguments to the **rolesList** parameter must not include spaces.

Example

Linux

UNIX

For example, in a UNIX or Linux environment, use the following command:

```
/opt/IBM/JazzSM/ui/bin/tipcli.sh RemoveRolesFromPage --username DASH_username
--password DASH_user_password --pageUniqueName page_unique_name --rolesList
role_name1, role__name2 --accessLevelList level1, level2
```

Where */opt/IBM/JazzSM/* is location of the Dashboard Application Service Hub instance.

ListRolesForPortletEntity

Use the **ListRolesForPortletEntity** command to list all roles associated with a specified portlet.

Syntax

This command has the following syntax:

- **Linux** **UNIX** `tipcli.sh ListRolesForPortletEntity --portletEntityUniqueName portlet_entity_unique_name`
- **Windows** `tipcli.bat ListRolesForPortletEntity --portletEntityUniqueName portlet_entity_unique_name`

Where:

portlet_entity_unique_name is the unique ID for the portlet.

Example

Linux

UNIX

For example, in a UNIX or Linux environment, use the following command:

```
/opt/IBM/JazzSM/ui/bin/tipcli.sh ListRolesForPortletEntity --
portletEntityUniqueName portlet_entity_unique_name
```

Where */opt/IBM/JazzSM/* is location of the Dashboard Application Service Hub instance.

MapRolesToPortletEntity

Use the **MapRolesToPortletEntity** command to associate a comma-separated list of roles with a specified portlet.

Syntax

This command has the following syntax:

- **Linux** **UNIX** `tipcli.sh MapRolesToPortletEntity --username DASH_username --password DASH_user_password --portletEntityUniqueName portlet_entity_unique_name --rolesList role_name1, role__name2 --accessLevelList level1, level2`
- **Windows** `tipcli.bat MapRolesToPortletEntity --username DASH_username --password DASH_user_password --portletEntityUniqueName`

```
portlet_entity_unique_name --rolesList role_name1, role__name2 --
accessLevelList level1, level2
```

Where:

DASH_username is the portal administrator user ID.

DASH_user_password is the password associated with the portal administrator user ID.

portlet_entity_unique_name is the unique portlet ID with which to associate with the list of roles.

role_name1, role__name2 is a comma-separated list of roles that are to be associated with the portlet.

level1, level2 is a comma-separated list of access levels that relate to the list of specified roles. Each of the listed roles is assigned the access level that corresponds to its position in each list. For example, the second argument in the list associated with **rolesList** is assigned to the second argument associated **accessLevelList**.

Note: Individual role name arguments to the **rolesList** parameter must not include spaces.

Example

Linux

UNIX

For example, in a UNIX or Linux environment, use the following command:

```
/opt/IBM/JazzSM/ui/bin/tipcli.sh MapRolesToPortletEntity --username
DASH_username --password DASH_user_password --portletEntityUniqueName
portlet_entity_unique_name --rolesList role_name1, role__name2 --
accessLevelList level1, level2
```

Where */opt/IBM/JazzSM/* is location of the Dashboard Application Service Hub instance.

RemoveRolesFromPortletEntity

Use the **RemoveRolesFromPortletEntity** command to disassociate a comma-separated list of roles with a specified portlet.

Syntax

This command has the following syntax:

- **Linux** **UNIX** `tipcli.sh RemoveRolesFromPortletEntity --username DASH_username --password DASH_user_password --portletEntityUniqueName portlet_entity_unique_name --rolesList role_name1, role__name2`
- **Windows** `tipcli.bat RemoveRolesFromPortletEntity --username DASH_username --password DASH_user_password --portletEntityUniqueName portlet_entity_unique_name --rolesList role_name1, role__name2`

Where:

DASH_username is the portal administrator user ID.

DASH_user_password is the password associated with the portal administrator user ID.

portlet_entity_unique_name is the portlet ID associated with the roles that you want to remove.

role_name1, role__name2 is a comma-separated list of roles that are to be disassociated with the portlet.

Note: Individual role name arguments to the **rolesList** parameter must not include spaces.

Example

Linux

UNIX

For example, in a UNIX or Linux environment, use the following command:

```
/opt/IBM/JazzSM/ui/bin/tipcli.sh RemoveRolesFromPortletEntity --username
DASH_username --password DASH_user_password --portletEntityUniqueName
portlet_entity_unique_name --rolesList role_name1, role__name2
```

Where `/opt/IBM/JazzSM/` is location of the Dashboard Application Service Hub instance.

ListRolesFromUser

Use the **ListRolesFromUser** command to list all roles associated with a specified user.

Syntax

This command has the following syntax:

- **Linux** **UNIX** `tipcli.sh ListRolesFromUser --username DASH_username --password DASH_user_password --userID user_ID`
- **Windows** `tipcli.bat ListRolesFromUser --username DASH_username --password DASH_user_password --userID user_ID`

Where:

DASH_username is the portal administrator user ID.

DASH_user_password is the password associated with the portal administrator user ID.

user_ID is the unique ID for the user.

Example

Linux **UNIX** For example, in a UNIX or Linux environment, use the following command:

```
/opt/IBM/JazzSM/ui/bin/tipcli.sh ListRolesFromUser --username DASH_username --password DASH_user_password --userID user_ID
```

Where `/opt/IBM/JazzSM/` is location of the Dashboard Application Service Hub instance.

MapRolesToUser

Use the **MapRolesToUser** command to associate a comma-separated list of roles with a specified user ID.

Syntax

This command has the following syntax:

- **Linux** **UNIX** `tipcli.sh MapRolesToUser --username DASH_username --password DASH_user_password --userID user_ID --rolesList role_name1, role__name2`
- **Windows** `tipcli.bat MapRolesToUser --username DASH_username --password DASH_user_password --userID user_ID --rolesList role_name1, role__name2`

Where:

DASH_username is the portal administrator user ID.

DASH_user_password is the password associated with the portal administrator user ID.

user_ID is the unique user ID with which to associate with the list of roles.

role_name1, role__name2 is a comma-separated list of roles that are to be associated with the user.

Note: Individual role name arguments to the **rolesList** parameter must not include spaces.

Example

Linux **UNIX** For example, in a UNIX or Linux environment, use the following command:

```
/opt/IBM/JazzSM/ui/bin/tipcli.sh MapRolesToUser --username DASH_username --password DASH_user_password --userID user_ID --rolesList role_name1, role_name2
```

Where */opt/IBM/JazzSM/* is location of the Dashboard Application Service Hub instance.

RemoveRolesFromUser

Use the **RemoveRolesFromUser** command to disassociate a comma-separated list of roles with a specified user ID.

Syntax

This command has the following syntax:

- **Linux** | **UNIX** `tipcli.sh RemoveRolesFromUser --username DASH_username --password DASH_user_password --userID user_ID --rolesList role_name1, role_name2`
- **Windows** `tipcli.bat RemoveRolesFromUser --username DASH_username --password DASH_user_password --userID user_ID --rolesList role_name1, role_name2`

Where:

DASH_username is the portal administrator user ID.

DASH_user_password is the password associated with the portal administrator user ID.

user_ID is the user ID associated with the roles that you want to remove.

role_name1, role_name2 is a comma-separated list of roles that are to be disassociated with the specified user ID.

Note: Individual role name arguments to the **rolesList** parameter must not include spaces.

Example

Linux | **UNIX** For example, in a UNIX or Linux environment, use the following command:

```
/opt/IBM/JazzSM/ui/bin/tipcli.sh RemoveRolesFromUser --username DASH_username --password DASH_user_password --userID user_ID --rolesList role_name1, role_name2
```

Where */opt/IBM/JazzSM/* is location of the Dashboard Application Service Hub instance.

ListRolesForView

Use the **ListRolesForView** command to list all roles associated with a specified view.

Syntax

This command has the following syntax:

- **Linux** | **UNIX** `tipcli.sh ListRolesForView --viewUniqueName view_name`
- **Windows** `tipcli.bat ListRolesForView --viewUniqueName view_name`

Where:

view_name is the unique name for the view.

Example

Linux | **UNIX** For example, in a UNIX or Linux environment, use the following command:

```
/opt/IBM/JazzSM/ui/bin/tipcli.sh ListRolesForView --viewUniqueName view_name
```

Where */opt/IBM/JazzSM/* is location of the Dashboard Application Service Hub instance.

MapRolesToView

Use the **MapRolesToView** command to associate a comma-separated list of roles with a specified view and set an access level for each role.

Syntax

This command has the following syntax:

- **Linux** | **UNIX** `tipcli.sh MapRolesToView --username DASH_username --password DASH_user_password --viewUniqueName view_name --rolesList role_name1, role_name2 --accessLevelList level1, level2`
- **Windows** `tipcli.bat MapRolesToView --username DASH_username --password DASH_user_password --viewUniqueName view_name --rolesList role_name1, role_name2 --accessLevelList level1, level2`

Where:

DASH_username is the portal administrator user ID.

DASH_user_password is the password associated with the portal administrator user ID.

view_name is the unique view name with which to associate with the list of roles.

role_name1, role_name2 is a comma-separated list of roles that are to be associated with the view.

level1, level2 is a comma-separated list of page access levels that relate to the list of specified roles.

Each of the listed roles is assigned the access level that corresponds to its position in each list. For example, the second argument in the list associated with **rolesList** is assigned to the second argument associated **accessLevelList**.

Note: Individual role name arguments to the **rolesList** parameter must not include spaces.

Example

Linux | **UNIX** For example, in a UNIX or Linux environment, use the following command:

```
/opt/IBM/JazzSM/ui/bin/tipcli.sh MapRolesToView --username DASH_username --password DASH_user_password --viewUniqueName view_name --rolesList role_name1, role_name2 --accessLevelList level1, level2
```

Where */opt/IBM/JazzSM/* is location of the Dashboard Application Service Hub instance.

RemoveRolesFromView

Use the **RemoveRolesFromView** command to disassociate a comma-separated list of roles with a specified view.

Syntax

This command has the following syntax:

- **Linux** | **UNIX** `tipcli.sh RemoveRolesFromView --username DASH_username --password DASH_user_password --viewUniqueName view_name --rolesList role_name1, role_name2`
- **Windows** `tipcli.bat RemoveRolesFromView --username DASH_username --password DASH_user_password --viewUniqueName view_name --rolesList role_name1, role_name2`

Where:

DASH_username is the portal administrator user ID.

DASH_user_password is the password associated with the portal administrator user ID.

view_name is the unique view name associated with the roles that you want to remove.
role_name1, role_name2 is a comma-separated list of roles that are to be disassociated with the specified view.

Note: Individual role name arguments to the **rolesList** parameter must not include spaces.

Example

Linux **UNIX** For example, in a UNIX or Linux environment, use the following command:

```
/opt/IBM/JazzSM/ui/bin/tipcli.sh RemoveRolesFromView --username DASH_username  
--password DASH_user_password --viewUniqueName view_name --rolesList  
role_name1, role_name2
```

Where */opt/IBM/JazzSM/* is location of the Dashboard Application Service Hub instance.

Working with views

tipcli commands for working with views.

The tipcli commands are entered in the *tip_home_dir/profiles/TIPProfile/bin* directory, for example, *C:\Program Files\IBM\JazzSM\ui\bin\tipcli.bat* on Windows or */opt/IBM/JazzSM/ui/bin/tipcli.sh* on Linux or UNIX.

ListViews

List all views.

AddViewMembers --username *DASH_username* --password *DASH_user_password* --view *view_unique_name* [--members *members1, member2*] [--launchMembers *launch_member1, launch_member2*]

Add members or launch members for a specified view.

Important: When you add members to a view at the command line, your updates are not reflected in the portal until the next time that you log in.

ListViewsForRole --roleName *role_name*

List the views associated with a specified role.

MapViewstoRole --username *DASH_username* --password *DASH_user_password* --roleName *role_name* --viewList *view_unique_name1, view_unique_name2* --accessLevelList *level1, level2*

Associate a comma separated list of views with a particular role and set the access level for the role for each view.

RemoveViewsFromRole --username *DASH_username* --password *DASH_user_password* --roleName *role_name* --viewList *view_unique_name1, view_unique_name2*

Disassociate a comma separated list of views from a particular role.

Working with users

tipcli commands for working with users.

ListUsersFromRole --roleName *role_name*

List the users associated with a specified role.

MapUsersToRole --username *DASH_username* --password *DASH_user_password* --roleName *role_name* --usersList *user_ID1:user_ID2*

Associate a colon (:) separated list of user IDs with a particular role.

Note: Arguments to the *usersList* parameter should not include a colon (:).

RemoveUsersFromRole --username *DASH_username* --password *DASH_user_password* --roleName *role_name* --usersList *user_ID1:user_ID2*

Disassociate a colon (:) separated list of user IDs from a particular role.

Working with preference profiles

tipcli commands for working with preference profiles.

DeletePreferenceProfile `--username DASH_username --password DASH_user_password --profileName profile_name`

Delete the specified preference profile.

ListPreferenceProfiles `[--name profile_name]`

Return a list of console preference profiles. Optionally, you can specify a comma separated lists of preference profiles, to return their unique names.

ShowPreferenceProfile `--uniqueName profile_unique_name`

List all the attributes for a specified profile preference.

AddUpdatePreferenceProfile `--username DASH_username --password DASH_user_password --profileName profile_name [--newProfileName new_profile_name] [--themeDir theme_dir] [--showNavTree true|false] [--componentDir default|ltr|rtl] [--textDir default|contextual|ltr|rtl] [--views view_unique_name1, view_unique_name2] --roles role_name1, role_name2] [--defaultView view_unique_name]`

Use the AddUpdatePreferenceProfile command to create a new profile preference or update an existing profile.

Parameter and arguments	Description
<code>--username DASH_username</code>	Mandatory parameter. A user with the <code>iscadmins</code> role.
<code>--password DASH_user_password</code>	Mandatory parameter. The password for the user with the <code>iscadmins</code> role.
<code>--profileName profile_name</code>	Mandatory parameter. The name of the profile that is to be created or modified.
<code>[--newProfileName new_profile_name]</code>	Optional parameter. The new name for the specified profile.
<code>[--themeDir theme_dir]</code>	Optional parameter. Used to specify the directory for the theme that you want to apply.
<code>[--showNavTree true false]</code>	Optional parameter. Used to specify whether or not you want the navigation pane to be displayed for preference profile.
<code>[--componentDir default ltr rtl]</code>	Optional parameter. Used to specify component display direction, that is, whether you want items to display left-to-right, right-to-left, or to use the default browser settings.
<code>[--textDir default ltr rtl]</code>	Optional parameter. Used to specify text direction, that is, whether you want text to display left-to-right, right-to-left, or to use the default browser settings.
<code>[--views view_unique_name1, view_unique_name2]</code>	Optional parameter. Used to specify the views that you want to assign to the preference profile. Comma separated list.
<code>--roles role_name1, role_name2]</code>	Optional parameter. Used to specify the roles that you want to assign to the preference profile. Comma separated list.

Table 15. AddUpdatePreferenceProfile command arguments (continued)	
Parameter and arguments	Description
<code>[--defaultView view_unique_name]</code>	Optional parameter. Used to specify the view that you want displayed when a user logs into the portal.

Working with portlets

tipcli commands for working with portlets.

The tipcli commands are entered in the `tip_home_dir/profiles/TIPProfile/bin` directory, for example, `C:\Program Files\IBM\JazzSM\ui\bin\tipcli.bat` on Windows or `/opt/IBM/JazzSM/ui/bin/tipcli.sh` on Linux or UNIX.

ListPortletEntitiesForRole --roleName role_name]

List the portlets entities associated with a specified role.

MapPortletEntitiesToRole --username DASH_username --password DASH_user_password --roleName role_name --portletEntityList portletEntity_unique_name1, portletEntity_unique_name2 --accessLevelList level1, level2

Associate a comma separated list of portlets with a particular role and set the access level for the role for each portlet.

RemovePortletEntitiesFromRole --username DASH_username --password DASH_user_password --roleName role_name --portletEntityList portletEntity_unique_name1, portletEntity_unique_name2

Disassociate a comma separated list of portlets with from particular role.

Working with pages

tipcli commands for working with pages.

ListPages [--viewList view_unique_name1, view_unique_name2] [--customizePages true|false]

List all pages. You can optionally filter the list by using the `viewlist` parameter and providing a comma separated list of views. You can also use the `customizePages` (set to `true`) to return a list of custom pages only.

ListPagesForRole --roleName role_name

List the pages associated with a specified role.

MapPagesToRole --username DASH_username --password DASH_user_password --roleName role_name --pageList page_unique_name1, page_unique_name2 --accessLevelList level1, level2

Associate a comma separated list of pages with a particular role and set the access level for the role for each page.

RemovePagesFromRole --username DASH_username --password DASH_user_password --roleName role_name --pageList page_unique_name1, page_unique_name2

Disassociate a comma separated list of pages from a particular role.

Working with user groups

tipcli commands for working with user groups.

The tipcli commands are entered in the `/opt/IBM/JazzSM/ui/bin` directory, for example, `C:\Program Files\IBM\JazzSM\ui\bin\tipcli.bat` on Windows or `/opt/IBM/JazzSM/ui/bin/tipcli.sh` on Linux or UNIX.

ListGroupsFromRole --roleName role_name

List the user groups associated with a specified role.

MapGroupsToRole --username DASH_username --password DASH_user_password --roleName role_name --groupsList group_name1: group_name2

Associate a colon (:) separated list of groups with a particular role.

Note: Arguments to the groupsList parameter should not include a colon (:).

RemoveGroupsFromRole --username *DASH_username* --password *DASH_user_password* --roleName *role_name* --groupsList *group_name1: group_name2*
 Disassociate a colon (:) separated list of groups from a particular role.

Charting *tipcli* commands

tipcli commands for working with charting.

ListCharts --username *DASH_username* --password *DASH_user_password*
 Use ListCharts to review the charts that are configured in the environment.

ChartConnection --action *action* [--name *name*] [--protocol *protocol* --hostname *hostname* --port *port* --serviceName *serviceName* --username *username* --password *password* --renderFormat *render_format* --DataSource_Username *datasource_username* --credentialType *credential_type*] --username *DASH_username* --password *DASH_user_password*

ChartConnection is used to configure a connection to any IBM Tivoli Charting Web Service. The ITM Web Service is just one example.

ChartExport --dir *output_directory* --type *all|customcharts|page* [--pageID *page_ID* | --pageName *page_name*] --username *DASH_username* --password *DASH_user_password*

ChartExport is used to export chart data.

<i>Table 16. ChartExport command arguments</i>	
Parameter and arguments	Description
--dir <i>output_directory</i>	Mandatory parameter. The directory where the exported data is saved. If the directory does not exist, it is created.
--type <i>all customcharts page</i>	Mandatory parameter. If you set the --type to <i>all</i> , then all charts are exported. If you set it to <i>customcharts</i> , then only customized charts are exported. If you set it to <i>page</i> , then you can use either the --pageID or the --pageName parameter to specify the page for which you want to export chart data.
[--pageID <i>page_ID</i> --pageName <i>page_name</i>]	Optional parameter. If you set the --type parameter to <i>page</i> , then you can use either the --pageID or the --pageName parameter to specify the page for which you want to export chart data.
--username <i>DASH user name</i>	Mandatory parameter. The user name for a user with either the <i>chartAdministrator</i> or <i>chartCreator</i> role.
--password <i>DASH user password</i>	Mandatory parameter. The password for the specified user name.

ChartImport --dir *source_directory* --username *DASH_username* --password *DASH_user_password*

ChartImport is used to import chart data from a specified directory.

<i>Table 17. ChartImport command arguments</i>	
Parameter and arguments	Description
<code>--dir source_directory</code>	Mandatory parameter. The directory where the data to be imported is located. BIRT Designer file format is .rptdesign.
<code>--username DASH user name</code>	Mandatory parameter. The user name for a user with either the chartAdministrator or chartCreator role.
<code>--password DASH user password</code>	Mandatory parameter. The password for the specified user name.

ChartProperties [`--name property_name --value property_value`] `--username DASH_username --password DASH_user_password`

ChartProperties is used to view or modify properties for charting. If you only provide username and password details and no other arguments, then the current properties are listed. It is useful to run this command first so that you can review the current property names and values before you decide to make updates.

<i>Table 18. ChartProperties command arguments</i>	
Parameter and arguments	Description
<code>--name property_name --value property_value</code>	Optional parameter. The name of the property that you want to update and the value that you want to set. For example, to set the timeout value to 10,000,000 milliseconds, enter <code>--name AXIS_TIMEOUT --value 10000000</code> .
<code>--username DASH user name</code>	Mandatory parameter. The user name for a user with the chartAdministrator role.
<code>--password DASH user password</code>	Mandatory parameter. The password for the specified user name.

ListRestoreTimestamp

Use the ListRestoreTimestamp command to return a list of charting store backups by timestamp.

RestoreChartStore `--BackupTimestamp backup_timestamp --username DASH_username --password DASH_user_password`

Use the RestoreChartStore command to restore a chart store by timestamp.

<i>Table 19. RestoreChartStore command arguments</i>	
Parameter and arguments	Description
<code>RestoreChartStore --BackupTimestamp</code>	Mandatory parameter. The timestamp of the charting store backup.
<code>--username DASH user name</code>	Mandatory parameter. The user name for a user with the chartAdministrator role.
<code>--password DASH user password</code>	Mandatory parameter. The password for the specified user name.

Dashboard Application Service Hub Export commands

Use these tipcli commands for to export Dashboard Application Service Hub customized data.

tipcli - Export plugins

Use the Export command to export customization data for an instance of Dashboard Application Service Hub. Use the ListExportPlugins command to list plugins that are available for export.

Syntax

ListExportPlugins

Use the ListExportPlugins command to list all plugins that can be exported. Use the list of returned plugins to assist you when you are specifying plugins to be exported.

Export [--includePlugins|--excludePlugins *plugin1,plugin2*] [--settingFile *setting_file*] --username *DASH_username* --password *DASH_user_password*

Parameters

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

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

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

Example 1 - Return a list of plugins available for exporting

The following example returns a list of plugins that can be exported:

```
Windows C:\Program Files\IBM\JazzSM\ui\bin>tipcli.bat ListExportPlugins
```

Example 2 - Export a subset of available plugins

The following example exports the CMS plugin only:

```
Windows C:\Program Files\IBM\JazzSM\ui\bin>tipcli.bat Export
--includePlugins com.ibm.tivoli.tip.cli.cms.CmsExportPlugin
--username tbsmadmin --password DASHpassword
```

tipcli - Advanced Export options

Use the ExportPagePlugin *tipcli* command to export specific Dashboard Application Service Hub data.

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

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

Parameter and arguments	Description
<code>--exportFile <i>export_file</i></code>	Optional parameter. Specifies the path and file name for the exported data, for example, <code>c:/tmp/extest.zip</code> .
<code>--pages ALL NONE <i>page1,page2</i></code>	Optional parameter. If you do not use the pages parameter, the default setting is ALL unless either <code>exportPagesInViews</code> or <code>includeEntitiesFromApp</code> is defined, then the default setting is NONE. You can also provide a list of pages that you want to export.
<code>--views ALL NONE <i>view1,view2</i></code>	Optional parameter. If you do not use the views parameter, the default setting is ALL. You can also provide a list of views that you want to export and optionally specify that you want to export all pages associated with the specified views. Note: Whether the optional parameter <code>exportpageinviews</code> is set to <code>true</code> or <code>false</code> , if a view has a default node in the navigation pane associated with it, then the page associated with the node is always exported. This is also true, even if you specify NONE as the argument to the <code>--pages</code> parameter.
<code>--roles ALL NONE REQUIRED <i>role1,role2</i></code>	Optional parameter. You can export no roles, all roles, or a specific list of roles. The default setting is ALL unless the pages parameter or the <code>includeEntitiesFromApp</code> parameter is specified. Then, the default setting is set to REQUIRED.

<i>Table 21. ExportPagePlugin command arguments (continued)</i>	
Parameter and arguments	Description
<code>[--exportPagesInViews true false]</code>	Optional parameter. Use this parameter, set to <code>true</code> , to export the pages associated with an exported view . The default value is <code>false</code> .
<code>[--userPreferences ALL NONE REQUIRED user_ID1,user_ID2]</code>	Optional parameter. You can export preferences for all users, no users, or for a specified list of users by user ID. The default setting is <code>ALL</code> . This parameter overrides the <code>includeMytasks</code> and <code>includeMyStartupPages</code> parameters.
<code>[--consolePreferenceProfiles ALL NONE pref_ID1,pref_ID2]</code>	Optional parameter. You can export no preference profile data, all preference profile data, or data for a specific list of preference profiles. The default setting is <code>ALL</code> . Note: If a console preference profile has a custom view as its default view, then that view is automatically exported. If the exported view has a default node in the navigation pane, then the associated page is automatically exported with the view.
<code>[--includeEntitiesFromApp war1,war2]</code>	Optional parameter. You can provide a list of WARs to export pages that contain portlets associated with the listed WARs.
<code>[--includeCustomData true false]</code>	Optional parameter. The default value is <code>true</code> . If is set to <code>false</code> , no customization data is exported.
<code>[--includeCredentialData true false]</code>	Optional parameter. The default value is <code>true</code> . If is set to <code>false</code> , no credential data is exported.
<code>[--includeMytasks true false]</code>	Optional parameter. The default setting is <code>true</code> . This parameter only applies when the <code>includeEntitiesFromApp</code> parameter is also specified.
<code>[--includeMyStartupPages true false]</code>	Optional parameter. The default setting is <code>true</code> . This parameter only applies when the <code>includeEntitiesFromApp</code> parameter is also specified.
<code>[--includeTransformations true false]</code>	Optional parameter. The default setting is <code>true</code> .
<code>--username DASH user name</code>	Mandatory parameter. The user name for a user with the <code>iscadmins</code> role.
<code>--password DASH user password</code>	Mandatory parameter. The password for the specified user name.

tipcli - Charting Export options

Use the `ChartExportPlugin tipcli` command to export Dashboard Application Service Hub chart data.

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

Export [--includeCharts ALL|NONE|page_ID1,page_ID2] --username DASH_username --password DASH_user_password

Table 22. ChartExportPlugin command arguments	
Parameter and arguments	Description
[--includeCharts ALL NONE page_ID1,page_ID2]	Optional parameter. You can export all charts, no charts, or specify a list of charts to be exported. The default setting is ALL. Note: If you run the Export command using the --includeCharts parameter, it must be run by the same user that started the Dashboard Application Service Hub Server.
--username DASH user name	Mandatory parameter. The user name for a user with the chartAdministrator role.
--password DASH user password	Mandatory parameter. The password for the specified user name.

Import tipcli commands

tipcli commands for importing Dashboard Application Service Hub data.

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

ListImportPlugins

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

Import [--includePlugins|--excludePlugins plugin1,plugin2] [--settingFile setting_file] [--backupDir backup_dir] --username DASH_username --password DASH_user_password

Use the Import command to import customization data into a Dashboard Application Service Hub environment. If you provide no parameters to the Import command, all custom data is imported by default.

Table 23. Import command arguments	
Parameter and arguments	Description
[--includePlugins --excludePlugins plugin1,plugin2]	Optional parameter. You can choose to include or exclude a list of plugins when you run the Import command.
[--settingFile setting_file]	Optional parameter. You can specify your import requirements in a properties file instead of specifying your requirements using separate parameters at the command line. Provide a path to the settings file as the argument to the settingFile parameter. On systems running Windows you must use double backslashes characters (\\) when specifying the path to your settings file, for example, C:\\tmp\\import.properties. Command line parameters take precedence over entries in the settings file.
[--backupDir backup_dir]	You can specify a directory to save the backup data during an import operation so that if it is required you can subsequently restore settings.

<i>Table 23. Import command arguments (continued)</i>	
Parameter and arguments	Description
<code>--username DASH user name</code>	Mandatory parameter. The user name for a user with the <code>iscadmin</code> role.
<code>--password DASH user password</code>	Mandatory parameter. The password for the specified user name.

Additional commands

Additional `tipcli` commands.

cmsUpdateRemoteEntries [`--username username --password password`] (`-toremove` | `-fromremote` | `-deleteremote`) [`-force`]

Save system information in the file specified.

<i>Table 24. cmsUpdateRemoteEntries command arguments</i>	
Parameter and arguments	Description
[<code>--username username --password password</code>]	Optional parameters. User name and password for a Dashboard Application Service Hub user. If you do not provide user name and password details at the command line, you must enter the user name and password in an interactive mode.
<code>-toremove</code>	Optional parameter. Indicates that the update is to occur to the remote data store, that is, the local information is to be written to the remote database.
<code>-fromremote</code>	Optional parameter. Indicates that the update is to occur from the remote data store. Any information saved locally is downloaded and updated from the remote database.
<code>-deleteremote</code>	Optional parameter. Indicates that the launch entries provided by this Dashboard Application Service Hub instance to the remote database is to be deleted from the database. Additionally, this command prevents any further updates from being sent to the remote database. On execution, the <code>cmsUpdateRemoteEntries</code> command with the <code>toremove</code> and <code>force</code> options updates the database and re-enables automatic updates to the remote database. Note: There is no difference between <code>deleteremote</code> with the <code>force</code> option and <code>deleteremote</code> without the <code>force</code> option.
<code>-force</code>	Optional parameter. Indicates that any caching or optimization mechanisms for the data should be ignored and that the data should be updated regardless of the state. Any existing cached information is discarded. All data in the database is refreshed for the <code>toremove</code> case, including the resource bundles.

Version

List the versions of the products and components installed in the environment.

SystemInfo [--outputFile *outputFile*]

Save system information in the file specified.

ITMLogin --hostname *hostname* --port *port* --username *username* --password *password* [--servicename]

ITMLogin is used to configure the ITM Web Service to connect to the Tivoli Enterprise Portal Server. For example, this command in Windows configures the username and password for a new ITM Web Service to be added to the application server instance.

```
C:\IBM\tivoli\tip\bin\tipcli.bat ITMLogin --hostname
localhost --port 1920 --username sysadmin --password
sysadm1n --servicename ITMWebService2
```

You can use the ITMLogin command to change the hostname, port, username, and password of an existing Tivoli Enterprise Portal Server instance. Changing a configured ITM Web Service to a different Tivoli Enterprise Portal Server is not supported, because the two portal servers may have different configurations. If you need to use a different portal server, you can install another instance of the ITM Web Service and use this command (along with the -serviceName option) to configure.

TADDMLogin --hostname *hostname* [--port *port*] --username *username* --password *password*

Log in to the Tivoli Application Dependency Discovery Manager.

Appendix A. 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
224A/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.

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.

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.

Index

A

- about this profile [166](#), [209](#)
- adding JDBC drivers [54](#)
- administrator
 - default user [25](#)
- advanced commands [214](#)
- advanced installation [31](#)
- agent
 - Business Service Management [77](#)
 - configuring [77](#)
 - Discovery Library Toolkit Log attribute group [94](#)
 - disk capacity planning [94](#)
 - Event Broker Log attribute group [88](#)
 - installing [77](#)
 - Performance Object Status attribute group [86](#)
 - Service Indicators attribute group [89](#)
 - Service Status attribute group [90](#)
 - Service Status Change Event attribute group [91](#)
 - URL Monitor attribute group [93](#)
- agent for TBSM
 - workspaces [81](#)
- Agent Management Services
 - enabling [80](#)
- application server
 - ports [165](#), [208](#)
 - profile [166](#), [209](#)
- architecture [11](#)
- attribute groups
 - Business Service Management Agent [83](#)
- attributes
 - Business Service Management Agent [83](#)
 - Discovery Library Toolkit Log attribute group [94](#)
 - Event Broker Log [88](#)
 - Performance Object Status attribute group [86](#)
 - Service Indicators [89](#)
 - Service Status [90](#)
 - Service Status Change Event [91](#)
 - URL Monitor [93](#)
- authentication
 - user registry [23](#)

B

- back up
 - server settings [222](#)
- basic commands [212](#)
- books, *See* publications
- Business Service Management
 - agent [77](#)
 - installing agent [77](#)
- Business Service Management Agent
 - attributes [83](#)

C

- certificate [164](#)

- certificate authority
 - changing Secure Socket Layer (SSL) [148](#)
- certificates
 - signer data for dashboard servers [139](#)
- CGI support [222](#)
- changing password [210](#)
- ChartExportPlugin tipcli command
 - export [242](#)
- cloning
 - server settings [222](#)
- CMS
 - access [224](#)
 - configure hostname [206](#)
 - create remote database [205](#)
 - data source [204](#)
- commands
 - tbsm_db [38](#)
- configuring VMM [172](#)
- Context Menu Service
 - access [224](#)
- conventions
 - typeface [3](#)
- core library [1](#)

D

- dashboard server [62](#)
- Dashboard server
 - LDAP configuration [24](#)
- data fetchers
 - failover [108](#)
 - Time Window Analyzer [108](#)
- data server [63](#)
- Data server
 - configuring for failover [114](#)
 - installation worksheet [51](#)
- data source
 - failover configuration [108](#)
- data sources
 - failover [108](#)
- DB2
 - create TBSM schema [38](#)
 - install TBSM schema [31](#)
 - installing on UNIX [31](#)
 - Tivoli System Automation [111](#), [112](#)
- DBConfig [65](#)
- default groups [25](#)
- Discovery Library Toolkit
 - Log attribute group for agent [94](#)
 - log files [163](#)
 - requirements [110](#)

E

- education, *See* Tivoli technical training
- EIF probe
 - installing [43](#)

environment variables, notation [15](#)

error

MSIEXEC [155](#)

ETai [191](#)

Event Broker Log

attribute group for agent [88](#)

events

user permissions [25](#)

exporting

basic export console preference profiles [213](#)

basic export pages [212](#)

basic export views [213](#)

export all [214](#)

export pages [216](#)

export views [217](#)

rules [218](#), [220](#)

settings file [215](#)

ExportPagePlugin tipcli command

export [241](#)

F

failover

configuring Data servers [114](#)

custom canvases [108](#)

data source configuration [108](#)

limitations [108](#)

ObjectServer

communication configuration [113](#)

manual configuration [114](#)

overview [108](#)

service trees [108](#)

verification [118](#)

view definitions [108](#)

failover environment

setup requirements [110](#)

failover **starting servers** [117](#)

features

new for 6.2.0 [7](#)

federated repositories

VMM for ObjectServer [172](#)

files

logs [163](#)

sample response [163](#)

G

groups

default [25](#)

user [25](#)

H

historical data

disk capacity for agent [94](#)

hostname [223](#)

HTTP and HTTPS [202](#)

HTTP server

configuring [127](#), [181](#)

HTTP server plug-in SSL configuration

load balancing [108](#), [124](#), [128](#), [129](#), [132](#), [179](#), [183](#), [184](#), [186](#)

I

IBM Tivoli Monitoring

Agent Management Services [80](#)

IBM Tivoli Service Level Advisor

event forwarding requirements [110](#)

impact server details [44](#)

impact server installation [44](#)

importing

import data [219](#)

rollback [219](#)

install

response file [62](#), [63](#), [65](#)

silent [62](#), [63](#), [65](#)

installation

advanced [31](#)

command line [67](#)

console [67](#)

for single sign-on [173](#)

impact server [44](#)

notes [18](#)

silent [61](#)

simple [31](#)

installation user [18](#)

K

KR9_Process_Data_Unavailable [96](#)

KR9_TBSM_Critical [96](#)

KR9_TBSM_Web_App_Critical [97](#)

L

language support [99](#)

LDAP

adding [166](#)

configuring [168](#), [169](#)

Dashboard server [24](#)

SSL [169](#)

user registry consideration [22](#)

load balancing

clone IDs [128](#), [129](#), [183](#), [184](#)

server-to-server trust [124](#), [179](#)

load balancing cluster

join [126](#), [181](#)

log files

Discovery Library Toolkit [163](#)

login

configure for HTTP and HTTPS [202](#)

logon [164](#), [207](#)

M

manuals, *See* publications

migrating [101](#)

Monitor role

Context Menu Service [224](#)

MSIEXEC [155](#)

N

nameserver password

changing [210](#)

- Netcool/Impact
 - install on TBSM host [19](#)
 - installing on TBSM host [161](#)
- Netcool/OMNIBus
 - secure communications [143](#)
- Netcool/OMNIBus
 - triggers [21](#)
- notation
 - environment variables [15](#)
 - path names [15](#)
 - typeface [15](#)

O

- ObjectServer
 - failover
 - manual configuration [114](#)
 - permissions for viewing events [25](#)
 - SSL connection [171](#)
- online publications
 - accessing [2](#)
- operating system agents
 - IBM Tivoli Monitoring [80](#)
- ordering publications [3](#)
- overview [164](#)

P

- pages [237](#)
- password
 - change [209](#)
 - SSL [209](#)
- passwords
 - tipadmin [158](#)
- Performance Object
 - attribute group for agent [86](#)
- permissions
 - users and groups [25](#)
- policies
 - failover [108](#)
- port
 - numbers [166](#), [209](#)
- port assignments [165](#), [208](#)
- product library [1](#)
- publications
 - accessing online [2](#)
 - ordering [3](#)

R

- registry
 - default security [221](#)
 - user authentication [23](#)
- requirements
 - system [17](#)
- roles
 - user [25](#)

S

- secure communications
 - Netcool/OMNIBus [143](#)
- secure connections

- secure connections (*continued*)
 - certificates [138](#)
- Secure Socket Layer (SSL) certificate
 - adding signer certificate [150](#), [151](#)
 - adding signer exchange [152](#)
 - changing default [148](#)
 - creating certificate request [149](#)
 - receiving from certificate authority [149](#)
- security
 - certificate [164](#)
 - default registry [221](#)
- server
 - stopping or starting [208](#)
- server settings
 - cloning [222](#)
- Service Indicators
 - attribute group for agent [89](#)
- Service Status
 - attribute group for agent [90](#)
- Service Status Change Event
 - attribute group for agent [91](#)
- signer data for certificates [139](#)
- silent install [62](#), [63](#), [65](#)
- silent installation [61](#)
- simple installation [31](#)
- single sign-on
 - configuring [173](#)
 - ETai trust association [191](#), [192](#)
 - installing ETai [191](#)
- situations
 - Availability [96](#)
 - KR9_Process_Data_Unavailable [96](#)
 - KR9_TBSM_Critical [96](#)
 - KR9_TBSM_Web_App_Critical [97](#)
 - URL Monitor [97](#)
- SQL database DSA [54](#)
- SSH
 - installing TBSM on SUSE Linux [160](#)
- SSL
 - configuring [132](#), [170](#), [186](#)
 - HTTP server plug-in [132](#), [186](#)
 - SSL [170](#)
 - to ObjectServer [171](#)
 - stopping the application server [208](#)
- SUSE Linux
 - installing TBSM over SSH [160](#)

T

- TBSM
 - core library [1](#)
 - database requirements [110](#)
 - installation
 - firewall [161](#)
- TBSM agent
 - installing [77](#)
- TBSM groups
 - creating manually [159](#)
- tbsm_db
 - create tbsm schema [38](#)
- terminology [2](#)
- Time Window Analyzer
 - metric data store [108](#)
- tipcli

- tipcli (*continued*)
 - AddRole [225](#)
 - DelRole [226](#)
 - exporting plugins [240](#)
 - ListRoles [225](#)
 - ListRolesForPage [228](#)
 - ListRolesForPortletEntity [230](#)
 - ListRolesForView [233](#)
 - ListRolesFromGroup [227](#)
 - ListRolesFromUser [232](#)
 - MapRolesToGroup [227](#)
 - MapRolesToPage [229](#)
 - MapRolesToPortletEntity [230](#)
 - MapRolesToUser [232](#)
 - MapRolesToView [234](#)
 - RemoveRolesFromGroup [228](#)
 - RemoveRolesFromPage [229](#)
 - RemoveRolesFromPortletEntity [231](#)
 - RemoveRolesFromUser [233](#)
 - RemoveRolesFromView [234](#)
 - UpdateRole [226](#)
- tipcli command
 - additional commands [244](#)
 - charting [238](#)
 - import [243](#)
 - ITMLogin command [244](#)
 - portlets [237](#)
 - preference profiles [236](#)
 - SystemInfo command [244](#)
 - TADDMLogin [244](#)
 - user groups [237](#)
 - users [235](#)
 - views [235](#)
- Tivoli
 - integrated applications [13](#)
- Tivoli Access Manager WebSEAL [191](#)
- Tivoli Common Reporting
 - schema name [107](#)
- Tivoli Directory Server
 - installing [24](#)
 - LDAP configuration [24](#)
- Tivoli Documentation Central [2](#)
- Tivoli Event Integration Facility probe
 - requirements [110](#)
 - uninstalling [44](#)
- Tivoli System Automation
 - UNIX configuration [111](#)
 - Windows configuration [112](#)
- Tivoli technical training [3](#)
- tivoli_eif.props
 - modify [153](#)
- training, Tivoli technical [3](#)
- troubleshoot
 - installation issues [155](#)
- typeface conventions [3](#)

U

- uninstall
 - tbsm [68](#), [70](#), [72](#)
- uninstall tbsm [68](#), [70](#), [72](#)
- UNIX
 - DB2 installation [31](#)
- URL Monitor

- URL Monitor (*continued*)
 - attribute group for agent [93](#)
- user
 - administrator default [25](#)
 - groups and roles [25](#)
- user registry
 - default [221](#)
- users
 - external authentication [23](#)
 - registry [23](#)

V

- variables, notation for [15](#)
- VMM
 - for ObjectServer [172](#)

W

- workspaces
 - TBSM agent [81](#)



Part Number:
Product Number:

SC27-8781-03



(1P) P/N: