Business Service Manager
Version 6.2.0

Service Configuration Guide

SC27-8783-03

Note

Before using this information and the product it supports, read the information in Appendix A, “Notices,” on page 235.

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.

Contents

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 Availabilty:

- 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 archictecture 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. Accessing the TBSM console

This section describes how to access IBM Tivoli Business Service Manager (TBSM). The TBSM Console user interface is displayed by an IBM Dashboard Application Services Hub component, which is installed with TBSM.

Logging in

This topic describes how to log in to TBSM.

About this task

Before you log in to TBSM, make sure the ObjectServer is running so that you can send and receive events.

To log in to TBSM, you need to have the following information:

- Your user name
- Your password
- The name or IP address of the TBSM Dashboard server host
- Port used for the TBSM Dashboard server

Two or more users cannot log on using the same web browser on the same workstation at the same time.

To log in, complete the following steps:

Procedure

1. Open a supported web browser and enter the URL of the **Login** page. This URL must be in the format:

   ```
   https://host:port
   ```

   Where:

   - `host` is the host name or IP address of the TBSM Dashboard server
   - `port` is the HTTPS port of the TBSM Dashboard server

     The default URL is `https://`*`hostname`*`:16311/ibm/console/logon.jsp`

     Note: If you get a login error, check that you have specified the URL correctly. If the URL is correct, the IBM Dashboard Application Service Hub server may not be running. Contact your IBM Dashboard Application Service Hub administrator.
2. Enter your user name and password.

   Note: User names and passwords are case sensitive. If you receive a login error message, and are sure that you have entered your user name and password correctly, then contact your Dashboard Application Service Hub administrator.
3. Click **Log In**.

   The default page for your user opens. This page is set by the IBM Dashboard Application Service Hub administrator.

   Note: If you logged in as the IBM Dashboard Application Service Hub administrative user (by default, `tbsmadmin`), you can access Dashboard Application Service Hub administrator functionality. This

functionality enables you to perform user and content management for all IBM Dashboard Application Service Hub applications.

The TBSM pages are listed in the left navigation pane under Administration, Availability, and Troubleshooting. Your user roles determine what pages you see. TBSM administrators (`tbsmadmin`) can access the **BSM Menu** > **Service Configuration, Service Availability, Service Administration, TWA** pages. Regular users can only access the Service Availability pages. You can access the System Status, Audit Log Records, Audit Log Configuration in **Troubleshooting Menu** > **System Status, Audit Log Records, Audit Log Configuration**.

You can also build custom Dashboard Application Service Hub views and pages and add them to this list of tasks. For more information about creating custom views, see the *IBM Tivoli Business Service Manager: Customization Guide*.

If you are having problems logging in, clicking the TBSM product name on the **Welcome** page displays the TBSM **About** page. Click the **TBSM** tab provides version and support information. It also provides links to additional resources. Clicking the **Support Information** tab provides information that may be valuable for problem determination situations.

4. To display a page in Dashboard Application Service Hub, click a page name in the left navigation pane. To view the default TBSM page, select **BSM Menu** > **Service Administration** from the pages list.

   The TBSM **Service Administration** page opens. If this is the first time you have logged on, it shows the default view, with no configured services listed in the **Service Navigation** portlet or in the Service Editor.

   After you have configured your services, the dependencies are displayed in the service tree in the Service Navigation portlet on the left and graphically in the Service Editor on the right.

   Note: In Internet Explorer, TBSM automatically logs out when you press the backspace key on the keyboard. You must log in again to resume your session.

Logging out

This topic describes how to log out of TBSM.

About this task

Click **User Menu** > **Logout** at the top, right side of the page to log out of TBSM.

Note: Do not close the browser window or tab to exit the TBSM console. Always click **User Menu** > **Logout** to exit. Otherwise, your user ID remains active until your session times out.

Chapter 6. TBSM console overview

This topic describes the TBSM default pages and common portlet buttons.

In the left navigation pane, the default IBM Tivoli Business Service Manager (TBSM) console consists of default pages: Service Administration, Service Configuration, Service Availability, and the Time Window Analyzer page. Your permissions control which page you can view and use.

The default **Service Administration** page has the following portlets:

Note: The Service Administration page will not be part of subsequent TBSM releases. Use the Service Configuration page to configure your services models.

- **Service Navigation** portlet on the upper left side shows summary information about service templates, service instances, views, and data. The templates and services are listed in trees.
- Service Editor on the upper right side shows a graphical service-dependency model and custom view definitions and lets you configure your service templates and service instances.
- **TBSM Charting** portlet on the lower left side under the **Service Navigation** portlet shows a chart that summarizes the events that affect a given service model. **This feature will not be part of subsequent TBSM releases.**
- **Service Details** portlet on the lower right side under the Service Editor shows detailed information about the events that affect a service model, as well as data on service level agreements and template rules.

The default **Service Availability** page has the following portlets:

- **Service Tree** portlet on the upper left side shows summary information about the service instances that are listed in trees.
- **Urgent Services** portlet on the lower left side under the **Service Tree** portlet shows a list of services that have a status defined as urgent.
- **Service Details** portlet on the lower right side under the Service Viewer shows detailed information about the events that affect a service model as well as data on service level agreements and template rules. **This feature will not be part of subsequent TBSM releases.**

The default **Service Configuration** page has two portlets:

- **Service Navigation** portlet on the left side shows summary information about service templates, service instances, views, and data. The templates and services are listed in trees.
- Service Editor on the right side shows a graphical service-dependency model and custom view definitions. Use the Service Editor to configure your service templates and service instances.

Note: If you are building services or configuring templates, this page is more efficient because it loads faster than the Service Administration page.

The default Time Window Analyzer page can be used to show service trends over time. It can compare trends between different services or compare trends for a service against recent historical data for the same service. The Time Window Analyzer can also show service affecting events in relation to the service trends. When you configure the view you select the following options:

- One or more services where you want to show key performance indicator (KPI) values for a given time period
- A KPI value you want to plot over time. This value is a numeric rule that is part of the service's template.
- An Overlay that shows when the service status changed over the time period.
- Other recent time periods showing the same data. The chart plots the data for the recent time period on top of the chart for the most recent time period. The Time Window Analyzer contains two elements:
  - The Compare frame lets you select the services you want to view in the chart, and history for those services.

– The Chart frame lets you select the KPI you want to plot and an Overlay you want to display in the chart.

From the left navigation pane, expand **Troubleshooting and Support > Tivoli Business Service Manager**, there are three default TBSM Troubleshooting pages:

- System Status
- Audit Log Records
- Audit Log Configuration

For more information, see the "Getting started with diagnosis" section in the *IBM Tivoli Troubleshooting Guide*.

These portlets are explained in more detail in the sections that follow.

Common portlet buttons

This section describes the common refresh and navigation buttons for portlets in the TBSM console.

Clicking the **Expand** button at the top right of the four portlets exposes the buttons that control how these portlets display.

Table 1 on page 21 describes the common refresh and navigation buttons.

Note: Not all buttons will be available on every portlet.

Table 1. Common Refresh and Navigation Buttons		
Button	**Button name**	**Description**
	Edit Options	Clicking the **Edit Options** button opens a menu with the options to **Restore**,**Minimize**, and **Maximize**. Clicking the **Maximize** button maximizes the portlet to the size of the page. Clicking **Restore** will return the portlet to normal size. Clicking **Minimize** hides the portlet's contents. Clicking **Restore** returns the portlet to normal. **Note:** If you click **Minimize** in a maximized view, the portlet returns to normal size and its contents will be hidden. If you click **Maximize** in a minimized view, the portlet expands to the size of the page, showing its contents. **Important:** Save all changes before clicking **Restore** or any changes you made will be lost. On some portlets, the **Edit options** menu has two additional options: **Personalize** or **Edit Shared Settings**. **Personalize** only changes settings for the current user. **Edit Shared Settings** changes settings for all users. **Note:** After you have set your own preferences by selecting **Personalize**, any future changes to the Default Preferences will no longer apply. If you want similar preferences as those made by the Administrator after configuring your preferences, you will have to manually **Personalize** to make the corresponding changes. For more information see the *IBM Dashboard Application Services Hub Administration Guide* and the *IBM Tivoli Business Service Manager: Customization Guide*.

Table 1. Common Refresh and Navigation Buttons (continued)		
Button	**Button name**	**Description**
	Refresh	Clicking this button reloads the display for a given portlet. For example, whenever you change your service instances or templates in the Service Editor, click the **Refresh** button in the **Service Navigation** portlet to show the changes. Also, clicking this button returns the portlet to the default view. For example, in the **Services** tree, if you have expanded elements of the tree, clicking this button closes them. **Note:** If you click the **Refresh** button in the Service Editor, it refreshes to the default view. If you did not set a starting instance, it returns to the splash screen.
	Help	Clicking this button allows you to access help for the portlet.

Chapter 7. Service Navigation portlet

This topic describes the drop-down menu in the Service Navigation portlet.

The **Service Navigation** portlet, on the upper left of either the Service Administration or the Service Configuration page contains a drop-down menu with the objects:

- Services
- Templates
- Service Component Repository
- Data
- Data Fetcher

Templates is open by default.

Right-click menus are accessible from the **Service Navigation** portlet. The menus give you quick access to many of the editing functions in TBSM.

Services

This topic describes the elements in Services in the Service Navigation portlet.

Services in the Service Navigation portlet contains trees of service-dependency models that have been configured for TBSM. The service instances you see depend on your user and group permissions.

To see the full tree, click the plus symbols to the left of the services. To see a view of a service or template in the Service Editor, click the name of the service or template.

Table 2 on page 23 describes the **Services** elements.

Table 2. Services elements

Services element	Element type	Description
	**Create New Service** button	Click this button to create a service. When you click this button, a blank **Edit Service** tab opens in the Service Editor. Enter the parameters for the new service instance and click the **Save** button.
	**Delete Services** button	Click this button to select the services you want to delete. When you click this button, the **Delete Instances** tab opens in the Service Editor. In the Service Editor, select the services you want to delete and click the **Delete** button.

Table 2. Services elements (continued)		
Services element	**Element type**	**Description**
	**Search** button	Click this button to search for services. When you click the **Search** button, a new window opens. Enter a string in this field and click the **Search** button to search for services that contain the string you entered. The system displays the search results alphabetically by display name and service name. For example, if you want to find all the services with names like `webserver1`, enter webs in the **Search** field. **Important:** Only services that are persisted will return as search results. The search function will not find services that are not persisted.
	**Tree Template Editor** button	Click this button to open an editor where you can create, design, and edit tree templates.
	**Help** button	Click this button to access help.

Templates

This topic describes the elements in Templates in the Service Navigation portlet.

Templates in the **Service Navigation** portlet contains a tree of service templates that have been configured for TBSM. The service templates you see and the actions you can perform depend on your user and group permissions.

To see the full tree, click the plus symbols to the left of the templates. A list of templates is displayed. To display a template in the Service Editor, click the template name in the tree. These buttons are disabled if you do not have permission to use them.

Table 3. Templates elements		
Template element	**Element type**	**Description**
	**Create New Template** button	Click this button to create a service template. When you click this button, a blank **Edit Template** tab opens in the Service Editor. Enter the parameters for the new service template and click the **Save** button.
	**Delete Templates** button	Click this button to select the templates you want to delete. When you click this button, the **Delete Templates** tab opens in the Service Editor. In the Service Editor, select the templates you want to delete and click the **Delete** button.
	**Configure Event Enrichment** button	Click this button to configure Event Enrichment. Event Enrichment allows you to monitor an event source for new events. Impact "catches" new and updated alerts as they are sent to the ObjectServer, and then goes to one or more external data sources to correlate information in the alerts with business data. For more information, see the topic in the *Tivoli® Netcool/Impact Solutions Guide*.

Table 3. Templates elements (continued)		
Template element	**Element type**	**Description**
	**Help** button	Click this button to access help.

Service Component Repository

This topic describes the Service Component Repository in the Service Navigation portlet.

The services in the Service Component Repository (SCR) are required for the optional Discovery Library Toolkit. When you import the SCR services and templates, TBSM creates a set of services and service templates. The default SCR services and service templates enable TBSM to discover and create new services from Discovery Library data retrieved by the Toolkit. The discovered services display in the **Service Component Repository**, accessed from the **Service Navigation** drop-down list. To monitor these services, you must create a dependency between the SCR service and one of the services in the **Services** tree.

Note: The information that is displayed for the Service Component Registry in the **Services** tree reflects the last validation of top level resources such as AppServers, Oracle, and CICS. The Service Component Registry does not update the **Services** tree in the same way as it updates the resources that are displayed the Services tab.

CAUTION: Do not alter these services. Do not alter any of the service templates with a prefix of SCR or BSM. Altering these services and templates may cause the Discovery Library Toolkit to malfunction. For more information about setting up the Discovery Library Toolkit and working with SCR services, see the *IBM Tivoli Business Service Manager: Administrator's Guide* and the *IBM Tivoli Business Service Manager: Customization Guide*.

Data

This topic describes the elements in Data in the Service Navigation portlet.

Data in the **Service Navigation** portlet shows a list of data sources that provide data for your service model. You can only configure data sources if you have administrator permissions for TBSM. When you edit or add a data source, the **Edit** tab for data sources is displayed in the Service Editor. The tab for creating or editing data sources opens when you click the 'Create New Data Source' button in the Service Navigation portlet or click on an existing data source.

Table 4 on page 25 describes the **Data** elements.

Table 4. Data elements		
Data element	**Element type**	**Description**
	Refresh button	Click this button to refresh the list of data sources. If you created a data source in the **Edit** tab for data sources, click the **Refresh** button in **Data** to update the display to show the new data source.
	Create New Data Source button	Click this button to create a data source. When you click this button, a blank **New Data Source** tab opens in the Service Editor. Enter the parameters for the new data source and click the **Save** button.

Table 4. Data elements (continued)		
Data element	**Element type**	**Description**
	Help button	Click this button to access help.

Table 5 on page 26 describes the **Data** right-click menus. These options are disabled if you do not have permission to use them.

Table 5. Right-click menu options for Data Sources	
Right-click menu option	**Description**
Edit	This right-click option opens the **Edit** tab in the Service Editor, where you can edit a data fetcher.
Delete	This right-click menu option deletes the data source.

Data Fetcher

This topic describes the elements in Data Fetcher in the Service Navigation portlet.

The **Data Fetcher** in the **Service Navigation** portlet shows a list of data fetchers that have been configured to deliver data periodically from databases. The **Data Fetcher** allows you to query a database and use the data to affect the status of a service model. The data can also be used to auto-populate the service model.

You can define a data fetcher for the following types of databases:

- Oracle
- Sybase
- MS-SQL
- MySQL
- Informix
- DB2

Table 6 on page 26 describes the **Data Fetcher** elements. These buttons are disabled if you do not have permission to use them.

Table 6. Data Fetcher Elements		
Data Fetcher element	**Element type**	**Description**
	Refresh button	Click this button to refresh the list of data fetchers. If you create a data fetcher in the Data Browser, click the **Refresh** button in the **Data Fetcher** to update the display to show the new fetcher.
	Create New Data Fetcher button	Click this button to open the **New Data Fetcher** tab in the Service Editor, where you can configure a new data fetcher.
	Help button	Click this button to access help.

Table 7 on page 27 describes the **Data Fetcher** right-click menus. These options are disabled if you do not have permission to use them.

Table 7. Right-click options for Data Fetchers

Right-click menu option	Description
Edit	This right-click option opens the **Edit** tab in the Service Editor, where you can edit a data fetcher.
Fetch Now	This right-click option runs the data fetcher and the resulting events are processed by the server.
Show log	This right-click option opens a log for the data fetcher activity in the Service Editor.
Enable/Disable	This right-click option disables/enables the data fetcher.
Delete	This right-click option deletes the data fetcher.

The **Status** column indicates the status of the data fetcher. For example, a green status indicates a data fetcher is working correctly.

Chapter 8. Service Editor

This topic describes the Service Editor portlet.

When you click a service, service template, data source, or data fetcher in the **Service Navigation** portlet, a detailed display opens in the Service Editor, either a graphical view or an editor depending on what you selected. Some of these options have no graphical view.

The toolbar for each view or editor contains tools appropriate for that tab.

Note: If you click the **Refresh** button in the Service Editor, it refreshes to the default view. If you did not set a starting instance, it displays the "No service currently selected" message. Select an instance, and it will display in the Service Editor.

Right-click menus are accessible from the Service Editor. The menus give you quick access to many of the editing functions in TBSM.

Note: By default, the Service Editor portlet, which has only edit options, is featured on the Service Configuration and Administration pages.

Edit tabs

This topic describes the Edit tab in the Service Editor.

The **Edit** tab let you configure your service templates, service instances, service component repository, data sources, and data fetchers in TBSM.

Edit Service tab toolbar

Table 8 on page 29 describes the **Edit** tabs' tools.

Table 8. Edit tab toolbar

Button	Name	Description
	**Save** button	Click this button to save the information that you entered in one of the tabs in an editor. You must save your entries in each tab before continuing to the next one.
	**Help** button	Click this button to access help for the tab.
	**Refresh** button	Click this button to reload the last saved values for a given tab. (Not applicable to every editor.)

Edit tab editors

This topic describes the Edit tab editors in the Service Editor.

The **Edit** tab contains one of the editors, depending on what you selected from the **Service Navigation** portlet. These tabs are described in the following sections.

Edit Service tab

When you click a service or view, or create a service or view, the **Edit Service** tab is displayed. You configure template assignments, dependent services, and other attributes for the selected service in this tab.

Edit Template tab

Service templates define how services respond to external data and the status of other services. You can assign (or *tag*) a service with a template or templates. When you select a service template or create a template, the **Edit Template** tab is displayed. You configure rules, output expressions, and other attributes for the selected service template in this tab.

Edit Data Source tab

When you click a data source or create a data source, the tab for creating or editing data sources is displayed in the Service Editor. You set up a data source to access additional data about your services. You can only configure data sources if you have administrator permissions for TBSM. In the **Edit Data Source** tab, you enter the information you need to access the data in a database such as an ObjectServer or an SQL database.

Edit tab for Data Fetcher

When you click a data fetcher or create a data fetcher, the **Edit Data Fetcher** tab opens in the **Edit** tab of the Service Editor. You set up data fetchers to query external databases and use the returned data to affect the status of the service model.

Delete tab

When you click a **Delete** button for services or templates, the appropriate **Delete** tab is displayed in the Service Editor. Select the item or items you want to delete from the **Delete** tab and click the red **Delete** button in the **Delete** tab toolbar.

Chapter 9. TBSM service model configuration overview

TBSM lets you create models of the services required to monitor your organization. In TBSM, these models are called *service models*. The sequence of the stages is a general guideline, since you can perform some of these stages out of sequence. .

For more information about each stage see:

1 Plan your service model

Before you start to configure a service model in TBSM, it is important to develop a plan that includes information about the services and service dependencies you want TBSM to monitor. Identify distinct service types that have common characteristics.

To help you identify the common service types, create a draft model of this service hierarchy in TBSM that includes the following information:

- What do you monitor in your environment now with the applications such as IBM Tivoli Monitoring, Netcool/OMNIbus, and other applications? How would you visualize this incoming data for your clients?
- What business units or customers do you want to create dashboards for? The dashboard for the IT department is different from a dashboard for an executive. What data sources do you need to access and filter for your dashboards?
- What are the key performance indicators and other data you want to highlight on your dashboards.
- What are the highest level services you want to monitor on your dashboard? Do these services represent customers, business departments, or a business? You can map out services to represent them and their dependencies.
- What are the applications, devices–and so on–that support the high-level services? That is, what are the dependencies for the high-level services? You can map out child services to represent these items.
- How do the child services affect the high-level or parent services at the top of the service hierarchy?
- What are the Netcool/OMNIbus events and other data that affect each service type?

2 Create service templates

When you have figured out the service hierarchy, create service templates for services that share common behavior. For example, if you want to monitor multiple web servers, you create a web server template that defines the attributes common to all the web servers. When you create services, you assign these service instances to a given service template.

3 Create structural dependencies

In service templates, you define the relationships between the various services to set the structure for a given service model. These rules aggregate the overall service status based on the rule-output values for multiple child services and are called dependency-aggregation rules. These rules are based on status values such as Bad or Marginal that are passed from child service templates.

4 Identify data sources

Identify the data sources that contain information about the services you want to create. You use this data to build your service model structure, and to monitor the status of the services in the model. TBSM uses the default ObjectServer that is installed with TBSM as a data source for your service models. You can also configure SQL databases as data sources and select data with SQL queries.

Also, you must configure Netcool/OMNIbus or database queries so that a single event or row shows the status of a given service.

5 Create incoming status and numerical rules

In TBSM, three types of rules determine the status of a given service or calculate a numerical rule-output value based on external data:

- Incoming status rules
- Numerical aggregation rules
- Numerical formula rules

Incoming status rules

TBSM service templates have the following types of incoming status rules:

- Good, Marginal, and Bad rules
- Numerical rules

TBSM uses Good, Marginal, and Bad incoming status rules to obtain the general state of a service from an incoming ObjectServer event or other data. The general service state is determined by comparing threshold values set in the rule definition to event field values or field values in other incoming data. When this type of rule is processed, TBSM obtains a general service state of Good, Marginal, or Bad and assigns this value to the service instance.

TBSM uses numerical incoming status rules to obtain a numerical value from fields in ObjectServer events or other data and then to associate this value with a service instance as a rule-output value. The resulting rule-output value is stored in memory and identified by the rule name where referenced elsewhere in TBSM.

TBSM uses numerical and text incoming status rules to obtain a value from fields in ObjectServer events or other data and then to associate this value with a service instance as a rule-output value. The resulting rule-output value is stored in memory and identified by the rule name where referenced elsewhere in TBSM.

Numerical aggregation rules

TBSM uses numerical aggregation rules to combine rule-output values of the same type across a set of child service instances. This type of rule is defined as part of the parent service template. The rule-output values combined by a numerical aggregation rule can be generated by incoming numerical-status rules or by other aggregation and numerical formula rules.

Numerical formula rules

TBSM uses numerical formula rules to combine rule-output values of different types within the same service instance. These output values can be generated by incoming numerical-status rules or by numerical aggregation rules. When a numerical formula rule is processed, the resulting combined output value is assigned to the same service instances where the original output values existed.

6 Create services

When you have defined the rules for service templates in your model, then you create the service instances for your service model. A service instance, or service, is a unique instance of a given service template. For example, if you have an E-mail Server service template, you can assign services called `e-mail1`, `e-mail2`, and `e-mail3` to the service template.

Optionally, you can automatically create the services by creating auto-population rules, External Service Dependency Adaptor (ESDA) rules or with the Discovery Library Toolkit.

Service name restrictions

Important: TBSM checks the Service Name field for invalid characters. The Name field for services must not contain these special characters:

```
" < > \ * ? | ;
```

TBSM will not create a service when the Name field contains any of these invalid characters. If the object does not save, remove any special characters from the name field.

Service names longer than 127 characters can cause display and performance issues.

Service names longer than 127 characters will be truncated in the Service Details portlet display. This does not affect the functioning of TBSM, but the truncated service name will appear in the Service Details portlet.

If you use the Node column in Netcool/OMNIBus to create or match your service name: The node column is limited to 64 characters. If the service name is greater than 64 characters, it will be truncated and TBSM will not receive any events for the service.

You can either change the incoming status rule to use different service instance naming fields, or you can change the identification fields for a service with a long name. The identification field or fields can be different than the service naming field or fields. For example, if you have a service name that uses the Node column for its name, and the name is longer than 64 characters, the service will not receive events. To fix this, you can specify another field, such as a field that contains the IP address as the Identification field for the service.

7 Create custom service trees and views

When you configure your numerical rules, you can display the rule-output values in the Services tree (Service Availability) or in the Service Editor (Service Configuration or Service Administration).

8 Create SLAs

Optionally, you can define service level agreement (SLA) settings for the service templates in your model. You can measure the downtime and the number of outages over a given time period in relation to the SLAs for your organization. For more information about SLA rules, see the TBSM *Service Configuration Guide*.

9 Set permissions

For users and user groups, you can define the view and edit permissions for a given service template or service instance. For example, if you have 10 service models, you can set user group permissions to view and edit only two of the models. You use the Dashboard Application Service Hub **Users and Groups** tasks to manage permissions for pages and portlets. To manage object level permissions, use the Security tab in the service and template editors.

10 Monitor services

When you set up your TBSM service model, you can monitor it and change its appearance in the *Service Viewer*. For more information, see the *TBSM Service Configuration Guide*.

Chapter 10. Data sources

This section describes the use of data sources in IBM Tivoli Business Service Manager (TBSM) and explains how to configure them.

About data sources

This topic describes data sources along with unique driver-specific requirements.

A TBSM data source provides connection information to an external source of data (for example, a relational database) that contains information that affects the status of services in your service model.

To establish a database connection between TBSM and a data source such as an ObjectServer or a database, you configure the connection information in the **Edit** tab for data sources.

You must create a data source mapping for each data source that you want to access within TBSM. Data sources store database information such as the host name, port, and user information you need to connect to the data source.

Only certain types of data sources are supported. These types are listed in **SQL Type** drop-down list. The TBSM software supports the following external applications for data sources:

- ObjectServer (see first note)
- Informix
- Sybase
- Oracle
- MySQL
- MS-SQL
- Postgres
- DB2

Note: The default ObjectServer is optimized for TBSM. Do *not* use an ObjectServer data source unless you want to use a table other than `alert.status`.

Note: If you want to use any non-SQL data sources, you need to use Netcool/Impact to extend TBSM for these environments. See the Netcool/Impact documentation for more information.

MySQL data source requirements

Before you add a MySQL data source to TBSM, you need to grant privileges to TBSM server's host machine on your MySQL server. Otherwise, you will not be able to establish a connection between your TBSM host and your MySQL data source.

From your MySQL host, you can grant access with the command:

```
grant all on <database>.* to <username>@<tbsm server hostname>
identified by <password>;
```

For more information about granting host privileges, see your MySQL documentation.

DB2 data source driver requirements

When creating a DB2 data source in TBSM, the **Use Type 4 JDBC Driver** check box on the **New Data Source** window is checked by default. If the check box is cleared, then the server will attempt to establish connections to the DB2 database using a legacy driver which is no longer shipped or installed with the product. For information about using this driver or the legacy driver, see the README file in the TBSM_HOME/`contrib` directory.

Creating a data source

This task describes how to create your own data source.

About this task

To create a data source, complete the following steps:

Procedure

1. Expand **Administration** in the left navigation pane. Click **Service Administration** or **Service Configuration**.
2. Select **Data** in the **Service Navigation** drop-down menu.

   Data lists any data sources that have already been configured, in this case, the `TBSMComponentRegistry` data source that was installed with TBSM.
3. Click the **Creat New Data Source** button.

   The **New Data Source** tab opens in the Service Editor.

   Note: The Name field for rules, data sources, maintenance schedules, view definitions and other objects must not contain these special characters:

   ```
   " & < > \ / * ? | ( ) : ; $ ! %
   ```

   Names must not contain spaces. Otherwise, these objects may not be saved properly in the TBSM database. If the object does not save, remove any special characters or spaces from the name field.
4. Select the type of database you want from the **SQL Type** drop-down list.

   The fields shown depend on the type of SQL type selected.

   Table 9 on page 36 describes the entries for an ObjectServer SQL Type:

Table 9. Edit tab for data sources elements: ObjectServer selected

Field	Description
Data Source Name	A unique name for the data source. See important Warning below. **Note:** Do not under any circumstances name a data source of any type ObjectServer. Doing so will damage your application.
Username	The user name required to access the database.
Password	The password required to access the database.
Primary Source:	
Host name	The name or IP address of the host system for the database.
Port	The port number on the host system for the primary database.
Failover Policy	If no backup data source is available, click **Disabled**. To fail over to a backup data source, click **Failover**. If you want the primary data source to take over when it becomes active again, click **Failback**.
Backup Source (Optional):	
Host name	The name or IP address of the host system for the backup database.
Port	The port number on the host system for the backup database.

Table 10 on page 37 describes the fields for the remaining SQL databases (not all fields apply to every type of database):

Table 10. Edit tab for data source elements: SQL database selected

Field	Description
Data Source Name	A unique name for the data type.
Username	The user name required to access the database.
Password	The password required to access the database.
Primary Source:	
Host name	The name or IP address of the host system for the database.
Port	The port number on the host system for the database.
SID (Applies to Oracle databases only)	Enter an Oracle service identifier. The default value is ORCL. For more information, see your Oracle documentation.
Server	The server name where the database resides. (Informix only)
Use Type 4 JDBC Driver (DB2 databases only)	When creating a DB2 data source in TBSM, the `Use Type 4 JDBC Driver` check box on the **New Data Source** window will be checked by default. If the check box is cleared, then the server attempts to establish connections to the DB2 database using a legacy driver which is no longer shipped or installed with the product. For more information, see the readme file located in `$TBSM_HOME/contrib`.
Database (Not applicable for an Oracle database)	The name of the database you want to connect to.
Failover Policy	If no backup data source is available, click **Disabled**. To fail over to a backup data source, click **Failover**. If you want the primary data source to take over when it becomes active again, click **Failback**.
Backup Source (Optional)	
Host name	The name or IP address of the host system for the backup database.
Port	The port number on the host system for the backup database.
Use Custom URL (Applies to Oracle databases only	By clicking on "Yes", you can use customized URL to connect to oracle datasources, which allows for connection using service names. This is the same as using 'Customized URL' for connection settings in Impact GUI. If "No" is selected you must use SID for connection to an Oracle database. The default selection is "No".
URL (Applies to Oracle databases only)	Description: This field is required if "Yes" is selected for Use Custom URL. The url prefix ""jdbc:oracle:thin:" is not required. Please see: http://www.ibm.com/support/knowledgecenter/SSSHYH_6.1.1.3/com.ibm.netcoolimpact.doc/user/rac_cluster_support.html?lang=en.
SID (Applies to Oracle database only)	This field is required if "No" is selected for Use Custom URL. Enter an Oracle service identifier. The default value is ORCL. For more information, see your Oracle documentation.
Server	The server name where the backup database resides.
Database (Not applicable for an Oracle database)	The name of the backup database you want to connect to.

5. Verify that you have entered the correct information by clicking the **Test Connection** button. A window pops up, indicating whether database connection succeeded or failed.
6. Click **Save**.

   Note: If you create an Informix data source, restart the Data Server.

Editing a data source

This task describes how to edit data sources.

About this task

To edit an existing data source:

Procedure

1. In the **Service Navigation** drop-down menu, select **Data**.
2. Click the data source you want to edit to open it in the **Edit** tab in the Service Editor.
3. Edit the entries using the information in Table 9 on page 36 as your guide.

Deleting a data source

This task describes how to delete a data source.

About this task

To delete an existing data source:

Procedure

1. In the **Service Navigation** portlet drop-down menu, select **Data**.
2. Right-click the data source you want to delete, and select **Delete**.

   A message window opens asking you to confirm the deletion.
3. Click **OK** to delete the data source.

Chapter 11. Data fetchers

This section describes how to set up and use a data fetcher in IBM Tivoli Business Service Manager (TBSM).

Data fetcher overview

This topic describes data fetchers.

Data fetchers query a data source at intervals and then provide the data returned from the query to TBSM for use in determining the status of a service. When you configure a data fetcher, you specify the terms of the query and the query interval.

After you have created a connection to an external data source, you can configure a data fetcher to send queries to the database. The database returns the results of the query to TBSM. Data fetchers can be configured in SQL and IBM Tivoli Monitoring Policy (backed by an ITM web service).

You can configure queries to any of the databases defined in TBSM.

Note: The default ObjectServer is optimized for TBSM. Do *not* use an ObjectServer data fetcher unless you want to use a table other than `alert.status`.

After you have set up the criteria for the data fetcher, you can use the queried data to complete the following steps in TBSM:

- Create incoming status rules to update the status of service instances
- Create auto-population rules that are based on the incoming status rules that are based on the data fetcher data

You configure the Data Fetcher's polling period to run either periodically or at the same time each day.

Note: The data fetcher results can be cached. There are benefits to configuring the data fetcher to allow it to cache data. However, you should configure the data fetcher to cache the numbers of rows that is appropriate for your configuration and also be aware of the limitations of caching. For more information about configuring data fetcher caching, seeData fetcher tuning in the *IBM Tivoli Business Service Manager Administrator's Guide*.

Data fetcher set up

Use the procedures in this section to set up a new data fetcher. Some of the examples in this section use data from the `TBSMDEMO.tickets` table in the `tbsmDB` database that is included with your installation of TBSM. Those examples are based on the tutorials included in the *TBSM Scenarios Guide*.

Time interval hints

When you enter an invalid time, a hint is displayed next to the field showing the acceptable format for the time. The format expressed is like the Java class `java.text.SimpleDateFormat`.

Table 11. Time hint notation

Notation	Description
h	12 hour clock time.
H	24 hour clock time.
t	The am or pm indicator for 12 hour clock time.
hh	Hours with leading zero (01, 02, and so on).

Table 11. Time hint notation (continued)	
Notation	**Description**
mm	Minutes with leading zero (01, 02, and so on).
h	Hours without a leading zero (1, 2, and so on).
m	Minutes without a leading zero (1, 2, and so on).

Setting up a data fetcher

Use the procedures in this section to set up a new data fetcher. Some of the examples in this section use data from the `TBSMDEMO.tickets` table in the `tbsmDB` database that is included with your installation of TBSM. Those examples are based on the tutorials included in the *TBSM Scenarios Guide*.

Creating the data fetcher

This task describes how to create a data fetcher.

About this task

To create a data fetcher called `RegionalTickets`, complete the following steps:

Procedure

1. Click **Data Fetcher** in the **Service Navigation** drop-down menu.

   The **Data Fetcher** opens.

   Note: Before you can set up a data fetcher in TBSM, you must first configure a data source connection to the database.
2. Click **Create New Data Fetcher** in the **Data Fetcher** toolbar to set up a new data fetcher.

   Figure 1. Create New Data Fetcher button

   The **New Data Fetcher** tab opens in the Service Editor.
3. Enter a unique name for the data fetcher.

   Note: Data Fetcher names are not case-sensitive. For example, TBSM sees `DBFetch1` and `dbfetch1` as the same data fetcher.

   Important: The Name field for data fetchers must not contain these special characters:

   ```
   $ ! % & " < > | : / * \ ?
   ```

   Names must not contain spaces. Otherwise, these objects may not be saved properly in the TBSM database. If the data fetcher does not save, remove any special characters or spaces from the name field.
4. After you enter a valid name for the data fetcher you can do the following:
   - Set the polling criteria
   - Set the query for the data fetcher
   - If you want to set up a Tivoli Monitoring Policy data fetcher, see *Data fetchers for Tivoli Monitoring* in the *TBSM Customization Guide*.

What to do next

Note: If you select to delete a Data Fetcher while the Data Fetcher is currently being edited in the Editor, the Editor will not be automatically closed.

Setting the polling criteria

This task describes how to set the polling criteria, configure a daily fetch, and set the fetching interval.

About this task

This section explains how to select the polling criteria. First you need to decide whether you want the data returned at an interval that you specify or whether you want the data returned at a specific time each day. The next section describes how to set a daily fetch.

Note: When you configure a query in the data fetcher for the first time and test it, it is better to select the **Daily Fetch at** option. If you configure a query to use the fetching interval option, you could be running an incomplete data fetcher many times, which could produce strange results. Before testing your query each time, use the **Fetch Now** right-click menu in the Service Navigation portlet to reset the data. You can find the **Fetch Now** if you right-click your test data fetcher in the **Data Fetcher** in the **Service Navigation** portlet.

Note: The default query timeout is 2 minutes. You can change the timeout length in the `RAD_server.props` file. The property name is `server.timeout` and the timeout is in milliseconds. The default timeout of `server.timeout=120000` is 2 minutes.

Configuring a daily fetch

About this task

To configure a data fetcher to poll data from a database once a day, select the **Fetch Daily at** radio button and enter or select the time you want to retrieve the data. Since this example explains how to set up a test data fetcher, it uses the `Daily Fetch at` option.

Note: When setting up a daily fetch, you do not have to enter any information in the interval setting fields.

Procedure

1. Select the **Fetch Daily at** radio button.
2. Accept the default setting to run the query once a day at 12:00. For this example, you use the **Fetch Now** right-click menu in the **Service Navigation** portlet's **Data Fetcher** to run this data fetcher.
3. To enter the polling criteria manually, see “Setting up the query criteria” on page 42. To use the Query Builder, see Chapter 16, “Query Builder,” on page 105.

Setting the fetching interval

About this task

This section describes how to set the minimum and maximum intervals between fetches.

To set a fetching interval, you select the minimum and maximum intervals between fetches. Since the length of time a fetch takes depends on the number of rows returned, the length of time before the next fetch is determined by multiplying the number you set in the **Fetcher Interval Multiplier** field by the time it took to return and process the records from the last fetch. Use the information in Table 12 on page 41 to help you set the fetching interval.

Table 12. Setting the fetching interval

Editor element	Description
Minimum Interval Between Fetches spin field	Enter or select the minimum interval between fetches.

Table 12. Setting the fetching interval (continued)	
Editor element	**Description**
Maximum Interval Between Fetches spin field	Enter or select the maximum interval between fetches.
Fetcher Interval Multiplier	The system determines the fetching interval by multiplying the number (for example, 5) entered in this field by the time it took to fetch and process the records from the previous fetch. The fetching interval will not exceed the number of seconds set in the **Maximum Interval Between Fetches** field, for example, 300. It will also not fall below the number of seconds set in the **Minimum Interval Between Fetches** field, for example, 30. If the Data Fetcher takes 60 seconds to fetch and process the records, the next fetch would begin in 300 seconds (60 x 5).

Setting up the query criteria

This task describes how to set up the query criteria, enter the SQL query manually, view the data fetcher results using fetch now, and add an expression.

About this task

After you have set up the polling criteria, you are ready to select the data you want from the database by building an SQL query.

Note: When you create your query, be sure to select only the fields that you need to identify unique service names and create incoming status rules for the services you want to model. For a given service instance, your query must select only one row that contains the values you want to retrieve with a given incoming status rule.

You have two options for building the query:

- Select a previously configured data source from the drop-down list and enter an SQL statement manually in the space provided.
- Click **Query Builder** to use the Query Builder wizard. Within the Query Builder wizard, you can either select an existing data source or configure a new one and then use the wizard to help you build the query. For information about how to use the Query Builder, see the *TBSM Scenarios* guide.

  Note: If you use the Query Builder to build the query, you should *only* edit it using the Query Builder. If you change the query manually in the text area, those changes will not be reflected in the Query Builder windows. You would have to rebuild the query again in the Query Builder.

Entering the SQL query manually

About this task

This section describes how to enter an SQL query manually.

Note: If you want to enter the query manually, you must create the data source first, if it has not already been created.

To enter an SQL query manually, complete the following steps:

Note: Unless you are familiar with the databases you will be working with, using the Query Builder is much easier than entering the statement manually. The Query Builder gives you access to the names of all the tables in the database and the fields in the tables that you select to include in the query.

Procedure

1. Select a previously configured data source from the **Data Source** selection list.

2. Enter a `Select` statement to select data from the table.

   Setting case in queries: All `Select` statements should reflect the case sensitivity of the database being used. When you select a column in your query, by default, your results show the column name as defined in the database. You can use the **AS** clause to control case of meta data coming back from database. Enclose the metric column name in quotes to make sure the returned column name matches the case you want. For example, to have a query return and uppercase result, enter:

   ```
   select columnname as "COLUMNNAME"
   ```

   In this example, you are going to create a data fetcher called `REGIONALTICKETS` that selects all open tickets with a severity of 4 or higher for a group of services in the `TBSMDEMO.tickets` table and displays them in descending order. The `as "OPENTICKETS"` clause creates a metric named `OPENTICKETS`. Enclose `"OPENTICKETS"` in quotes to make sure that the column name returned from the query is in all UPPERCASE letters, that is, `OPENTICKETS`.

   ```
   select CUSTOMER,HIGHLEVELSERVICEID,LOWLEVELSERVICEID,REGION,SEVERITY,
   STATUS,COUNT(*)as "OPENTICKETS" from TBSMDEMO.TICKETS where SEVERITY>=4
   and STATUS='Open' group by CUSTOMER,HIGHLEVELSERVICEID,
   LOWLEVELSERVICEID,REGION,SEVERITY,STATUS order by OPENTICKETS desc
   ```

3. Select **View** to view the data from that database.

   The **View Data** window opens.

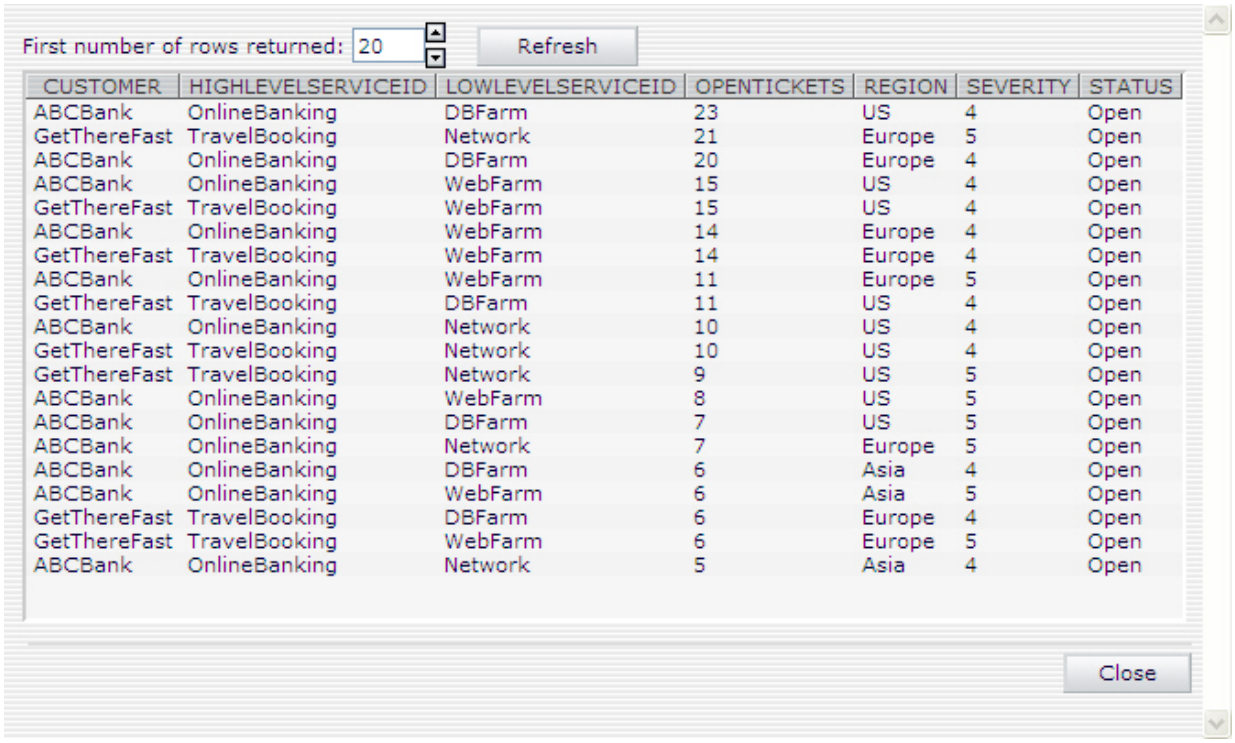


Figure 2. View Data window

   The first 20 rows are displayed by default. You can change the number of rows displayed by entering or selecting a different number of rows in the **First** field and then clicking the **Refresh** button.

   Note: If you enter the query by hand you may get exception errors when you execute a complex query. For example, the TBSM database does not accept queries with column aliases in the where clause. You can get similar errors if you pasted your query in from another application. When you paste in a query, your query may contain invalid characters. If you get such errors, check your query or try entering it in the Query Builder.

4. Close the **View Data** window.

Viewing the data fetcher results using fetch now

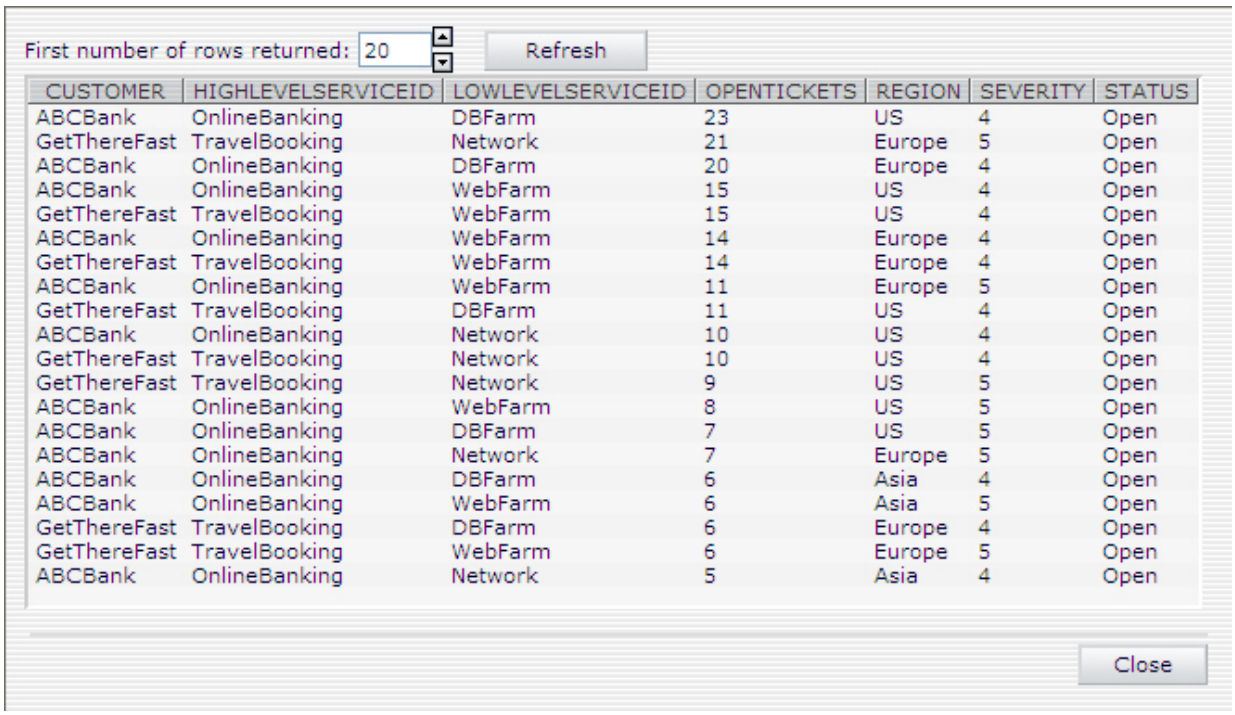
About this task

When you have set up your SQL query, you can run the data fetcher to view the results:

Procedure

1. Select **Data Fetcher** in the drop-down menu for the **Service Navigation** portlet.
2. Select **Fetch Now** from the right-click menu for the REGIONALTICKETS data fetcher to execute the query.
3. To see the results of the query, click the data fetcher to open it in the Service Editor.
4. Click **View**.

   The **View Data** window opens.



CUSTOMER	HIGHLEVELSERVICEID	LOWLEVELSERVICEID	OPENTICKETS	REGION	SEVERITY	STATUS
ABCBank	OnlineBanking	DBFarm	23	US	4	Open
GetThereFast	TravelBooking	Network	21	Europe	5	Open
ABCBank	OnlineBanking	DBFarm	20	Europe	4	Open
ABCBank	OnlineBanking	WebFarm	15	US	4	Open
GetThereFast	TravelBooking	WebFarm	15	US	4	Open
ABCBank	OnlineBanking	WebFarm	14	Europe	4	Open
GetThereFast	TravelBooking	WebFarm	14	Europe	4	Open
ABCBank	OnlineBanking	WebFarm	11	Europe	5	Open
GetThereFast	TravelBooking	DBFarm	11	US	4	Open
ABCBank	OnlineBanking	Network	10	US	4	Open
GetThereFast	TravelBooking	Network	10	US	4	Open
GetThereFast	TravelBooking	Network	9	US	5	Open
ABCBank	OnlineBanking	WebFarm	8	US	5	Open
ABCBank	OnlineBanking	DBFarm	7	US	5	Open
ABCBank	OnlineBanking	Network	7	Europe	5	Open
ABCBank	OnlineBanking	DBFarm	6	Asia	4	Open
ABCBank	OnlineBanking	WebFarm	6	Asia	5	Open
GetThereFast	TravelBooking	DBFarm	6	Europe	4	Open
GetThereFast	TravelBooking	WebFarm	6	Europe	5	Open
ABCBank	OnlineBanking	Network	5	Asia	4	Open

Figure 3. View Data window with the query results

The order-by clause and the group-by clause that are entered result in the row with the highest number of open tickets appearing at the top of the retrieved data. Also, for each unique combination of fields, there are two rows of data: one for severity 4 tickets, and one row for severity 5 tickets. The first row in the OPENTICKETS field contains the count for the severity 4 tickets for the following unique combination of field values:

- CUSTOMER = ABCBank
- HIGHLEVELSERVICEID = ABCOnlineBanking
- LOWLEVELSERVICEID = DBFarm
- REGION = US

  The OPENTICKETS field contains a count for the severity 5 tickets for the same combination of fields. This combination of field values is used to uniquely identify a given instance of a service template. In this case, the service instance is DBFarm in the US REGION for the ABCBank CUSTOMER.

Adding an expression

About this task

Use the query expression to limit the rows returned by the Data Fetcher, improving the efficiency of the fetcher processing. Use a query expression. If you add an expression to the query, the last record from the previously fetched data will be used to build the expression. The expression is added to the fetcher query as an additional filtering clause that limits the rows returned by the fetcher. Use the following procedure to add an expression to the query.

Procedure

1. Click the **Use Expression?** icon.

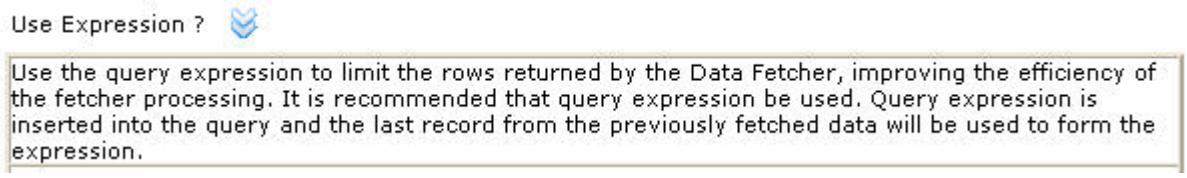


Figure 4. Data Fetcher editor - expression builder section

   This icon opens the expression builder section of the Data Fetcher editor.

2. Enter an expression or click **Expression Builder** to open the **Expression Builder** window.

   The **Expression Builder** window opens. You can use this window to help you build the expression.

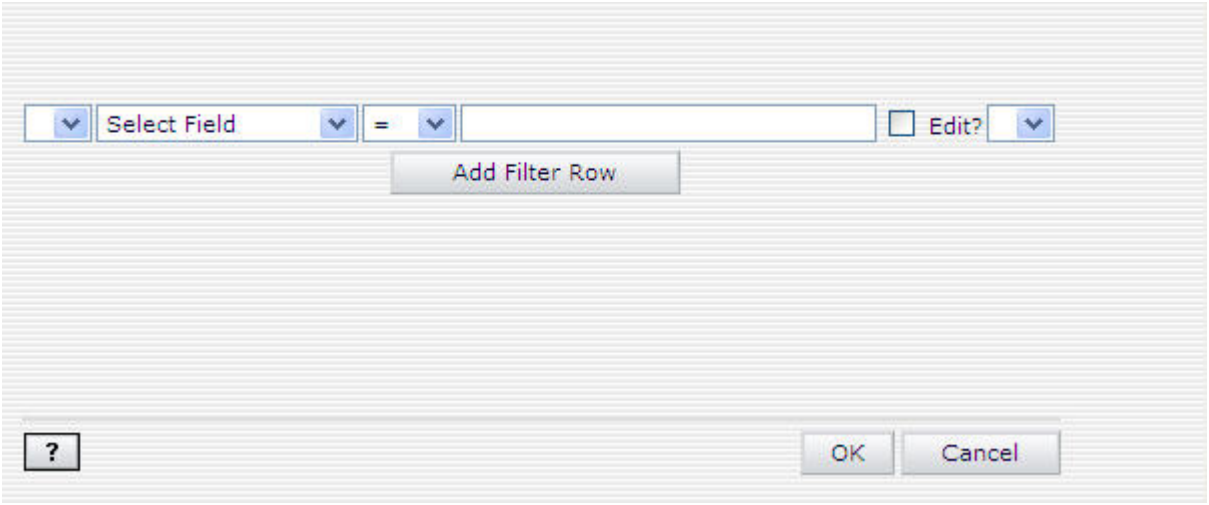


Figure 5. Data Fetcher Expression Builder window

   An expression can be entered that tells the data fetcher to fetch records based on a time that is greater than the last record received or when `Process` equals TRUE.

3. Select the open parenthesis by clicking the down arrow on the left side of the window.
4. Select a field. For this example, select the **SEVERITY** field.
5. Select an operator. For this example, select **>** (greater than).
6. Select the *_SEVERITY_* variable.

   In the example below, the underscores on either side of the `severity` value indicate a variable.

   This variable is replaced by the time value of the last row from the previous fetch.

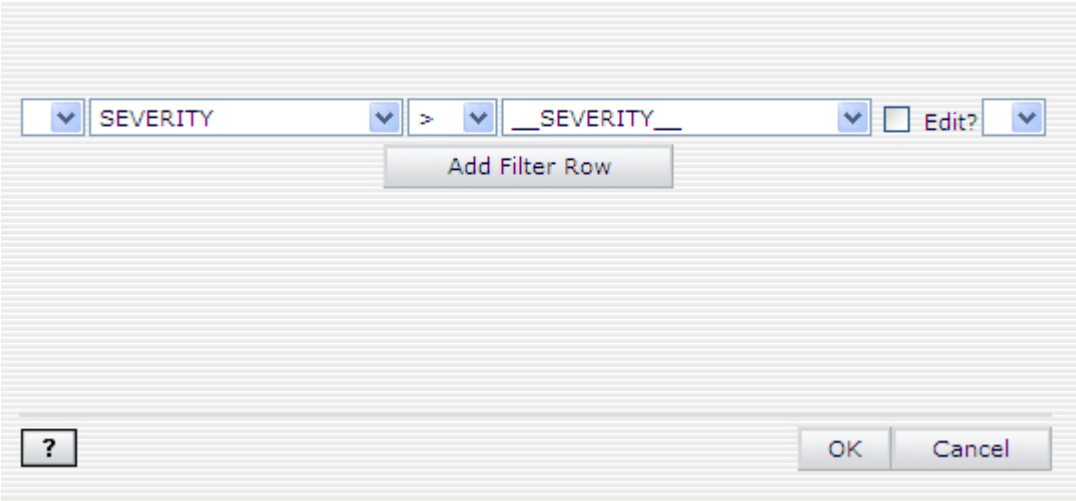


Figure 6. Expression Builder - time variable added

7. Click **Add Filter Row** to add a second row to add a second condition.

   In the example in Figure 7 on page 46, the process condition is added.
8. Select the OR condition.
9. Select the STATUS field.
10. Select the = operator.
11. Enter BAD in the next field.

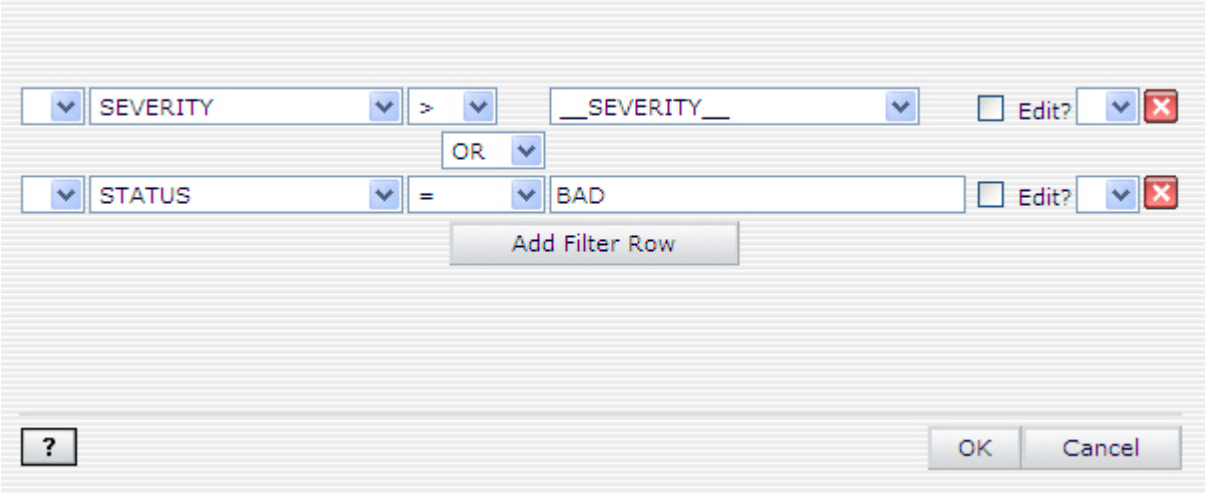


Figure 7. Expression Builder with second expression added

12. Click **OK** to save the expression and close the window.

    The expression is added in the text field.

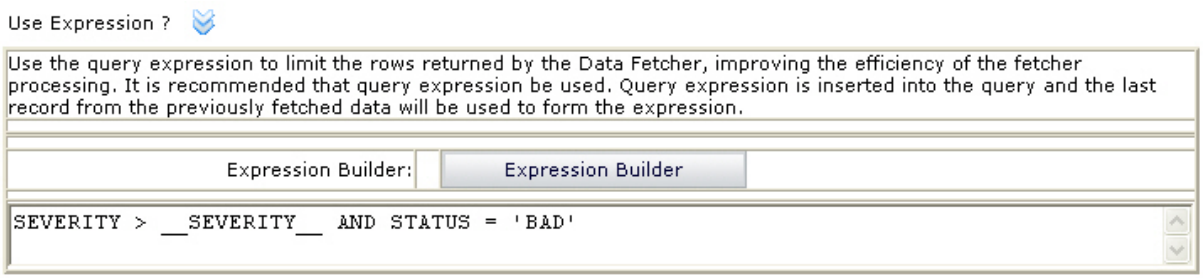


Figure 8. Data Fetcher editor - expression added

13. Click **Save** to save the data fetcher.

Using the data fetcher results

When you have created the data fetchers you need, you can use the data to create incoming status rules and auto-population rules in your service templates.

Chapter 12. Service templates

This section explains how to build and edit service templates in IBM Tivoli Business Service Manager (TBSM).

Service templates overview

This topic describes service templates.

To simplify the process for defining and configuring services, the TBSM console provides tools to help you create service templates for services instances with common behavior. Rather than define each of your services and their dependencies individually, you can create one template for a type of service and then assign applicable services to the template.

Note: If you change rule or template names, you can cause issues with other features of TBSM. TBSM uses these names to reference the rule values for other rules, such as auto-population or ESDA rules and for custom displays such as service trees and view definitions.

If you want to change a rule or template name, change the name before you use the rule or template elsewhere in TBSM.

For example, when you create numeric and text-based rule, you need to create a custom service tree or view definition to display the output of the rule in the TBSM console. The trees and view definitions use the template and rule names to map the rule values to a display object. As a result, if you change the name of a rule or a template used in a custom display, the custom tree or view definition will no longer display the rule values properly.

Service instances

This topic describes services instances and service instance tagging.

You can create a template for physical components such as servers, or abstract components such as webservers. You can also configure templates to monitor the ObjectServer for enriched events from other products such as Netcool/Impact.

Service instances represent actual services that are assigned a template. The template defines how a service responds to incoming data and the status of other services. You should assign services of the same type to a common template. This allows you to use the same template rules to evaluate the status of multiple services.

Service instance tagging

When you assign a template to a service, you *tag* the service with the template. Templates eliminate the necessity of creating the same rules for a service type more than once.

For example, `webserver1` and `webserver2` are services tagged with the `WebServer` template. The status of all services tagged with the `WebServer` template are evaluated using the rules defined in the `WebServer` template.

Service template model: Example 1

This topic describes an example template model, templates with incoming status rules, and templates with aggregation rules.

You can create dependencies for each service type. Figure 9 on page 50 shows a simple service template model example that explains how template dependencies work. This model contains a hierarchy of service templates where the highest level service template depends on the status of the subservices below it. In this case, the status of the services is based on ObjectServer events.

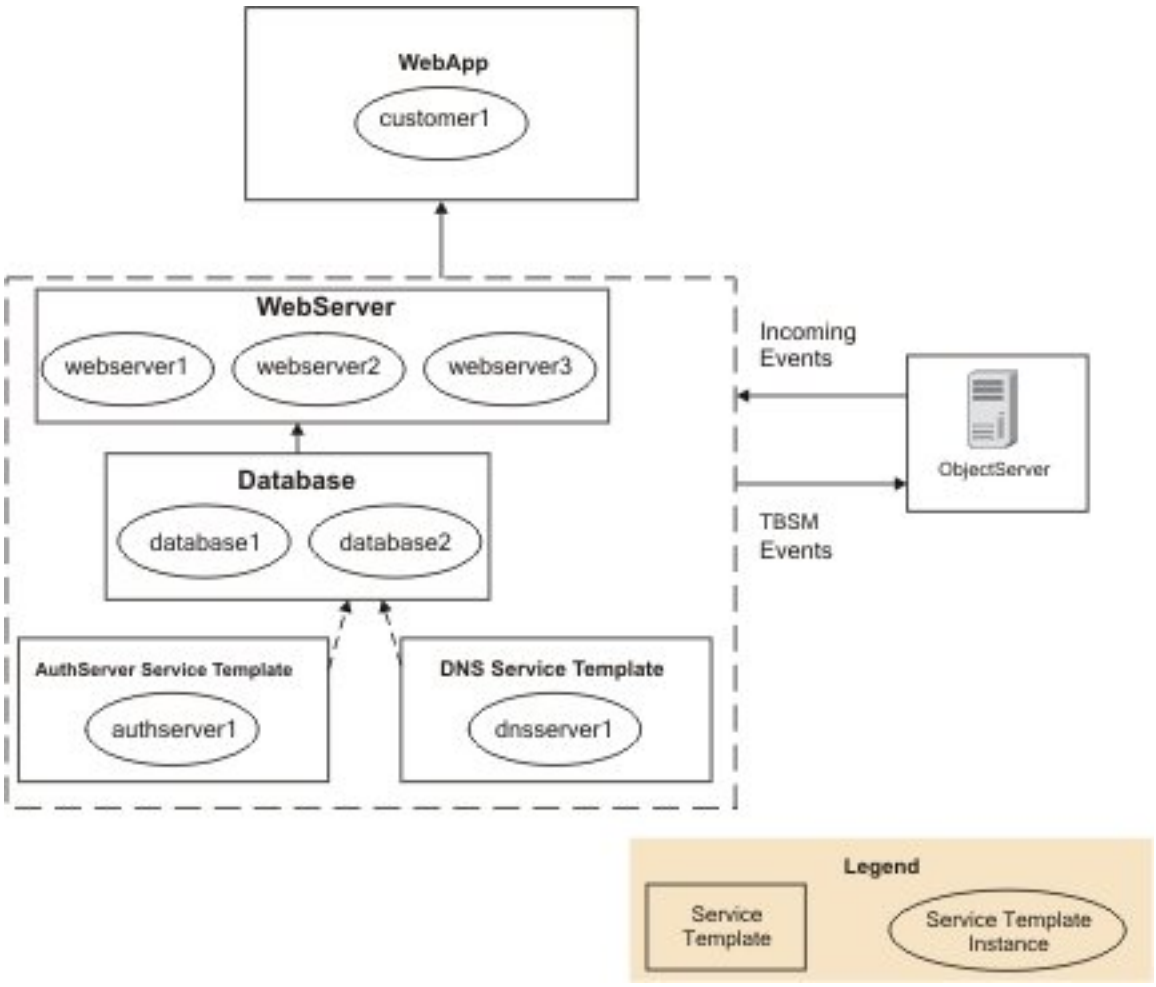


Figure 9. Simple service model example 1

Table 13 on page 50 describes the dependencies for the example service model in Figure 9 on page 50.

Table 13. Service model dependencies (1 of 2)

Template	Service instances	Dependent rules	Incoming status rules
`WebApp`	`customer1`	Status of `Webserver` services.	None.

`WebServer`	`webserver1`
	`webserver2`
	`webserver3`
		Status of `Database` services.	Netcool/OMNIbus events as defined in `WebServer` template.

`Database`	`database1` `database2`	Status of `DNSServer` services. Status of `AuthServer` services.	Netcool/OMNIbus events as defined in `Database` template.
`DNSService`	`dnsserver1`	None.	Netcool/OMNIbus events as defined in `DNSService` template.
`AuthServer`	`authserver1`	None.	Netcool/OMNIbus events as defined in the `AuthServer` template.

Templates with Incoming Status rules

The service templates in Figure 9 on page 50 within the dotted line rectangle include rules that evaluate events from Netcool/OMNIbus. The dependencies for these templates are based on the incoming events. These dependencies are called incoming status rules. For example, the templates for the DNS and the authentication servers evaluate service-related events from Netcool/OMNIbus. The status of these two servers affects the status of the two databases and the three web servers because they require these two servers to function properly. Accordingly, you could add both incoming and aggregation rules to the `Database` and `WebServer` templates.

Templates with Aggregation rules

The status of the database servers affects the status of the web servers. The status of all these servers, in turn, affects the status of service instances tagged with the `WebApp` service template. The status of services tagged with the `WebApp` template are based on the status of the services tagged with the other templates in the service model, such as `Database` and `AuthServer`. `WebApp` does not depend on the incoming events directly. These dependencies are defined with aggregation rules within the template. These rules aggregate the status of multiple child services to determine the status of the parent service.

You need to model these types of dependencies for your network in the TBSM console to monitor the service status in the Service Editor.

Service model: example 2

This topic describes an example service hierarchy, the dependencies for the example service hierarchy, and the service hierarchy displayed in the View Service tab of the Service Editor/Viewer.

Figure 10 on page 52 shows another way you could create a service hierarchy for a similar service. In this example, the status of the services depends on both ObjectServer events and data from a DNS SQL data source.

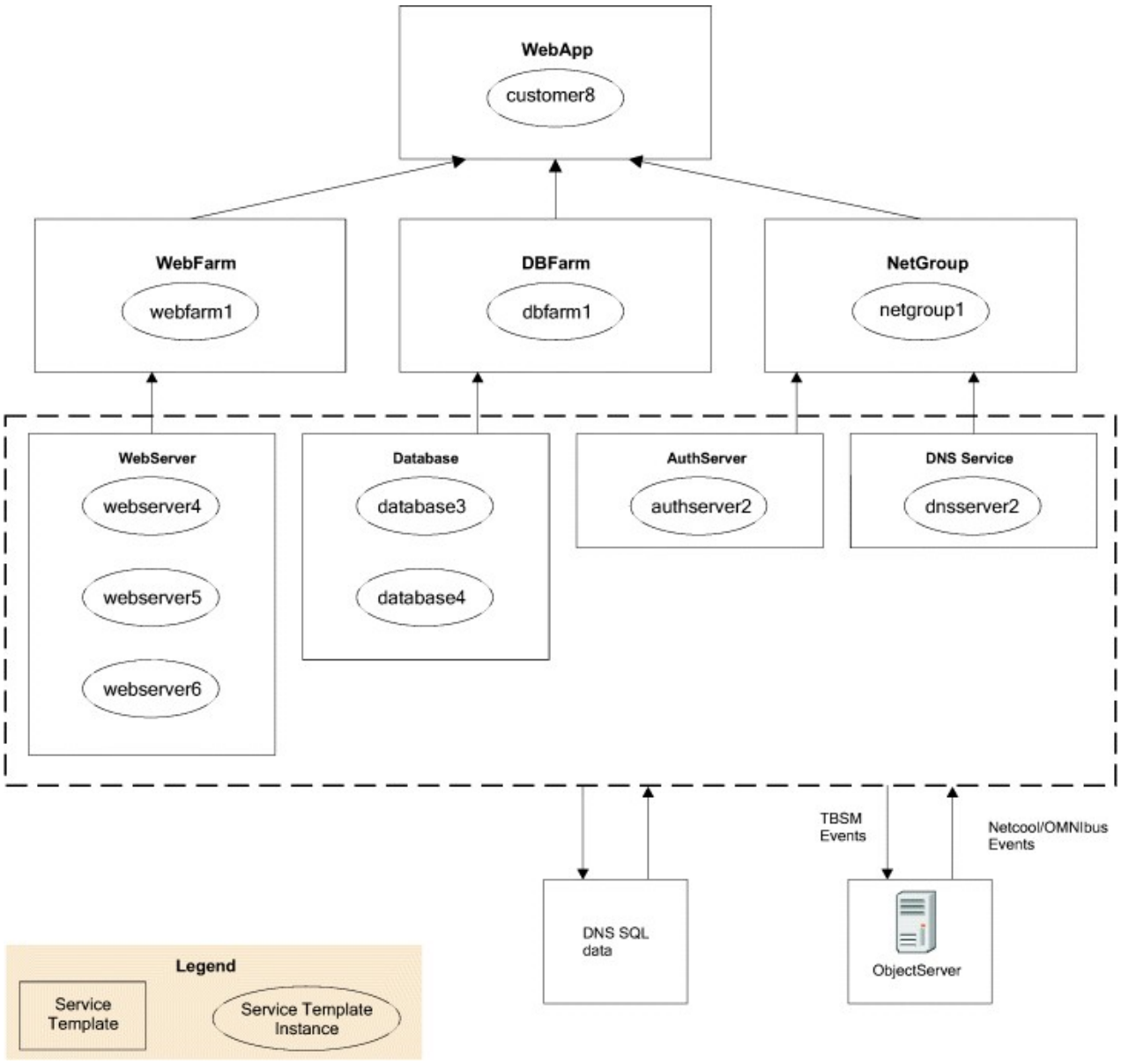


Figure 10. Example service hierarchy

Table 14 on page 52 describes the dependencies for the example service model in Figure 10 on page 52.

Table 14. Service model dependencies

Template	Service instances	Aggregation rules	Incoming status rules
`WebApp`	`customer8`	Status of `WebFarm`, `DBFarm`, and `NetGroup` services	None
`WebFarm`	`webfarm1`	Status of `Webserver` services	None

`WebServer`	`webserver4`
	`webserver5`
	`webserver6`
		None	Netcool/OMNIbus events for web servers 4, 5, and 6

`DBFarm`	`dbfarm1`	Status of `Database` services	None.

Table 14. Service model dependencies (continued)

Template	Service instances	Aggregation rules	Incoming status rules
Database	database3 database4	None	Netcool/OMNIbus events for databases 3 and 4
NetGroup	netgroup1	Status of DNSServer services Status of AuthServer services	None.
DNSService	dnserver2	None	Data from a database that contains DNS availability information for dnsserver2
AuthServer	authServer2	None	Netcool/OMNIbus events for authserver2

In Figure 10 on page 52, the provider has created service templates for:

- WebApp (with a service called customerB) that depends on the status of services tagged with the WebFarm, DBFarm and NetGroup templates as defined in the WebApp template's aggregation rules.
- WebFarm (with a service called webfarm1) that depends on the status of services tagged with the WebServer service template as defined in the WebFarm template's aggregation rules.
- WebServer (with services webserver4, webserver5, and webserver6) that gets its status from Netcool/OMNIbus events as defined in the template's incoming status rules.
- DBFarm (with a service called dbfarm1) that depends on the status of services tagged with the Database service template as defined in the DBFarm template's aggregation rules.
- Database (with services database3 and database4) that gets its status from Netcool/OMNIbus events as defined in the template's incoming status rules.
- NetGroup (with a service called netgroup1) that depends on the status of services tagged with the Authserver and DNSserver service template as defined in the NetGroup template's aggregation rules.
- Authserver (with a service called authserver2) that gets its status from Netcool/OMNIbus events as defined in the template's incoming status rules.
- DNSserver (with one instance called dnsserver2) that gets its status from an SQL query into a database with information about DNS events.

Service hierarchy example 2 in Service View

Figure 11 on page 54 shows how the service would look in the TBSM **Service Editor** window.

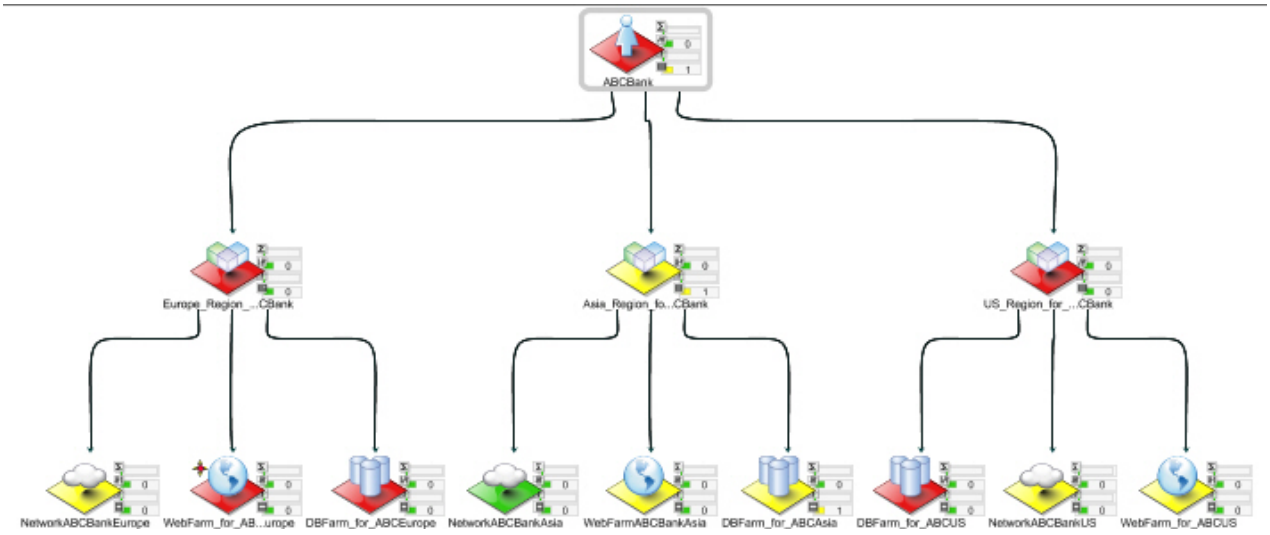


Figure 11. Alternate service model in Service Editor: View Service

For more examples of service models that include external databases, see the *TBSM Scenarios* guide.

Service template configuration

When you know your service hierarchy, create a template for each service type in TBSM. This section describes how to create and modify service templates.

Opening the Edit Template tab

This topic describes the elements of the Edit Template tab in the Service Editor.

You create and edit service templates in the **Edit Template** tab of the Service Editor.

To create a new service template, click the **Create New Template** button from the toolbar in **Templates** in the Service Navigation portlet.

To edit an existing service template, click a template name in **Templates** in the **Service Navigation** portlet; then click the **Edit Template** tab for the template in the Service Editor.

Table 15 on page 54 describes the elements of the **Edit Template** tab.

Table 15. Edit Template tab elements

Element	Description	Reference
Toolbar	The toolbar options let you save and refresh the service template settings. You can also display online help.	"Edit tabs" on page 29
Template Properties	Template Properties fields let you set the basic service template information and choose an icon to represent the template in the console.	"Basic service template properties" on page 55
Rules tab	The settings in this tab let you configure the service incoming status and aggregation rules for a service template. You can also configure auto-population and ESDA rules for the template.	Chapter 13, "Incoming status rules," on page 67 Chapter 14, "Aggregation and numerical formula rules," on page 81 Chapter 15, "Auto-population rules," on page 93 Chapter 17, "ESDA rules," on page 111

Table 15. Edit Template tab elements (continued)		
Element	**Description**	**Reference**
Tagged Services tab	The settings in this tab let you tag services with the settings in the service template.	“Tagging services” on page 58
Output Expressions tab	The settings in this tab let you combine the status of multiple rules to get the status of a service.	“Output expressions” on page 59
SLA tab	Optional. The settings in this tab let you configure your SLA settings.	Chapter 23, “Service level agreement configuration,” on page 165
Additional tab	Optional. The settings in this tab let you set an hourly SLA penalty for templates with SLA settings and add custom properties to the service template. For example, you can create an additional attribute that gets its value from a field in a database row.	Chapter 23, “Service level agreement configuration,” on page 165 “Additional tab” on page 62
Security tab	The settings in this tab let you assign user and group permissions for viewing and editing the service template. You only see this tab if you have the `tbsmTemplateAdmin` user role for the service template.	Chapter 21, “Granting user and group permissions to templates and services,” on page 145

Basic service template properties

This topic describes template properties, how to select a display icon, and how to set template properties.

The Template Properties section of the **Edit Template** tab lets you specify the following information:

- Unique service **Template Name**

  Note: Service template names must not contain spaces. Otherwise, these objects may not be saved properly in the TBSM database.
- Text **Description** of the service template
- Display icon for service template
- Advanced options - Edit Properties

Note: Note: The Name field for templates, rules, data sources, maintenance schedules, view definitions and other objects must not contain these special characters:

```
" & < > \ / * ? | : $ ! % , .
```

Names must not contain spaces. Otherwise, these objects may not be saved properly in the TBSM database. If the object does not save, remove any special characters or spaces from the name field.

Selecting a display icon

To select a display icon for a service template, complete the following steps:

1. Click **Browse** for the **Display Icon:** field.

   The **Browse Icons** window opens.
2. Click the icon you want to display for this service template and the services tagged with the template.

   The **Browse Icons** window closes and the new icon displays in the **Display Icon** field.
3. Click the **Save** button in the toolbar.

Setting template properties

You can set standard properties for templates that have auto-population and ESDA rules. You can make the data permanent and you can specify when permanent data should be deleted if the system does not detect a change in the data within a specified time period.

To set the standard properties for a template, complete the following steps:

1. In the **Edit template** tab, click the `Edit Properties` link next to the **Properties** field.

   The **Standard Template Properties** window opens.
2. For more information about standard properties, see "Service persistence settings" on page 124

   .

Rule configuration

This topic describes the elements of the Rules tab in the Service Editor.

The settings in this tab let you configure the incoming status, aggregation, auto-population, and ESDA rules for a service template. To edit rules:

1. Click the **Rules** tab in the Service Editor.
2. Table 16 on page 56 describes the **Edit Template Rules** tab elements.

Table 16. Rules tab elements

Tab element	Element name	Description
	Select All button	Click **Select All** to select all the rules for deletion.
	Deselect All button	Click **Deselect All** to deselect all the rules selected for deletion.
	Delete button	Click **Delete** to delete the selected rules.
Status rules buttons See Chapter 13, "Incoming status rules," on page 67 for more information about these rules.		
	Create Incoming Status Rule button	Click **Create Incoming Status Rule** to open the **New Incoming Status Rule Type** window. Enter the data that maps ObjectServer events or other data to services which are tagged with this template.
Children buttons See Chapter 14, "Aggregation and numerical formula rules," on page 81 for more information about these rules.		
	Create Good, Marginal, Bad Aggregation Rule button	Click **Create Good, Marginal, Bad Aggregation Rule** to open the **Edit Good, Marginal, Bad Aggregation Rule** window. Enter the data that creates a status aggregation on another service template. That is, the status of every service you tag with this template depends on the status of other services which are tagged with the template you select in the **Good, Marginal, Bad Aggregation Rule** window.

Table 16. Rules tab elements (continued)		
Tab element	**Element name**	**Description**
	Create Numerical Aggregation Rule button	Click **Create Numerical Aggregation Rule** to open the **Edit Numerical Aggregation Rule** window. Numerical aggregation rules are used by TBSM to aggregate rule output values of the same type across a set of child service instances.
	Create Numerical Formula Rule button	Click **Create Numerical Formula Rule** to open the **Edit Numerical Formula Rule** window. Numerical formula rules are a type of rule used by TBSM to combine rule output values of different types within the same service instance. These output values can be generated by incoming numerical status rules or by numerical aggregation rules.
Additional rule types		
	Create Auto-population Rule button	Click **Create Auto-population Rule** to open the **Create Auto-population Rule** window where you can configure services automatically based on data from an incoming status rule. See Chapter 15, “Auto-population rules,” on page 93.
	Create ESDA Rule button	Click **Create ESDA Rule** to open the **ESDA Model Rule** window. External Service Dependency Adapters (ESDAs) let TBSM dynamically import service hierarchies from any data source you can configure for TBSM. See Chapter 17, “ESDA rules,” on page 111. ESDA rule will be disabled by default in the **Rules** tab. To enable the rule, edit the following file in the Dashboard server: `/opt/IBM/JazzSM/profile/installedApps/JazzSMNode01Cell/isc.ear/sla.war/etc/RAD_sla.props` Set `impact.esda.enable=true`
Rule Description Columns		
Select Column	Select column	Select the check boxes in this column to mark rules for deletion.
Type column		This column shows the type of rule listed in the **Rule Name** column. The multi-colored icon indicates an incoming status rule and the single-colored icon with three circles indicates an aggregation or formula rule.
Rule Name column		This column lists all the rules configured for the service template. If you want to edit the rule, click the rule name to display the appropriate rule editor window.

Table 16. Rules tab elements (continued)		
Tab element	**Element name**	**Description**

Rule Settings column		This column shows a summary of the rule attributes for each rule.
		Incoming status rules have summaries like:
		`Class: Default Class(0)`
		`Bad When: (AlertKey = 'Webserver') AND (Severity >= 5)`
		Aggregation rules have summaries like:
		`Bad When: Any AuthServer child is in a Bad state or worse.`

Note: The Name field for rules, data sources, maintenance schedules, view definitions and other objects must not contain these special characters:

```
" & < > \ / * ? | ( ) : ; $ ! %
```

Names must not contain spaces. Otherwise, these objects may not be saved properly in the TBSM database. If the object does not save, remove any special characters or spaces from the name field.

Note: If you change rule or template names, you can cause issues with other features of TBSM. TBSM uses these names to reference the rule values for other rules, such as auto-population or ESDA rules and for custom displays such as service trees and view definitions.

If you want to change a rule or template name, change the name before you use the rule or template elsewhere in TBSM.

For example, when you create numerical and text-based rule, you need to create a custom service tree or view definition to display the output of the rule in the TBSM console. The trees and view definitions use the template and rule names to map the rule values to a display object. As a result, if you change the name of a rule or a template used in a custom display, the custom tree or view definition will no longer display the rule values properly.

Tagging services

This topic describes the elements of the Tagged Services tab and how to tag services to a template.

After you create a service template, you can assign the template to a given service in your model. When a service has a template assigned, the service is tagged with the template. When a service is tagged with a service template, the template determines how the service responds to incoming data and the status of other services. To tag services with a service template:

1. From the Service Editor **Edit** tab, click the **Tagged Services** tab.
2. Select the Services you want to tag using Table 17 on page 58 as your guide.

Table 17. Tagged Services tab elements	
Element	**Description**
Search the Available Services field and **Search** button	Enter a string in the search field and click **Search** to search for services that contain the string you entered. The system displays the search results in **Available Services** list. For example, if you want to find all the services with the letters *serv* in their name, enter `serv` in the field and click **Search**.

Table 17. Tagged Services tab elements (continued)	
Element	**Description**
Available Services list	This list shows all the services you have permission to edit in your system. You can press **Ctrl** or **Shift** while you click to select multiple services in this list. Services from this list are added to **Selected Services** list.
Maximum results	**Maximum results** specifies the maximum number of services listed in the **Available Services** list. The default is 100.
>> Add button **<<** Remove button	Highlight a service or services and click the **>>** button to move services from the **Available Services** list to the **Selected Services** list. Click the **<<** button to remove the highlighted services back to the **Available Services** list.
Selected Services list	This list shows all the services that have been tagged with the service template. You only see the services you have permission to view in the system. You can use press **Ctrl** or **Shift** while you click to select multiple services in this list.

Output expressions

This topic describes the Output Expressions tab in the Service Editor. This topic also describes output expression parameters.

You can use the **Output Expressions** tab to create special rules to determine the status of a service. The settings in this tab let you combine the status of multiple rules to get a different output status of a service tagged with the template. You can also specify an unknown status if a given service does not receive events for a specified time period.

Output expressions parameters

When you create an output expression, you specify an expression as described in Table 18 on page 59. The **Output Expressions** tab contains a drop-down list for each of these parameters. For more information about using these fields, see “Creating and editing output expressions” on page 61.

Table 18. Output expression parameters	
Parameter drop-down list	**Description**
Rule	The incoming status and or aggregation rules configured for the template.
Value and State Change	You can select either `value` or `LastStateChange` for this parameter. `Value` refers to the status of the rule: either Bad or `Marginal`. `LastStateChange` refers to the time of the latest state change for the rule in UNIX time. This parameter is useful if you want to determine which rule first changed a service's status to bad.
Comparison operators	You can select the following comparison operators: • = • >= • <= • > • <

Table 18. Output expression parameters (continued)	
Parameter drop-down list	**Description**
Rule status	You can select either Bad or `Marginal` rule status. You can specify good status by selecting < `Marginal` status.
Boolean operator	You can select either `AND` or `OR` from this field.

If you never want a service status to change to Bad or `Marginal`, enter `False` in the **Bad** or **Marginal Output Expression** field.

You can use TBSM functions to parse the values in the expression. For more information about using TBSM functions, see the *TBSM Customization Guide*.

Example output expressions

This topic describes a value output expression example and a LastStateChange output expression example.

This section contains example expressions using both the `Value` and `LastStateChange` parameters.

Value output expression example

In this example, the template `DBFarm` has two aggregation rules:

- The `NetGroupWorst` rule changes the `DBFarm` status to Bad or `Marginal` if a `NetGroup` service's status changes to bad or `marginal`.
- The `DBWorst` rule changes the `DBFarm` status to bad if any database service's status changes to bad or `marginal`.

To create a rule that sets the `DBFarm` status to change to bad when both `NetGroupWorst` rule and `DBWorst` rule report a `marginal` status, enter the following value in the **Bad Output Expression** field:

```
(NetGroupWorst.Value = Marginal) AND (DBWorst.Value=Marginal)
```

LastStateChange output expression example

In this example the template `WebFarm` has two aggregation rules:

- The `NetGroupWorst` rule changes the `WebFarm` status to bad or `marginal` if a Netgroup service's status changes to bad or `marginal`. The Netgroup Services include DNS and Authentication servers, and in this case also part of an enterprise service-dependency model.
- The `WebserverWorst` rule changes the `WebFarm` status to bad if any web server's status changes to bad or `marginal`.

The `WebFarm` template's main purpose is to monitor web servers, but you still have a `NetGroup` dependency to help identify a network problem. However, you do not want a `WebFarm` service status to change to bad when there are network problems, since the web server's processes are still running when the network has troubles.

If you want to distinguish between outages caused by the network group and outages caused by a web server, you create an expression using the `LastStateChange` parameter that determines which service failed first: a web server, or a network service.

In this example, the `WebFarm` status changes to bad when:

- The `WebserverWorst` rule status is Bad AND the `WebserverWorst` status changed before the `NetGroupWorst` status changed to bad

  or
- The `WebserverWorst` rule status is Bad AND the `NetGroupWorst` status is less than bad

To create an expression for this example, enter the following value in the **Bad Output Expression** field:

```
((WebserverWorst.Value = Bad) AND (WebserverWorst.LastStateChange <
NetGroupWorst.LastStateChange)) OR ((WebserverWorst.Value = Bad) AND
(NetGroupWorst.Value < Bad))
```

In the **Marginal Output Expression** field, enter:

```
((WebserverWorst.Value = Marginal) AND (WebserverWorst.LastStateChange <
NetGroupWorst.LastStateChange)) OR ((WebserverWorst.Value = Marginal) AND
(NetGroupWorst.Value < Bad))
```

Creating and editing output expressions

This task describes how to create and edit output expressions.

About this task

To create or edit output expressions, complete the following steps:

Procedure

1. From the **Edit Template** tab of the Service Editor, click the **Output Expressions** tab.

   The **Output Expressions** tab opens.
2. Enter the special rule data using Table 19 on page 61 as your guide.

Table 19. Output Expressions tab elements

Element	Description
Report Unknown Severity if no Events after the following number of seconds check box	When you select this check box, the service status changes to unknown (purple) if the service does not detect events for a given time period. TBSM also sends a severity 1 (indeterminate) event to the ObjectServer. For example, if you set the seconds to 60 and no service events are detected for more than 1 minute, the service status changes to unknown and sends a severity 1 event to the ObjectServer.
Expression builder drop-down lists and **Insert** buttons	These fields enable you to enter output expression parameters in the open text field for both **Bad**-and **Marginal Output Expressions.** The first field lists all the rules for the service template. To use these fields, select the value you want from the drop-down list and click **Insert** to the right of the field. For non-English browsers, translated values will be displayed in the drop down lists, followed by English, displayed in brackets. The **Insert** buttons will insert the English only, as this is required for TBSM processing. The selected value is displayed at the end of the expression. You cannot insert values into the middle of an expression with these fields.

Table 19. Output Expressions tab elements (continued)	
Element	**Description**
Bad Output Expression text field	Type your expression that results in a bad status for the service in this field. That is, the service status is bad if the expression is true. You can also use the expression builder fields to insert the values you want as described in this table. You must enter values for both the **Bad** and **Marginal Output Expression** fields. You can enter `false` in one of the fields if you do not want a service to change states. You can cut and paste text in this field by right-clicking the highlighted text and selecting **Cut**, **Copy**, or **Paste** from the menu. There are also **Select All** and **Undo** options in this menu.
Marginal Output Expression text field	Type your expression that results in a `marginal` status for the service in this field. See **Bad Output Expression** for more information about how to use this text field. You can enter `false` in one of the fields if you do not want a service to change states.

3. When you have completed your expressions, click the **Save** button in the toolbar.

   TBSM performs a syntax check on the expressions before saving and displays an error message if syntax errors are found.

SLA tab

For information about configuring SLA calculation parameters, see Chapter 23, "Service level agreement configuration," on page 165.

Additional tab

This topic describes the Additional tab. This topic also describes Reserved Additional Properties.

The settings in the **Additional** tab are optional. With the default SLA Penalty property, you can set an hourly SLA penalty for templates that have SLA settings configured. The SLA penalty calculations let you set a cost estimate for each hour of downtime caused by an outage. For more information about adding hourly SLA penalties for templates, see "Setting SLA penalty calculations" on page 63.

In this tab, you can also add custom properties. For example, you can create an additional attribute that obtains its value from a field in a database row.

Reserved additional properties

You can add Reserved Additional Properties to a template and set them as a default. Every service assigned to the template will have the additional property. You can also manually set a specific value for the additional property for each service instance. To add additional properties to the template, follow these steps:

1. Click the **New Parameter** button on the toolbar.
2. Enter the parameter name and the default value.

Table 20 on page 63 describes the reserved properties:

Table 20. Reserved additional properties		
Reserved property name	**Sample value**	**Property description**
ViewDefName	AppManagerView	This view definition is used for any template/instance where this value is defined.
MapName	maps/Europe.ivl	This map is used for an template/instance where this value is defined.
WebtopViewName	NewView	This Webtop view name is used in the **Service Details** portlet and the AEL when launching from an instance that is assigned to the template where this value is defined.
UseESDAIdentifiersOnly	True or False	When the property is set to true, all existing event identifiers are deleted from the memory model during each invalidation run. For new identifiers, the service accepts only identifiers that are sent from External Service Dependency Adapter (ESDA) policies or the Service Component Repository (SCR). This means what user will no longer be able to use identifiers set from the TBSM UI, as it will be erased at every invalidation. When the property is set to false, all event identifiers are preserved for use by the service. To delete an event identifier, you must do so manually. The property is set to false by default. **Note:** When enabling this property, it is good practice to invalidate the service tree from the top level node once after saving the template. This is to ensure that the template and identifier changes get propagated correctly throughout the service tree for the services which have the template applied on.

Setting SLA penalty calculations

This topic describes how to set SLA penalty calculations.

About this task

By using the SLA penalty calculations, you can set a cost estimate for each hour of downtime caused by an outage.

You must first save your SLA settings on the **SLA** tab before you can set the SLA penalty calculations.

To set the SLA hourly penalty calculations, complete the following steps:

Procedure

1. From the **Service Editor: Edit Templates** tab, click the **Additional** tab.

   The **Additional** tab opens.
2. Enter the dollar amount for the hourly penalty in the **Default Value** column of the `HourlySLAPenalty` parameter.
3. Click **Save** in the toolbar.

Results

The only parameter that is available is `HourlySLAPenalty`. You can change the default value for this parameter. The parameter and default value only display when SLA parameters have been set for the service template.

Adding security levels

For information about adding security levels, see Chapter 21, "Granting user and group permissions to templates and services," on page 145.

Event enrichment solution

Event enrichment is the process by which Netcool/Impact monitors an event source for new events, looks up information related to them in an external data source and then adds the information to them.

An event enrichment solution consists of the following components:

- A data model that represents the data you want to add to events
- An OMNIbus event reader service that monitors the event source
- One or more event enrichment policies that look up information related to the events and add the information to them

For a sample event enrichment solution, see "Event enrichment tutorial" on page 225.

Configuring Event Enrichment in TBSM

This topic describes how to enable Event Enrichment in TBSM.

Before you begin

Before you can use the Event Reader tool, the event reader and event sources must be created. See these topics for more information:

- "Creating the event source" on page 227
- "OMNIbus event reader service" on page 222

About this task

The configuration changes outlined in this topic ensure that TBSM does not read events until Impact has enriched them. To avoid this issue, configuration changes are necessary to ensure that Impact updates a field in OMNIbus called `RAD_SeenByImpact` after Impact is finished with the event. An additional TBSM restriction added to the TBSMOMNIbusEventReader ensures that events are not read unless `RAD_SeenByImpact = 1`.

Important: If you configure Event Enrichment for a specified Event Reader, that Event Reader must be running for TBSM to read any events from the ObjectServer. If the Impact Event Reader is turned off, TBSM will not read any events from the ObjectServer.

Procedure

1. Click the **Event Enrichment** button on the Service Navigation portlet.

   The **Event Enrichment** window opens.
2. Check the **OMNIbusEventReader** box to enable Event Enrichment for the Impact Service.

   Remember: The Event Reader must be running for the Event Reader checkbox to be active.

   Event readers from a remote Impact server will be listed in this UI with their clustername in parentheses.
3. Click **OK**.

“Event enrichment tutorial” on page 225
The goal of this tutorial is to develop an event enrichment solution to enhance the value of an existing Netcool/Impact installation.

Chapter 13. Incoming status rules

This section contains information about incoming status rules in IBM Tivoli Business Service Manager (TBSM).

Incoming status rules overview

This topic describes incoming status rules.

Incoming status rules are sets of conditions that allow TBSM to obtain general or detailed status parameters from field values in incoming ObjectServer events or other data.

You use incoming status rules as part of a logical model that takes information derived from ObjectServer events or other incoming data and uses it to determine different aspects of the status of a service.

You define incoming status rules when you create or edit a service template. Each template can have any number of associated incoming status rules. You can configure TBSM to display the results of incoming status rules in the **Services Tree** or *Service Editor* features of the application console. This display allows users to view and respond to changes in the general or detailed state of the service in real time.

TBSM has the following types of incoming status rules:

- Good, Marginal, and Bad rules
- Numerical rules
- Text-based rules

Note: If you change rule or template names, you can cause issues with other features of TBSM. TBSM uses these names to reference the rule values for other rules, such as auto-population or ESDA rules and for custom displays such as service trees and view definitions.

If you want to change a rule or template name, change the name before you use the rule or template elsewhere in TBSM.

For example, when you create numerical and text-based rule, you need to create a custom service tree or view definition to display the output of the rule in the TBSM console. The trees and view definitions use the template and rule names to map the rule values to a display object. As a result, if you change the name of a rule or a template used in a custom display, the custom tree or view definition will no longer display the rule values properly.

For more information about mapping rule output values to the **Services Tree** and Service Editor, and examples, see the *IBM Tivoli Business Services Manager: Scenarios Guide*.

Incoming status rules properties and conditions

This reference topic describes these incoming status rule properties and conditions: rule name, data feed name, event discriminators, name expression, filter, and rule output expression.

Incoming status rules have the following properties and conditions:

- Rule name
- Data feed name
- Event discriminators
- Name expression
- Filter
- Rule output expression

Rule name

The rule name is a unique string that identifies the rule. This name is also used to identify the rule output value when referenced elsewhere in TBSM.

Note: You cannot use these special characters in rule names:

```
" & < > \ / * ? | ( ) : ;
```

Otherwise, the rule may not be saved properly in the TBSM database. If the rule does not save, remove any special characters from the rule name.

Description

Using a description will help you stay organized in your TBSM environment. When you mouse-over the rule or Display Name in the TWA, the Description will appear and help you determine if this is the rule or data you want to see.

Display Name

The Display Name will appear in the metric list in the Time Window Analyzer. The Display Name will also appear in the **Rule Name** column in the **Rules** tab in the Service Editor. If no unique name is specified, the name of the rule will be used.

Data feed name

The data feed name specifies which data feed is associated with the rule. When TBSM retrieves events from an ObjectServer or data from another data source, it evaluates only those rules that are associated with that data feed. For SQL data sources, the data feed name is the name of a data fetcher.

Event discriminators

Event discriminators are ObjectServer classes that you can use to pre-filter incoming events to make sure that TBSM only processes incoming events that are relevant to the defined rules. If you are using an ObjectServer as the data feed for the rule, you should specify classes that identify the types of events that are related to the rule. Event discriminators are not supported for non-ObjectServer data feeds.

For example, you can choose `Alcatel 5620 Corba(4950)` to specify that this rule applies to events related to the `Alcatel 5620` probe.

Name expression

The name expression is used by TBSM to determine which service instance is associated with a particular incoming event or other data. The expression can be the name of a field or a statement that uses the TBSM expression syntax to derive the service instance name from one or more fields.

When a rule is processed by the TBSM software, the value of the name expression is based on fields in the incoming event. This name expression is then compared to the rule identification field for each service instance that is tagged with the rule's template.

For example, to specify that the `Node` field in an incoming event contains the name of the related service instance, you can use the following instance name expression:

```
Node
```

When TBSM processes an event associated with this rule, it takes the value of the `Node` field and attempts to locate a service instance that is tagged with this template and that has an equivalent name. For example, if the value of `Node` is `db_server_01`, TBSM looks for the service instance with this name and attaches the output value that results from the rule.

Note: Ensure that you move at least 1 item from the **Available Instance Name Fields** list to the **Selected Instance Name Fields** list. If this **Selected Instance Name Fields** list is empty, TBSM cannot match a Service Instance to an incoming event or other data.

For more information about the expression syntax, see the *IBM Tivoli Business Service Manager: Customization Guide*.

Filter

The filter for metric rules lets you specify additional matching conditions that an incoming event or other data must fulfill in order for the rule to be processed. You must also specify the field that contains the value that TBSM will use to derive the output value as part of the filter.

Rule output expression

The rule output expression is an expression used by TBSM to determine the value that results when the rule is processed. The expression can be either the name of a field or a statement in the TBSM expression syntax. When TBSM evaluates the rule, it assigns either the value of the specified field or the result of the statement to the affected service instance.

Metric Collection

The Metric Collection options enable TBSM to collect and store the metric data for the rule. You can activate the storing of status changes of the rule and display the recent history of the metric. The collected metric data can be displayed in the Time Window Analyzer, and stored in the TBSM database. You can also store status changes of rules to the Tivoli Data Warehouse (TDW) using the BSM Agent. The TDW stores data for historical reporting.

Store data for this rule for TWA
Store the output values for this rule in the TBSM database for the Time Window Analyzer.

Store data for this rule for TDW
Store the output values for this rule in the TDW as a key performance indicator for the BSM Agent Service Indicators attribute group.

Note: If you want to store status changes to the warehouse, you need to install and configure the BSM agent and TDW.

Customizing Data Feed fields

Customizing Fields will enable you to use Impact Services as a Data feed.

When you select an Impact Service as a Data Feed for a rule, the **Customize Field** button appears. Clicking this button enables you to configure the Data Feed fields to match the Impact policy fields. The field names and types are default for Impact Services. You can add fields in the policy, and add the same field names in the rule. The names and types must match and are case sensitive. If they do not match, the rule will not work properly.

Note: You must configure the Impact policy and service before you customize fields in TBSM.

The table below describes the Customized Fields elements:

Table 21.

Element	Description
Select All button 	Click **Select All** to select all the fields.
Deselect All button 	Click **Deselect All** to deselect all the fields.

Table 21. (continued)	
Element	**Description**
Delete button 	Click **Delete** to delete the selected fields.
New button 	Click this button to create a new field. service instance. When you click this button, a blank field appears at the top of the list. Enter the name for the new field.

The table below describes the default field settings:

Table 22.	
Field Name	**Field Type**
`EventClass`	`String`
`EventType`	`String`
`ResourceName`	`String`
`SecondaryResourceName`	`String`
`Summary`	`String`
`Value1`	`Float`
`Value2`	`Float`
`Value3`	`Float`

For more information about configuring Impact policies and Impact Services, see the *IBM Tivoli Netcool/ Impact Solutions Guide*.

Good, marginal, and bad incoming status rules

This topic describes good, marginal, and bad incoming status rules.

TBSM uses Good, Marginal, and Bad incoming status rules to obtain the general state of a service from an incoming ObjectServer event or other data. The general service state is determined by comparing threshold values set in the rule definition to event field values or field values in other incoming data. When this type of rule is processed, TBSM obtains a general service state of `Good`, `Marginal`, or `Bad` and assigns this value to the service instance. If multiple rules of this type are defined in a single service template, the worst current output of all rules is reflected as the general service state.

Good, Marginal, and Bad incoming status rules consist of a set of properties and filters that specify which events or other data contain relevant information and how TBSM can derive the general state from this information. In addition, this type of rule contains a name expression that TBSM uses to determine the affected service instance.

TBSM processes Good, Marginal, and Bad incoming status rules during runtime after it retrieves events from the ObjectServer or data from another data source. For each incoming event or other data set, TBSM checks the currently defined rules and processes the ones that are associated with the event or data type. TBSM then assigns the general state to the service.

By default, the general service state is shown in the **Services Tree** or Service Editor of the application console as a green, yellow, or red light, where green corresponds to `Good`, yellow corresponds to `Marginal`, and red corresponds to `Bad`.

Note: The general service states of Good, Marginal, and Bad correspond to numeric rule output values of 0, 3, and 5 respectively. TBSM uses this numeric value when the resulting service state is combined with other rule-output values using a numerical aggregation rule or numerical formula rule.

For examples of these rules and how to map them to trees and views, see the *TBSM Scenarios Guide*.

Create and edit good, marginal, and bad incoming status rules

This topic describes how to create and edit good, marginal, and bad incoming status rules. This topic also describes properties and conditions for incoming status rules.

About this task

Use the following procedure to create or edit a good, marginal, and bad incoming status rule:

Procedure

1. Click **Incoming Status Rule** on the **Edit Template** tab in the Service Editor.

   The **Select Incoming Status Rule Type** window opens.

   Or, to edit an existing rule, click the name of the good, marginal, and bad rule you want to edit.
2. Select **Based on a Good, Marginal, and Bad Threshold** in the **Select Incoming Status Rule Type** window and click **OK**.
3. The **Edit Incoming Status Rule** window opens.
4. Enter the rule properties and conditions using Table 23 on page 71 as your guide.

Table 23. Edit Incoming Status Rule window: good, marginal, bad rule properties and conditions

Window element	Description
Rule Name field	The name of the good, marginal, and bad rule.
Description	Using a description will help you stay organized in your TBSM environment.
Display Name	The Display Name will appear in the metric list in the Time Window Analyzer. The Display Name will also appear in the **Rule Name** column in the **Rules** tab in the Service Editor. If no Display Name is specified, the name of the rule will be used.
Data Feed list	The data feed is the ObjectServer or data fetcher name that contains incoming events or other data used by TBSM when processing the rule. You can customize data fields to use Impact services. For more information, see the "Customizing Data Feed fields" topic.
Preview Data button	You can display events or other data that match the conditions specified in this window by clicking the **Preview Data** button. When you click this button, an event table is displayed in a new browser window. This table has the same format as the event table shown in the **Service Details** portlet. You must specify any event discriminators and filters before you will see matching data. For SQL data sources, you can view the selected data in the Data Fetcher **View Data** window by clicking the **View Data** button. Filter criteria, if specified, will affect the data displayed in the window. If you open the Data Fetcher window, the previewed data would be unfiltered.
Event Discriminators list	If you are using an ObjectServer data feed, you can select one or more ObjectServer classes from this list. You use event discriminators to specify which event types are relevant to the rule.

Table 23. Edit Incoming Status Rule window: good, marginal, bad rule properties and conditions (continued)	
Window element	**Description**
Instance name list	You can select one or more event fields from this list. TBSM uses the value of this field to determine the affected service instance when processing this rule. Alternatively, you can use the **Advanced** button as described below to specify an instance expression.
Advanced button	As an alternative to specifying an instance name field as described above, you can click this button to specify an instance expression. The instance expression allows you to specify multiple field names, to manipulate values, or both in the field using statements and functions in the TBSM expression syntax.
Store data for this rule for TWA checkbox	If you select this option, TBSM will collect and store status changes for the rule. This data is stored in the TBSM database. The collected metric data can be displayed in the Time Window Analyzer.
Store data for this rule for TDW checkbox	If you select this option, TBSM will collect and store the status changes of the rule to the warehouse (TDW) using the BSM Agent. The TDW stores data for historical reporting. **Note:** You need to install and configure the BSM agent and the TDW to use this option.
Automatic Roll-up button	If you use the Auto-Rule roll-up feature, you can automatically create rules in other service templates that contain the values from the incoming status rule. That is, you pass the rule output values from the incoming status rule to a rule in another service template.

5. Choose from available filter fields and specify threshold values and corresponding general service states.

   The **Available Filter Fields** list in this section of the window contains the fields in the incoming events (or other data provided by the data feed). You can choose any number of fields from this list. The **Threshold Filter**s table allows you to set one or more threshold values for each of the selected fields and a corresponding service state of Good, `Marginal`, or Bad for each threshold.

   When TBSM evaluates the rule, it compares the value of each selected field with the threshold value that you specified. The worst resulting service state of all comparisons then becomes the general state of Good, `Marginal`, or Bad for the service instance.
6. Click **OK**.

Numerical incoming status rules

This topic describes numerical incoming status rules.

Numerical incoming status rules are a type of rule used by TBSM to obtain a numeric value from fields in ObjectServer events or other data and then to associate this value with a service instance as a rule output value. The resulting rule output value is stored in memory and identified by the rule name where referenced elsewhere in TBSM.

Numerical incoming status rules consist of a set of properties and filters that specify which events or other data contain relevant information and how TBSM can derive the rule output value from this information. In addition, this type of rule contains a name expression that TBSM uses to determine the affected service instance. Optionally, numerical incoming status rules can also set the general state of the service to Good, `Marginal`, or Bad, depending on thresholds specified for the rule output value.

TBSM processes numerical incoming rules during runtime after it retrieves events from the ObjectServer or data from another data source through a data feed. For each incoming event or other data set, the application checks the currently defined numerical rules and processes the numerical rules that are associated with the event or data type. This check includes deriving the rule output value from field values and evaluating the instance name expression to determine the affected service instance. After the rule is processed, TBSM assigns the output value to the affected service.

You can configure TBSM to display the output values associated with a service instance as part of the Service Editor. You can also aggregate or combine the output values within a single service or across services using numerical aggregation and formula rules. In other words, these incoming status rules produce numerical values that you can use in your numerical aggregation rule calculations.

When you create the rule, you can enable TBSM to collect and store metric data using the Metric Collection options. You can activate the storing of status changes of the rule and display the recent history of the metric. The collected metric data can be displayed in the Time Window Analyzer, and stored in the TBSM database. You can also store status changes of rules to the warehouse (TDW) using the BSM Agent. The TDW stores data for historical reporting and must be installed and configured before you want to use this option.

You can define the aggregation rules automatically directly from the incoming status rule. Internally, the same aggregation rules is created. Externally, the aggregation rules can be directly edited on service templates after they are created automatically. TBSM automatically names aggregation status rules based on the name of the incoming status rule and template that the aggregation belongs to. However, you will not be allowed to add a template to the list if the name of the automatically generated aggregation rule would be too long.

The rules can also be managed directly by editing each service template. If you change an automatically generated aggregation rule yourself, the "automatic roll-up" function may no longer display the service templates. If the names of the incoming status rules or service templates change, the automatically created aggregation rules will still work but are not displayed if you press the **Automatic Roll-up** button for this incoming status rule.

To have more control over rolling up numerical status rules, you might need to create your own aggregation rules.

Create and edit numerical incoming status rules

This topic describes how to create and edit numerical incoming status rules. This topic also describes the properties and conditions of numerical incoming status rules.

About this task

To create or edit a numerical incoming status rule, complete the following steps:

Procedure

1. From the **Edit Template** tab in the Service Editor, click the **Incoming Status Rule** button.

   The **Select Incoming Status Rule Type** window opens.

   Or, to edit an existing rule, click the name of the numerical rule you want to edit.
2. Select **Based on a Numeric Value** in the **Select Rule Type** window and click **OK**.
3. The **Edit Incoming Status Rule** window opens.
4. Enter the rule properties and conditions using Table 24 on page 73 as your guide.

Table 24. Edit Incoming Status Rule window: numerical rule

Window element	Description
Rule Name field	The name of the numerical rule.

Table 24. Edit Incoming Status Rule window: numerical rule (continued)	
Window element	**Description**
Data Feed list	The data feed that contains incoming events or other data used by TBSM when processing the rule.
Preview Data button	You can display events that match the conditions specified in this window by clicking the **Preview Data** button. When you click this button, an event table is displayed in a new browser window. This table has the same format as the event table shown in the **Service Details** portlet. For SQL data sources, you can preview the selected data in the Data Fetcher **View Data** window by clicking the **View Data** button.
Event Discriminators list	If you are using an ObjectServer data feed, you can select one or more ObjectServer classes from this list. You use event discriminators to specify which event types are relevant to the metric rule.
Instance name list.	You can select an event field from this list. TBSM uses the value of this field to determine the affected service instance when processing this rule. Alternatively, you can use the **Advanced** button as described below to specify an instance expression.
Advanced button	As an alternative to specifying an instance name field as described above, you can click this button to specify an instance expression. The instance expression allows you to specify multiple field names, to manipulate values, or both in the field using statements and functions in the TBSM expression syntax.
Metric Collection.	Enable TBSM to collect and store the metric data for the rule in the TBSM database or Tivoli Data Warehouse (TDW). **Store data for this rule for TWA** Store the output values for this rule in the TBSM database for the Time Window Analyzer. **Store data for this rule for TDW** Store the output values for this rule in the TDW as a key performance indicator for the BSM Agent Service Indicators attribute group. If you want to store status changes to the warehouse, you need to install and configure the BSM agent and TDW.
Automatic Roll-up button	If you use the Auto-Rule roll-up feature, you can automatically create rules in other service templates that contain the values from the incoming status rule. That is, you pass the rule output values from the incoming status rule to a rule in another service template.

5. Choose from available filter fields and specify the matching conditions.

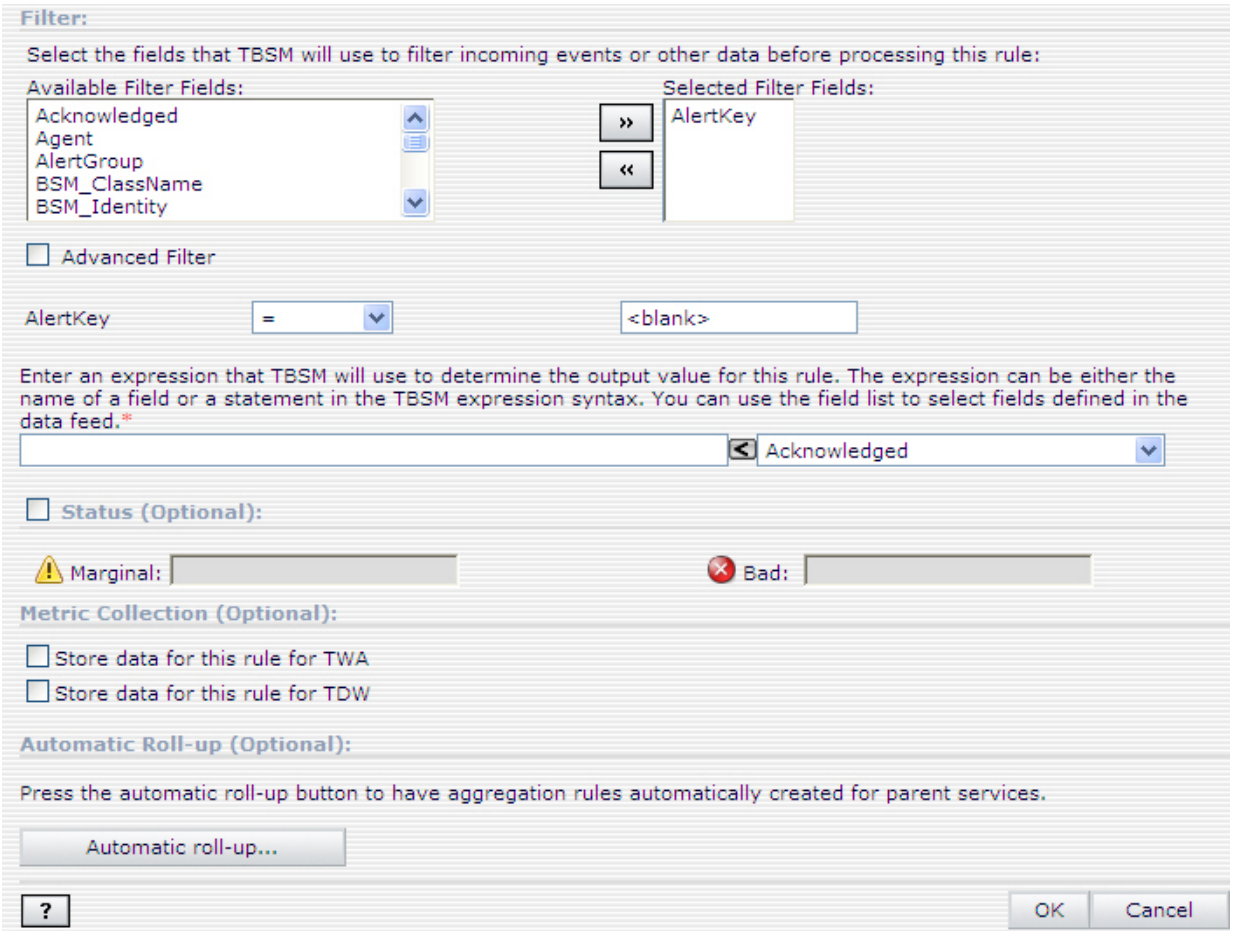


Figure 12. Edit Incoming Status Rule window: numerical rule

The **Available Filter Fields** list in this section of the window contains the fields in the incoming events (or other data provided by the data feed).

In Figure 12 on page 75, TBSM processes the rule for a service where the value of the `lowlevelserviceid` field equals `WebFarm` and the value of the `severity` field equals 5. When these filter conditions are met, the output value for the rule equals the value of the `opentickets` field.

You can choose any number of fields from this list. The lists and text fields below allow you to specify the condition that each field must match in order for TBSM to process this event using the rule.

6. Enter the rule output expression in the text field below. The expression can be either the name of a field or a statement in the TBSM expression syntax. You can use the field list to select fields defined in the data feed.
7. (Optional) If you want the numerical rule to also set the general state of the affected service instance, select **Status**, and enter threshold values for the two service states in the **Marginal** and **Bad** fields. If the **Marginal** threshold is lower than the **Bad** threshold, the values you set are treated as the maximum allowed values. Therefore, when the value returned by the rule exceeds the values you have set, the status changes. However, if the **Marginal** threshold is set to a value higher than the **Bad** threshold, the values you set are treated as the minimum allowed values. In this instance, when the value returned by the rule drops below these settings, the status is changed. For example, if the **Marginal** threshold was set to 70 and the **Bad** threshold was set to 90, the status returned worsens as the value returned increases, that is to say below 70 would provide a status of **Good**, between 70 and 90 would result in a status of **Marginal** and above 90 would result in a status of **Bad**. Alternatively, if the **Marginal** threshold was set to 90 and the **Bad** threshold was set to 70, the status returned worsens as the value returned decreases, that is to say above 90 would provide a status of **Good**, between 70 and 90 would result in a status of **Marginal** and below 70 would result in a status of **Bad**.

8. Click **OK**.

Text-based incoming status rules

This topic describes text-based incoming status rules.

Text-based incoming status rules are a type of rule used by TBSM to obtain a text value from fields in ObjectServer events or other data and then to associate this value with a service instance as a rule output value. Incoming status rules based on text values let you extract text strings from events or SQL data sources. The expression is evaluated as a string value and assigned to the rule. The resulting rule output value is stored in memory and identified by the rule name where referenced elsewhere in TBSM.

Text-based incoming status rules consist of a set of properties and filters that specify which events or other data contain relevant information and how TBSM can derive the rule output value from this information. In addition, this type of rule contains a name expression that TBSM uses to determine the affected service instance. Optionally, text-based incoming status rules can also set the general state of the service to Good, `Marginal`, or Bad, depending on thresholds specified for the rule output value. For example, if the value of the expression contains the string configured in the Bad status box then the service will be Bad. If the value of the expression contains the string configured in the Marginal status box then the service will be `Marginal`.

TBSM processes text-based incoming rules during runtime after it retrieves events from the ObjectServer or data from another data source through a data feed. Like the other incoming status rules, the text-based rules use a naming expression to map events or other data to a specific service instance. For each incoming event or other data set, the application checks the currently defined text-based rules and processes the text-based rules that are associated with the event or data type. This check includes deriving the rule output value from field values and evaluating the instance name expression to determine the affected service instance. After the rule is processed, TBSM assigns the output value to the affected service.

You can configure TBSM to display the output values associated with a service instance as part of the Service Editor. You can also aggregate or combine the output values within a single service or across services using text-based policy rules. In other words, these incoming status rules produce text values that you can use in your text aggregation rule calculations.

For more information about the configuration and policy dependency for propagation of text-based rules, see the *IBM Tivoli Business Service Manager: Customization Guide*.

Create and edit text-based incoming status rules

This topic describes to create and edit text-based incoming status rules. This topic also describes the properties and conditions of text-based incoming status rules.

About this task

To create or edit a text-based incoming status rule, complete the following steps:

Procedure

1. From the **Edit Template** tab in the Service Editor, click the **Incoming Status Rule** button.

   The **Select Incoming Status Rule Type** window opens.

   Or, to edit an existing rule, click the name of the rule you want to edit.
2. Select **Based on Text Value** in the **Select Rule Type** window and click **OK**.
3. The **Edit Incoming Status Rule** window opens.
4. Enter the rule properties and conditions using Table 25 on page 77 as your guide.

Table 25. Edit Incoming Status Rule window: text-based rule	
Window element	**Description**
Rule Name field	The name of the rule.
Data Feed list	The data feed that contains incoming events or other data used by TBSM when processing the rule.
Preview Data button	You can display events that match the conditions specified in this window by clicking the **Preview Data** button. When you click this button, an event table is displayed in a new browser window. This table has the same format as the event table shown in the **Service Details** portlet. For SQL data sources, you can preview the selected data in the Data Fetcher **View Data** window by clicking the **View Data** button.
Event Discriminators list	If you are using an ObjectServer data feed, you can select one or more ObjectServer classes from this list. You use event discriminators to specify which event types are relevant to the metric rule.
Instance name list.	You can select an event field from this list. TBSM uses the value of this field to determine the affected service instance when processing this rule. Alternatively, you can use the **Advanced** button as described below to specify an instance expression.
Advanced button	As an alternative to specifying an instance name field as described above, you can click this button to specify an instance expression. The instance expression allows you to specify multiple field names, to manipulate values, or both in the field using statements and functions in the TBSM expression syntax.

5. Choose from available filter fields and specify the matching conditions.

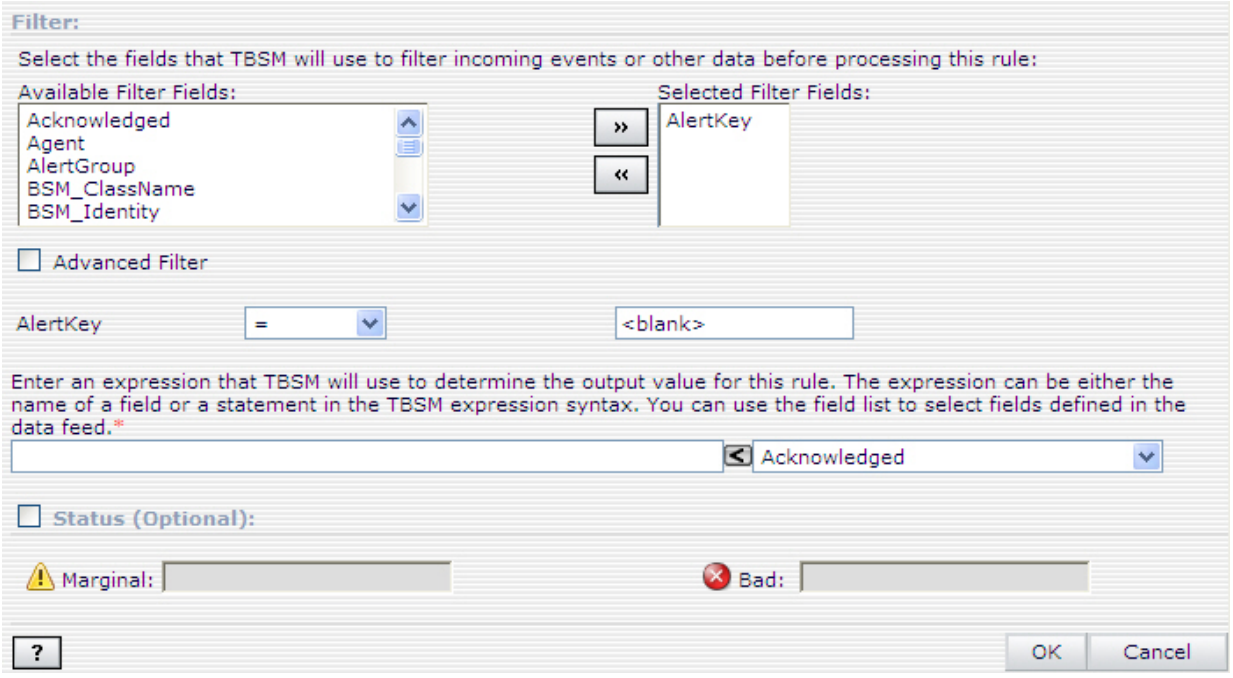


Figure 13. Edit Incoming Status Rule window: numerical rule

The **Available Filter Fields** list in this section of the window contains the fields in the incoming events (or other data provided by the data feed).

In the figure above, TBSM processes the rule for a service where the value of the `lowlevelserviceid` field equals `WebFarm` and the value of the `severity` field equals 5. When these filter conditions are met, the output value for the rule equals the value of the `opentickets` field.

You can choose any number of fields from this list. The lists and text fields below allow you to specify the condition that each field must match in order for TBSM to process this event using the rule.

6. Enter the rule output expression in the text field below. The expression can be either the name of a field or a statement in the TBSM expression syntax. You can use the field list to select fields defined in the data feed. The expression will be evaluated as a string value to assign to the rule a text output.
7. (Optional) If you want the text-based rule to also set the general state of the affected service instance, select **Status** and enter threshold values for the two service states in the **Marginal** and **Bad** fields. If the value of the expression contains the text string configured in the Bad status box then the service will be Bad. If the value of the expression contains the text string configured in the Marginal status box then the service will be `Marginal`.
8. Click **OK**.

Creating automatic roll-up rules

This topic describes how to create automatic roll-up rules.

Before you begin

You can automatically roll up rule output values to a service template. Instead of editing each level of the service model and creating an aggregation rule by hand, you can automatically roll-up the rule to a service model. Only roll up the rule automatically to a single parent template. The roll-up feature works best when there is only one parent template at each level of the service model. If the original template has multiple parent templates, you need to create a separate rule for each parent template. You then roll up each rule separately. Do not roll up a single to multiple service templates.

Note: The automatic rollup option is only available for numerical incoming status rules and numerical formula rules.

About this task

To automatically roll-up rule output values to service templates, complete the following steps:

Procedure

1. From the **Edit Template** tab in the Service Editor, click the **Rules** tab.

   If you are creating a rule for the roll-up, click **Save** on the **Edit Template** tab to save the rule. Otherwise, it will not be possible to configure the rule.
2. To edit an existing rule, click the name of the rule you want to configure for automatic roll-up from the list. The **Edit Incoming Status Rule** window opens.
3. Click **Automatic Roll-up**. The **Automatically Roll-up Numerical Rule** window opens.
4. For each template to which you want to roll-up:
   a) Select the template from the list of available templates.

      Initially, all templates except the one for which the incoming status rule is defined are in alphabetic order in the "available templates" list and no templates are in the "roll up to templates" list.

      The order of the templates is important. You should have the most immediate, or lowest level, at the bottom. The least immediate, or highest level, should be at the top.

   b) Click **Add selected template to top of Roll up to templates**.
5. To delete an aggregation rule:
   a) Select the template from the "Roll-up to templates" list.
   b) Click **Remove selected template from Roll up to templates**.

6. Choose one or more aggregation functions.
7. Any changes you make will not be carried out until you click **OK**.
8. You must save your rules in the **Edit Template** tab before they become active.
9. To verify, check the service template you selected for the rule roll-up for the new rule.

   Note: The names of the aggregation rules are automatically generated based on the name of the original rule, the names of the parent and child template at each level of the service model, and the name of the aggregation function. Do not change rule or template names. TBSM uses these names to reference the rule values. When you create numerical and text-based rule, you need to create a custom service tree or view definition to display the output of the rule in the TBSM console. The trees and view definitions use the template and rule names to map the rule values to a display object. As a result, if you change the name of a rule or a template used in a custom display, the custom tree or view definition will no longer display the rule values properly. If you want to change a rule name, change the name before you use the rule elsewhere in TBSM.

Chapter 14. Aggregation and numerical formula rules

This section explains how to work with aggregation and numerical formula rules in IBM Tivoli Business Service Manager (TBSM).

Aggregation and numerical formula rules overview

This topic describes aggregation and numerical formula rules.

Aggregation and numerical formula rules are sets of properties and conditions used by TBSM to aggregate and otherwise combine parameters related to the state of a service. These parameters consist of general, detailed, or both state information generated by incoming status rules or other aggregation and numerical formula rules.

You use aggregation and numerical formula rules as part of a logical model that takes information derived from ObjectServer events and other incoming data and uses it to determine different aspects of the status of a service. The rule definitions depend on how you want to aggregate or otherwise combine this information.

You define aggregation and numerical formula rules when you create or edit a service template. Each template can have any number of associated aggregation and numerical formula rules. You can configure TBSM to display the results of these rules in the **Service Tree** or Service Viewer. This display allows users to view and respond to changes in the general or detailed state of the service in real time.

Note: If you change rule or template names, you can cause issues with other features of TBSM. TBSM uses these names to reference the rule values for other rules, such as auto-population or ESDA rules and for custom displays such as service trees and view definitions.

If you want to change a rule or template name, change the name before you use the rule or template elsewhere in TBSM.

For example, when you create numerical and text-based rule, you need to create a custom service tree or view definition to display the output of the rule in the TBSM console. The trees and view definitions use the template and rule names to map the rule values to a display object. As a result, if you change the name of a rule or a template used in a custom display, the custom tree or view definition will no longer display the rule values properly.

Good, marginal, and bad aggregation rules overview

This topic describes good, marginal, and bad aggregation rules.

TBSM uses good, marginal, and bad aggregation rules to aggregate service states across a set of child service instances. This type of rule is defined as part of the parent service template. The general services states aggregated by rules of this type are generated by incoming status rules or other aggregation rules.

When good, marginal, and bad aggregation rules are processed, the resulting state is assigned to the parent service instance. This service state information is stored in memory and is identified by the rule name when referenced elsewhere in TBSM.

Good, marginal, and bad aggregation rules consist of a set of properties that specify the relevant child service instances and the calculations that must be performed to aggregate their service states. The rules aggregate these states by performing simple math calculations or by executing more complicated policy functions.

TBSM processes good, marginal, and bad aggregation rules during runtime when a service state associated with child service changes. When the service state changes, the application checks the parent instances of the associated service to see if related aggregation rules exist. TBSM then processes any related rules and assigns the resulting applicable service state to the parent instances.

Note: The general service states of Good, Marginal, and Bad correspond to numeric rule output values of 0, 3, and 5 respectively. This numerical value is used by TBSM when the resulting service state is combined with other rule output values using a numerical aggregation rule or numerical formula rule.

You can configure TBSM to display the results of a good, marginal, and bad aggregation rule as part of the Service Tree or Service Viewer features of the application GUI. You can also use the result as a factor of other aggregation rules.

For examples of these rules and how to map them to trees and views, see the *TBSM Scenarios Guide*.

Good, marginal, and bad aggregation rule properties

This topic describes good, marginal, and bad aggregation rules and properties. This topic also describes the aggregation rule properties: rule name, child template, condition type, and additional properties based on condition type.

Good, marginal, and bad aggregation rules have the following properties:

- Rule name
- Child template
- Conditions

Rule name

The rule name is a unique string that identifies the rule. This name is also used to identify the output value that results from processing the rule when used elsewhere in TBSM.

Note: The Name field for templates, data fetchers, rules, data sources, maintenance schedules, and other objects must not contain these special characters:

```
" & < > \ / * ? | ( ) : ;
```

Names must not contain spaces. Otherwise, these objects might not be saved properly in the TBSM database. If the object does not save, remove any special characters or spaces from the name field.

Child template

The child template is the service template of child service instances whose state you want to combine using the good, marginal, and bad aggregation rule. Select a template from the **Child Rule/Mapping** drop-down list.

Note: Child services not tagged with this template are not included in the aggregation calculation.

Description

Using a description will help you stay organized in your TBSM environment.

Display Name

The Display Name will appear in the metric list in the Time Window Analyzer. If no display name is specified, the name of the rule will be used.

Conditions

Propagation type

TBSM has two propagation types, which determine the type of child status used in the aggregation rule. A **Child instance** can be tagged with multiple templates. Each of these templates might have rules that affect the child status. When an event matches the template rule criteria, the child template status changes. When the **Child template** status changes, the overall status of the child service refreshes. The net overall status is the worse case of all the child templates. The propagation type

identifies whether the child template status or the overall status of the child service instance is propagated. The **Child template** propagation type is the default algorithm.

Condition type

TBSM supports two condition types for good, marginal, and bad aggregation rules.

The first type of condition, percentage type calculation (**% of children**), calculates the state of the parent service using the percentage of children in a particular state. For percentage type calculation (**% of children**) you specify the **Child status** (`Marginal`, Bad, or `All`) to monitor. The thresholds are based on the percentages of child services that match the specified state.

The second type of condition, **Any child**, calculates the general state of the parent service using the worst state of any child service. To use this condition, you specify a state for the parent service when the worst state of a child service instance is `Marginal` or Bad.

Under all circumstances, the state of the parent service is Good by default and only changes when the specified percentage or worst state condition is true.

Additional Properties for Condition type

These additional properties are dependent on the Condition type you selected. For **Any Child**, the additional property is Output Severity. For **% of Children**, the additional properties are Weight Property, Weight Default, Child Status, and Output Severity Threshold.

Any Child additional properties

Output Severity
You specify the child state (`Marginal` or Bad) at which the parent service is considered to be `Marginal` or Bad.

% of children additional properties

Weight Property
If a property name is specified, the child service instance value for that property is used to calculate a weighted percentage.

Weight Default
The default **Weight Default** value is 50. This default value is assigned if no Weight Property value or child service instance property value are specified.

Note:

If the child service instance property value is not numeric, the default **Weight Default** value of 50 is assigned.

Child Status
Select the child status (`Marginal`, Bad, or `All`) you want to monitor. The `All` state option specifies that all child services with `Marginal` and Bad states are considered. However, children with a `Marginal` status only affect 50% of the calculation, and children with a Bad have 100% effect on the calculation.

Output Severity Threshold
Specify the percentage threshold to change the parent service state. When the specified percentage of child services has been met, then parent state changes to Bad or `Marginal`.

Metric Collection

The Metric Collection options enable TBSM to collect and store the metric data for the rule. You can activate the storing of status changes of the rule and display the recent history of the metric. The collected metric data can be displayed in the Time Window Analyzer, and stored in the TBSM database. You can also store status changes of rules to the warehouse (TDW) using the BSM Agent. The TDW stores data for historical reporting.

Note: If you want to store status changes to the warehouse, you need to install and configure the BSM agent and TDW.

Store data for this rule for TWA
Store the output values for this rule in the TBSM database for the Time Window Analyzer.

Store data for this rule for TDW
Store the output values for this rule in the TDW as a key performance indicator for the BSM Agent Service Indicators attribute group.

Create and edit good, marginal, and bad aggregation rules

This task describes how to create and edit good, marginal, and bad aggregation rules. This topic also describes the properties of good, marginal, and bad aggregation rules.

About this task

To create or edit good, marginal, and bad aggregation rules:

Procedure

1. From the **Rules** tab of the **Edit Template** tab in the Service Editor, click the **Create Good, Marginal, and Bad Aggregation Rules** button. Or, to modify an existing rule, click the name of the rule that you want to edit.

   The **Good, Marginal, and Bad Aggregation Rule** window opens.
2. Enter the rule properties using Table 26 on page 84 as your guide.

Table 26. Good, marginal, and bad Aggregation rule: properties

Window element	Description
Rule Name field	The name of the good, marginal, and bad aggregation rule. **Note:** The Name field for templates, data fetchers, rules, data sources, maintenance schedules, and other objects must not contain these special characters: `" & < > \ / * ? | ( ) : ;` Names must not contain spaces. Otherwise, these objects might not be saved properly in the TBSM database. If the object does not save, remove any special characters or spaces from the name field.
Description	Using a description will help you stay organized in your TBSM environment.
Display Name	The Display Name will appear in the metric list in the Time Window Analyzer. If no Display Name is specified, the name of the rule will be used.
Child Rule / Mapping list	The service template of child service instances whose general state you want to aggregate using the rule.
Propagation Type selection	The type of child status you want to propagate. Options are `Child instance` or `Child template`.
Condition Type selection	The condition type associated with the rule. Options are `Percentage of Children` and `Any Child`.
Weight Property	The child service instance property whose value is used to calculate a weighted percentage. This is optional for `Percentage of Children` condition type only.
Weight Default	This is the default weighted value. This is optional for `Percentage of Children` condition type only.

Table 26. Good, marginal, and bad Aggregation rule: properties (continued)	
Window element	**Description**
Child Status list	The child status (`Bad`, `Marginal`, or `All`) to monitor which will be aggregated to determine the parent service status.`Percentage of Children` condition type only.
Output Severity Threshold selections	Thresholds for percentages of children with the specified **Child Status**. When the calculated percentages exceed the specified thresholds, TBSM sets the state of the parent service to `Bad` or `Marginal`. For `Percentage of Children` condition type only.
Output Severity lists	Matching conditions that specify the state of the parent service when the worst state of a child is `Marginal` or `Bad`. For `Any Child` condition type only.
Store data for this rule for TWA checkbox	If you select this option, TBSM will collect and store status changes for the rule. This data is stored in the TBSM database. Th collected metric data will be displayed in the Time Window Analyzer.
Store data for this rule for TDW checkbox	If you select this option, TBSM will collect and store the status changes of the rule to the warehouse (TDW) using the BSM Agent. The TDW stores data for historical reporting. **Note:** You need to install and configure the BSM agent and the TDW to use this option.

3. Click **OK**.

Numerical aggregation rules overview

This topic describes numerical aggregation rules.

TBSM uses numerical aggregation rules to combine rule output values of the same type across a set of child service instances. This type of rule is defined as part of the parent service template. The rule output values combined by a numerical aggregation rule can be generated by incoming numerical status rules or by other aggregation and numerical formula rules.

When a numerical aggregation rule is processed, the resulting aggregated rule output value is assigned to the parent service instance. This rule output value is stored in memory and is identified by the rule name when referenced elsewhere in TBSM.

Numerical aggregation rules consist of a set of properties that specify the relevant child service instances and the calculations that must be performed to combine the rule output values. The rules combine the values by performing simple math calculations or by executing more complicated custom policy functions. Like incoming numerical status rules, numerical aggregation rules can also contain a simple status condition that sets the state of the affected service to `Good`, `Marginal`, or `Bad`.

TBSM processes numerical aggregation rules during runtime when a rule output value associated with child service instance changes. When a rule output value changes, the application checks the parent instances of the associated services to see if related numerical aggregation rules exist. TBSM then processes the related rules and assigns the resulting rule output value to the parent instances.

You can configure TBSM to display the result of a numerical aggregation rule as part of the Services Tree or Service Editor features of the application GUI. You can also use the result as a factor of other aggregation and numerical formula rules.

Numerical aggregation rules properties

This topic describes the numerical aggregation rule properties: rule name, child template, child metric rule, aggregation function, and status.

Numerical aggregation rules have the following properties:

- Rule name
- Child template
- Child metric rule
- Aggregation function
- Status

Rule name

The rule name is a unique string that identifies the rule. This string is also used to identify the output value that results from processing the rule.

Note: The Name field for templates, data fetchers, rules, data sources, maintenance schedules, and other objects must not contain these special characters:

```
" & < > \ / * ? | ( ) : ;
```

Names must not contain spaces. Otherwise, these objects may not be saved properly in the TBSM database. If the object does not save, remove any special characters or spaces from the name field.

Child template

The child template is the service template of the child service that contains the output value that you want to combine using the numerical aggregation rule.

Child metric rule

The child metric rule is the name of a rule in the child template whose output value you want to aggregate using this numerical aggregation rule. If the output of the child template rule is a value of `Good`, `Bad`, or `Marginal`, TBSM maps these states to numerical values of 5, 3 and 0 respectively during aggregation.

Description

Using a description will help you stay organized in your TBSM environment.

Display Name

The Display Name will appear in the metric list in the Time Window Analyzer. If no Display Name is specified, the name of the rule will be used.

Aggregation function

The aggregation function is the mathematical calculation used by TBSM to aggregate the rule output values from across the child service instances.

Table 27 on page 87 shows the aggregation functions for numerical aggregation rules.

Table 27. Numerical aggregation rules: aggregation functions	
Function	**Description**
Average	Adds all the rule output values and then divides the result by the total number of child service instances. The multiplier expression allows you to specify the name of a custom field in the template that is used to weight values per service instance. For example, you can create a custom field in the template named `Weight` and then manually assign a value in each service instance between 1 and 5. Then, you can specify `Weight` as the multiplier expression in the rule definition. When TBSM evaluates the Average function, it then multiplies the value of each child output rule by the corresponding value of `Weight` to give weighted outputs for each service.
Maximum	Determines the greatest value among rule output values set in the child service instances.
Minimum	Determines the least value among rule output values set in the child service instances.
Percentile	Returns the rule output value associated with the child service closest to the specified percentile. For example, if there are 60 child services and the rule output values range from 1 to 60 among them, a percentile value of 75 would return the rule output value associated with the child service ranked 15 (or 25 percent) from the highest value among the children. Recommended for scenarios with a large number of child service instances.
Sum	Adds all the rule output values set in the child service instances.
Policy	You can also create custom policy functions that perform more complicated types of aggregations. TBSM provides a pre-defined function that determines the standard deviation among output values. The standard deviation represents the degree to which rule output values differ among themselves. The other functions determine the number of child service instances whose service state is currently set to Bad and the status of child services. For more information about writing custom functions in the expression language, see the *IBM Tivoli Business Service Manager: Customization Guide*.

Note: The percentile calculation for aggregation rules simply picks the value of the child that falls closest to the percentile specified in the rule. It does not perform any averaging or smoothing calculations.

Status

You can optionally specify a status to be used by the rule to set the state of the affected service instance to `Marginal` or Bad when the value of the resulting aggregated output value exceeds the thresholds you specify.

Metric Collection

The Metric Collection options enable TBSM to collect and store the metric data for the rule. You can activate the storing of status changes of the rule and display the recent history of the metric. The collected metric data will be displayed in the Time Window Analyzer, and stored in the TBSM database. You can also store status changes of rules to the warehouse (TDW) using the BSM Agent. The TDW stores data for historical reporting.

Store data for this rule for TWA
Store the output values for this rule in the TBSM database for the Time Window Analyzer.

Store data for this rule for TDW
Store the output values for this rule in the TDW as a key performance indicator for the BSM Agent Service Indicators attribute group.

Note: If you want to store status changes to the warehouse, you need to install and configure the BSM agent and TDW.

Create and edit numerical aggregation rules

This topic describes how to create and edit numerical aggregation rules. This topic also describes numerical aggregation rule properties.

About this task

To create and edit numerical aggregation rules:

Procedure

1. From the **Edit Template** tab in the Service Editor, click the **Create Numerical Aggregation Rule** button. Or, to modify an existing rule, click the name of the rule that you want to edit.

   The **Edit Numerical Aggregation Rule** window opens.
2. Enter the rule properties using Table 28 on page 88 as your guide.

Table 28. Numerical aggregation rule: properties

Window element	Description
Rule Name field	The name of the numerical aggregation rule.
Child Template list	The service template of child instances that contain the rule output value you want to aggregate.
Child Metric Rule	The name of a numerical rule in the child template. This rule generates the output value that you want to aggregate.
Description	Using a description will help you stay organized in your TBSM environment.
Display Name	The Display Name will appear in the metric list in the Time Window Analyzer. If no Display Name is specified, the name of the rule will be used.
Aggregation Function	The mathematical calculation used by TBSM to aggregate the output values across child service instances. You can choose `Average`, `Maximum`, `Minimum`, `Percentile`, `Sum`, and `Policy`. If you choose `Policy`, you must specify a custom policy that performs the aggregation.
Edit Policy button	Opens the TBSM policy editor. You can use the policy editor to modify one of the predefined functions or to create your own functions using the TBSM expression language. For more information about creating and modifying these functions, see the *IBM Tivoli Business Service Manager: Customization Guide*.
Status (Optional)	If you want this rule to set the general state of the affected service instance, select **Status** and enter threshold values for the two service states in the `Marginal` and Bad fields.
Store data for this rule for TWA checkbox	If you select this option, TBSM will collect and store status changes for the rule. This data is stored in the TBSM database. The collected metric data can be displayed in the Time Window Analyzer.
Store data for this rule for TDW checkbox	If you select this option, TBSM will collect and store the status changes of the rule to the warehouse (TDW) using the BSM Agent. The TDW stores data for historical reporting. **Note:** You need to install and configure the BSM agent and the TDW to use this option.

3. Click **OK**.

Numerical formula rules overview

This topic describes numerical formula rules.

TBSM uses numerical formula rules to combine rule output values of different types within the same service instance. These output values can be generated by incoming numerical status rules or by numerical aggregation rules. When a numerical formula rule is processed, the resulting combined output value is assigned to the same service instances where the original output values existed. This output value is stored in memory and is identified by the rule name when referenced elsewhere in TBSM.

Numerical formula rules consist of a set of properties that specify which output values are to be combined and the calculations that must be performed to combine them. These calculations can be simple math operations or more complicated policy functions. Like incoming numerical status rules and numerical aggregation rules, numerical formula rules can also contain a simple status condition that sets the state of the affected service to Good, `Marginal`, or Bad.

As with numerical aggregation rules, TBSM processes numerical formula rules during runtime when a rule output value associated with service instance changes. When the output value changes, the application checks the associated service instance to see if related numerical formula rule exists. TBSM then processes the related rules and assigns the resulting rule output value to the service instance.

Numerical formula rule properties

This topic describes these numerical formula properties: rule name, combination expression, and status.

This section contains information about creating and editing numerical formula rules.

Numerical formula rules have the following properties:

- Rule name
- Combination expression
- Status

Rule name

The rule name is a unique string that identifies the rule. This name is also used to identify the output value that results from processing the rule.

Note: The Name field for templates, data fetchers, rules, data sources, maintenance schedules, and other objects must not contain these special characters:

```
" & < > \ / * ? | ( ) : ;
```

Names must not contain spaces. Otherwise, these objects may not be saved properly in the TBSM database. If the object does not save, remove any special characters or spaces from the name field.

Description

Using a description will help you stay organized in your TBSM environment.

Display Name

The Display Name will appear in the metric list in the Time Window Analyzer. If no Display Name is specified, the name of the rule will be used.

Expression

The expression (e.g. Attribute1.Value + Attribute2.Value) specifies how TBSM combines the rule output values in the service instance. When you write the expression, you use the notation *rule_name*`.Value` to represent the output values that you want to combine, where *rule_name* is the name of the rule that generated the output value.

As an alternative to using a combination expression, you can create a custom policy function that TBSM will use to combine the wanted rule output values. Using a custom policy function allows you to take advantage of conditional statements like `If` and `When` that are not otherwise available in combination expressions.

If the output of a rule in the service is a value of `Good`, `Marginal`, or `Bad`, TBSM maps these states to numerical values of 0, 3 and 5respectively during combination.

For more information about the expression language, see the *IBM Tivoli Business Service Manager: Customization Guide*.

Status

Optionally, you can specify a status to be used by the rule to set the state of the affected service instance to `Marginal` or `Bad` when the resulting output value exceeds thresholds that you specify.

Metric Collection

The Metric Collection options enable TBSM to collect and store the metric data for the rule. You can activate the storing of status changes of the rule and display the recent history of the metric. The collected metric data will be displayed in the Time Window Analyzer, and stored in the TBSM database. You can also store status changes of rules to the warehouse (TDW) using the BSM Agent. The TDW stores data for historical reporting.

Store data for this rule for TWA
Store the output values for this rule in the TBSM database for the Time Window Analyzer.

Store data for this rule for TDW
Store the output values for this rule in the TDW as a key performance indicator for the BSM Agent Service Indicators attribute group.

Note: If you want to store status changes to the warehouse, you need to install and configure the BSM agent and TDW.

Automatic roll-up of incoming status rules

In TBSM, you can define the aggregation rules automatically directly from the incoming status rule. Internally, the same aggregation rules will be created. Externally, the aggregation rules can even be directly edited on service templates after they are created automatically. TBSM automatically names aggregation status rules based on the name of the incoming status rule and template that the aggregation belongs to. However, you will not be allowed to add a template to the list if the name of the automatically generated aggregation rule would be too long.

The rules can also be managed directly by editing each service template to which they are being rolled up. If you change an automatically generated aggregation rule yourself, the "automatic roll-up" function may no longer display the service templates. If the names of the incoming status rules or service templates change, the automatically created aggregation rules will still work but are not displayed if you press the **Automatic Roll-up** button for this incoming status rule.

To have more control over rolling up numerical status rules, you might need to create the aggregation rules yourself.

Create and edit numerical formula rules

This topic describes how to create and edit numerical formula rules.

About this task

To create a numerical formula rule:

Procedure

1. From the **Edit Template** tab in the Service Editor, click the **Create Numerical Formula Rule** button.

The **Edit Numerical Formula Rule** window opens.

To edit an existing rule, click the name of the numerical formula rule you want to edit.

2. Enter the rule properties using Table 29 on page 91 as your guide.

Table 29. Numerical formula rule properties

Window element	Description
Rule Name field	The name of the numerical formula rule. The Name field must not contain these special characters: `" & < > \ / * ? | ( ) : ;`
Description	Using a description will help you stay organized in your TBSM environment.
Display Name	The Display Name will appear in the metric list in the Time Window Analyzer. If no Display Name is specified, the name of the rule will be used.
Expression list	You can use the expression list to view and add the names of rule-output values to the combination expression.
Expression field	You can use this field to enter the expression to be used by TBSM to combine the values of output values associated with the service instance, for example, `NumericInternalRule1.value + NumericInternalRule2.value`.
Edit Policy button	Opens the TBSM policy editor. You can use the policy editor to modify one of the predefined functions or to create your own functions using the TBSM expression language. For more information creating and modifying these functions, see the *IBM Tivoli Business Service Manager: Customization Guide*.
Use Policy	If you created a policy for this rule, select this check box to enable the policy. To disable the policy, unselect this check box.
Text Rule	If you select the **Text Rule** check box, the output value of the rule will be text.
Status (Optional)	If you want this rule to set the general state of the affected service instance, select **Status** and enter threshold values for the two service states in the `Marginal` and Bad fields. For example, you can specify that if the result of the rule is `100` or higher, the service's status changes to **Bad**.
Store data for this rule for TWA checkbox	If you select this option, TBSM will collect and store status changes for the rule. This data is stored in the TBSM database. The collected metric data will be displayed in the Time Window Analyzer.
Store data for this rule for TDW checkbox	If you select this option, TBSM will collect and store the status changes of the rule to the warehouse (TDW) using the BSM Agent. The TDW stores data for historical reporting. **Note:** You need to install and configure the BSM agent and the TDW to use this option.
Automatic Roll-up button	If you use the Auto-Rule roll-up feature, you can automatically create rules in other service templates that contain the values from the incoming status rule. That is, you pass the rule output values from the incoming status rule to a rule in another service template.

3. Click **OK**.

Chapter 15. Auto-population rules

This section describes how you can set up IBM Tivoli Business Service Manager (TBSM) to automatically create services with the Auto-Population Rule feature.

Automatic service creation overview

This topic describes what to do before you create services automatically.

You can configure TBSM to automatically create services based on event data from the Netcool/OMNIbus ObjectServer or an SQL data source.

Before you set up auto-population rules, you must define your service template hierarchy in TBSM. See *TBSM Scenarios* guide for more information. You must also determine the information that follows:

1. The template and status rules for the services you want to create automatically.
2. The parent template and service that you want the created service to support. If a parent service name is identified in the data source or ObjectServer event, you can also assign or create parent services based on the data. You can create or assign the parent services for each level in the template dependency model.
3. For each higher level in the service hierarchy, you need to determine the parent service (if any) for this level.

   Figure 14 on page 93 shows a template structure that will be used in this chapter to create services automatically. See the *Service Template Structure* in the *IBM Tivoli Business Service Manager: Scenarios Guide* for more information about configuring this template structure.

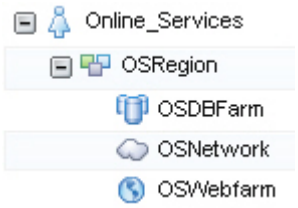


Figure 14. Auto-Population sample template service hierarchy

TBSM auto-population solutions overview

This topic describes auto-population solutions available in TBSM.

TBSM provides two solutions for automatically configuring services based on template rules:

- Using an incoming status rule in combination with auto-population rules.

  Auto-population rules are appropriate for simple models that do not constantly update. They work best when you can obtain all the data (ObjectServer or SQL data source) for the model from a single row in one table.
- ESDA auto-population works better for more complex models that update data in real time, when the model structure changes quickly whenever underlying data changes.

This section describes the first solution, using an incoming status rule in combination with auto-population rules.

Adding auto-population rules

This topic describes auto-population rules. This topic also describes how to set status auto-population rules.

About this task

This section describes how to set the status auto-population settings for a service template. The status auto-population rules control how services are generated for templates with status rules at the bottom tier of your service hierarchy.

For the template example in Figure 14 on page 93, services tagged with the `OSDBFarm, OSNetwork`, `OSWebfarm` templates at the bottom tier of the service hierarchy respond to events according to the incoming status rules in the template.

Note: The information provided in this section is based on examples in the *IBM Tivoli Business Service Manager: Scenarios Guide*. Refer to that document for clarification of specific steps described here.

Setting status auto-population rules

About this task

In the following examples, rules are created that automatically populate all service instances in the sample service model.

To set up auto-population rules for templates with status rules, complete the following steps:

Procedure

1. Select **Templates** from the drop-down list in the **Service Navigation** portlet. Select the template that you want to create services for using the Auto-Population feature.

   Using the example in the example hierarchy in Figure 14 on page 93, select the `OS_Network` template.

   When the *Template* editor opens in the Service Editor, the **Rules** tab is open by default.

   The examples that follow create auto-population rules for the `OSNetwork` template: one for the `HighNetworkTickets` incoming status rule and one for the `CriticalNetworkTickets` incoming status rule. You can find more information about the incoming status rules shown here in the *IBM Tivoli Business Service Manager: Scenarios Guide*.
2. Click **Create AutoPopulation Rule** to create a rule or click the name of an existing auto-population rule to edit it.

   The system displays the **New Auto-Population Rule** window. For the examples in this section, the latter was done.

   This window shows the service templates related to the selected template in the column on the left side of the window.

   Some auto-population rules have been created with the service hierarchy described in Figure 14 on page 93.

   The `OSNetwork` template is at the bottom tier of the service hierarchy. Since `OSNetwork` is the template selected, `OSNetwork` is the current template. The selected button is colored gray.

   The other template buttons in the hierarchy are colored dark blue. When you create a rule, the dark blue color does not show until you have created expressions which affect the template on that tab. For example, when you create a rule, when the window first opens, none of the tabs on the left have a dark blue background and you can only select the lowest level tab. When you enter rule expressions for the lowest level template, the background of the next higher one turns dark blue and you can select it. In

turn, when you enter rule expressions for this template, the background of the next higher one turns dark blue and you can select it and enter rule expressions on that tab.

Click the `Online Services` template, the highest level service template in the hierarchy.

The `Online Services` template is now colored gray and shows the auto-population rule that was created for it. The other two service template names in the hierarchy are colored blue.

When you are creating an auto-population rule, the system forces you to start from the lowest level template in the hierarchy. You cannot start with the parent template or skip a level in the hierarchy.

Note: It is not necessary to configure the parent level auto-population rule. It turns blue so that you can configure it, but it is not necessary.

3. Click the lowest level template in the hierarchy on the left side of the window (`OSNetwork`, for example.)
4. Table 30 on page 95 explains the field entries for this example.

Table 30. New Auto-Population Rule window

Field name	Description
Rule Name	Enter a name for the rule or accept the default. For this example, enter `NetAutopopHigh`. **Note:** The Name field for rules, data sources, maintenance schedules, view definitions and other objects must not contain these special characters: `" & < > \ / * ? | ( ) : ; $ ! %` Names must not contain spaces. Otherwise, these objects may not be saved properly in the TBSM database. If the object does not save, remove any special characters or spaces from the name field.
Incoming Status Rule	Since this template is the lowest in the hierarchy, you need to select the incoming status rule that triggers the auto-population rule. Select a rule to auto-create service instances from an event in the ObjectServer or a database. In this example, `HighNetworkTickets` is from a data fetcher called `REGIONALTICKETS`.
If an event matches the above status rule, but no service instance exists, create a instance using the following information: **Note:** When entering name expressions, select the field values from the drop-down menu whenever possible to avoid incorrectly typing the field names for the name expressions. Selecting the values eliminates spelling and case errors that will cause your rule to malfunction.	

Table 30. New Auto-Population Rule window (continued)	
Field name	**Description**

Instance Name Expression*	Enter an expression that will be used to construct the instance name of the new service that will be created. This expression should be written in the Netcool/Impact policy language using fields in the event. You can use the dropdown list on the right to help you construct the expression. It provides a selectable list of event fields that you can move into the expression box.
	An example expression is:
	`LOWLEVELSERVICEID+CUSTOMER+REGION`
	Important: The result of the naming expression field for automatically created services must not contain these special characters:
	`" < > \ * ? | ;`
	The result of the naming expression must not be longer than 127 characters. Names longer than 127 characters can cause display and performance issues.
	For more information about expressions, see the *IBM Tivoli Business Service Manager: Customization Guide*.

Display Name Expression	If you want a different display name other than the default name defined in the **Instance Name Expression** field, enter an expression in this optional field. This expression must be in the TBSM expression language. If no display name is configured, the display name will be the same as the instance name (which is configured in the **Instance Name Expression** field above. An example expression is: `LOWLEVELSERVICEID+'_for_'+(RExtract(CUSTOMER,'(...).*'))+REGION` which would build the display name value from the `LOWLEVELSERVICEID` plus the string `'_for_'` plus the first three characters of the CUSTOMER field value, and the REGION. An example resulting name is: `Network_for_ABCAsia`. See the *IBM Tivoli Business Service Manager: Scenarios Guide* for more examples and information.
Service Level Agreement	If the service template has multiple SLAs, select the SLA you want from this drop-down list. In the example, the default SLA level `Standard` is selected.
Restriction Filter	Enter an expression in this optional field if you want to apply additional conditions when a service instance is auto-created. For example, if you only want a service created when the ALERTKEY field has a value, type ALERTKEY<> `''` (ALERTKEY field value not equal to empty string) in this field. Otherwise, you can use the default value of `true`. A value of `true` means always create the instance without any additional conditions.
Custom Configuration	See “Custom auto-population options” on page 102.
Enabled	Select this check box to activate the rule.

5. Click **OK**.
6. Click the **Save** button in the *Edit Template* toolbar.

The system saves the auto-population settings.

7. Enter the parent service auto-population settings as described in "Example: adding settings for parent templates" on page 97 if there is a parent service for this template. Otherwise, close the **Auto-Population Rules** window.

Example: adding settings for parent templates

This task is an example which describes adding setting for parent templates.

About this task

In this example, you set rules for the `OSRegion` template, the next template in the hierarchy. Since the data for the parent `OSRegion` template is in the row retrieved by the `HighDBTickets` incoming status rule, you can automatically create the service names for the `OSRegion` services.

Procedure

1. Enter the values for the `OSRegion` service template using Table 31 on page 97 as your guide.

Table 31. Auto-Population settings for a parent template

Window element	Value
Rule Name	Enter a name for the rule or accept the default. For this example, enter `DBAutpopHigh`.
Template Name	Select the parent template, in this case `OSRegion`.
Instance Name Expression	Enter the fields that uniquely identify the service for a given rule. TBSM evaluates this expression against the incoming event to obtain the instance name. For this example, enter: `CUSTOMER+REGION`
Display Name Expression	Leave blank. Leaving this field blank generates the service names using the value in the **Instance Name Expression** field.
Service Level Agreement	If the service template has multiple SLAs, select the SLA you want from this drop-down list. In the example, the default SLA level `Standard` is selected.
Restriction Filter	Enter an expression in this optional field if you want to apply additional conditions when a service instance is auto-created. For example, if you only want a service created when the ALERTKEY field has a value, type ALERTKEY<> `''` (ALERTKEY field value not equal to empty string) in this field. For this example, accept the default value of `true`. A value of `true` means always create the instance without any additional conditions.

2. Now enter the rules for the highest template in the hierarchy, `Online Services`. Since the data for the parent `Online Services` template is in the row retrieved by the `HighNetworkTickets` rule, you can automatically create the service names for the services assigned to the `Online Services` template.
3. Select the `Online Services` template.
4. Enter the values for the `Online Services` template using Table 32 on page 98 as your guide.

Table 32. OSNetwork AutopopHigh rule settings for Online Services	
Window element	**Value**
Rule Name	Enter a name for the rule or accept the default.
Template Name	Select the parent template.
Instance Name Expression	Enter the fields that uniquely identify the service for a given rule. TBSM evaluates this expression against the incoming event to obtain the instance name.
Display Name Expression	If you want a different display name other than the default name defined in the **Instance Name Expression** field, enter an expression in this optional field. This expression must be in the TBSM expression language. Otherwise, the name will be the same as the service name defined in the service template status rule.
Service Level Agreement	Accept the default value of `Standard`.
Restriction Filter	Accept the default value of `true`. A value of `true` means always create the instance without any additional conditions.

5. Click **OK**.
6. Click **Save** in the *Edit Template* toolbar to save the new rules.

   Table 33 on page 98 describes how to set up the auto-population rules for an incoming status rule named `CriticalNetworkTickets`.
7. Click **Create Auto-Population Rule**.

   The *Auto-Population Rule* window opens.
8. Create the rule for `OSNetwork`, using the information in Table 33 on page 98 as your guide.

Table 33. OSNetwork template auto-population rule values	
Window element	**Value**
Rule Name	`NetAutopopCrit`
Incoming Status Rule	Since this template is the lowest in the hierarchy, you need to select the incoming status rule that creates the services. Select `CriticalNetworkTickets`
Instance Name Expression	`LOWLEVELSERVICEID+CUSTOMER+REGION`
Display Name Expression	`(RExtract (LOWLEVELSERVICEID, '( . . . ) . *')) + (Rextract (CUSTOMER, ' ( . . . ) . *)) +REGION`
Service Level Agreement	If the service template has multiple SLAs, select the SLA you want from this drop-down list. In the example, the default SLA level `Standard` is selected.
Restriction Filter	Enter an expression in this optional field if you want to apply additional conditions when a service instance is auto-created. For example, if you only want a service created when the `ALERTKEY` field has a value, type `ALERTKEY<> ''` (`ALERTKEY` field value not equal to empty string) in this field. For this example, accept the default value of `true`. A value of `true` means always create the instance without any additional conditions.

9. Click the OSRegion tab and enter the values shown in Table 34 on page 99.

Table 34. OSNetwork NetAutopopCrit rule settings for OSRegion

Window element	Value
Rule Name	`NetAutopopCrit`
Incoming Status Rule	Select `OSRegion`
Instance Name Expression	`CUSTOMER+REGION`
Display Name Expression	Leave this field blank to generate the service names using the entry in the **Instance Name Expression** field.
Service Level Agreement	`Standard`
Restriction Filter	`true` A value of `true` means always create the instance without any additional conditions.

10. Click the **Online Services** tab and enter the values shown in Table 35 on page 99.

Table 35. OSNetworkNetAutopopCrit rule settings for OSRegion

Window element	Value
Rule Name	`NetAutopopCrit`
Template Name	Select `Online Services`
Instance Name Expression	`CUSTOMER+HIGHLEVELSERVICEID`
Display Name Expression	`CUSTOMER+(Rextract (HIGHLEVELSERVICEID, ' . . . . . (.*) ) )` This expression extracts all the characters after the eighth character in the field value.
Service Level Agreement	`Standard`
Restriction Filter	`true` A value of `true` means always create the instance without any additional conditions.

11. Click **OK**.
12. Click the **Save** button in the *Edit Template* toolbar.

Deleting an auto-population rule set

This task describes how to delete an auto-population rule set.

About this task

To delete an auto-population rule, complete the following steps:

Procedure

1. In the Service Viewer **Edit Template** tab, select the **Rules** tab.
2. Select the check box(es) to the left of the auto-population rules that you want to delete.
3. Click the **Delete Selected** button.
4. Click **Save** in the toolbar.

Working with additional service properties

About this task

When services are created with auto-population rules, the services inherit the standard rules and attributes from their assigned service template. You can also add additional service properties to the auto-populated service by entering property-name and expression-value pairs in the **Custom Auto-Population Rule Configuration** window. After the new services are created with the auto-population rule, you can see the additional properties in the **Additional** tab of the Service Editor: **Edit Service** tab.

To add an additional service property:

Procedure

1. In the **Add property name/expression pairs to set properties on the instances** table, click the **New** button. A new row for the table opens.
2. Enter the **Property Name** and **Expression** for the additional service property.

Table 36. Additional service property fields

Field	Description
Property Name	The descriptive name of the additional attribute you want to add to the service.
Expression	The expression used to obtain the value for the property. This expression can be the name of a field or a statement that uses the TBSM expression syntax to derive the property value from one or more fields from the data source.

In Figure 15 on page 100, the **Property Name** is `Regional Group`, and the **Expression** specifies the value as the value of the `REGION` field from the source data feed.

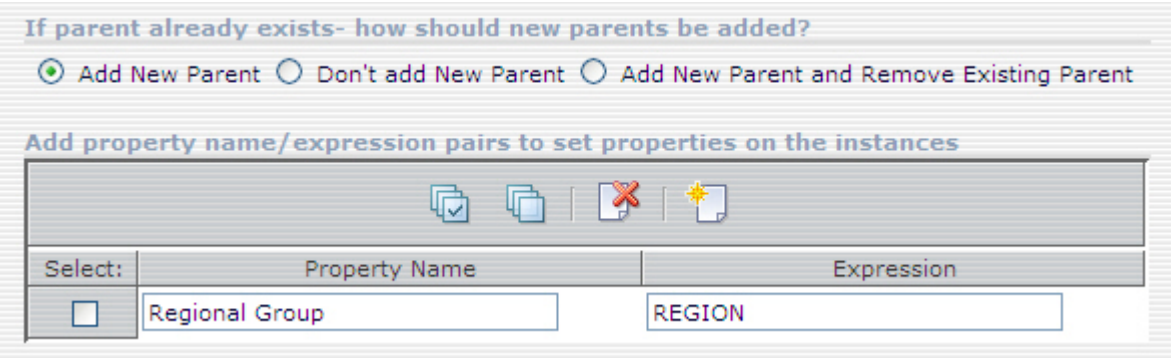


Figure 15. Custom auto-population configuration: Add additional properties

Results

For more information about expressions, see the *IBM Tivoli Business Service Manager: Customization Guide*.

Deleting an additional service property

About this task

To delete an additional service property, complete the following steps:

Procedure

1. Click the **Select** check box for the properties you want to delete.
2. Click **Delete** to delete the selected property or properties.

Specifying key performance indicators for services

This section describes how you can specify key performance indicators for the TBSM Service Indicators attribute group in the IBM Tivoli Business Service Management Agent.

The TBSM Service Indicators attribute group is designed to record the values of one or more key process indicators in the Tivoli Data Warehouse. The rules or attributes that are considered key process indicators vary between installations and even between service templates or instances. Therefore, it is necessary to specify whether a given TBSM rule or attribute values should be treated as a service indicator and recorded in the TBSM Service Indicators group.

In order for the system to treat a rule value or attribute as a key process indicator, you must define an additional attribute for the rules you want to use as key process indicators in TBSM. This attribute name must match the rule name and have a suffix of `_KPI`. This `_KPI` attribute must have the value set to true for the system to send the rule value to the service indicators attribute group.

For example, if you have a rule called Mystatus in a service template and you want to use the output value of that rule as a performance indicator, you must create an additional attribute called `Mystatus_KPI` for the service template.

Example KPI configuration

This section describes a simple configuration for a KPI.

Before you begin
You must install the Common Agent for TBSM on your IBM Tivoli Monitoring host before you set up performance indicators in TBSM.

About this task
In this example, we use one of the service templates and a formula rule from the service configuration in the TBSM *Scenarios Guide*.

Procedure

1. Open the **Edit Template** tab for the service template called **OS_Region**.
2. Double-click the rule name **RegSumCritical**.

   This rule calculates the total number of critical trouble tickets for a given region. The output value of this rule will be the key performance indicator in this example. This rule sums the trouble tickets for all the service types that affect a given region using the expression:

   ```
   NetSumCrit.Value + WebSumCrit.Value + DBSumCrit.Val
   ```

3. To create an additional attribute to store this rule value as a key performance indicator, click the **Additional** tab.
4. Add a new attribute called `RegSumCritical_KPI` and set the default value to `true`.

   Now, whenever the `RegSumCritical` rule is evaluated for a service assigned to the OS_Region template, its value is stored in the Tivoli Data Warehouse in the service indicator attribute group.

   This example shows how to control storage of service indicator values on a template basis. You can also control it on a service instance basis by adjusting the `RegSumCritical_KPI` property value in the **Additional** tab for a given service instance.

For more information about setting additional attributes, see the *IBM Tivoli Business Service Manager: Service Configuration* and *Customization* guides.

Service persistence settings

This topic describes service persistence settings.

For each service template, you can specify whether services discovered using auto-population remain in the TBSM database based on the data retrieved in the SQL query. For example, you can specify that a service will be deleted if TBSM does not detect any activity for the service within a given time period.

Important: The search function in the Service Navigation portlet and Service Tree will only find services that are persisted. Services that are not persisted will not return as search results.

Custom auto-population options

This topic describes the custom auto-population options.

In addition to the regular settings on the **Auto-Population Rule** window, you can configure custom options by opening the **Custom Auto-Population Configuration** window.

To open the **Custom Auto-Population Configuration** window, click the **Custom Configuration** link on the **Auto-Population Rule** window.

Settings for existing parent services

This topic describes the settings for existing parent services.

When your auto-population rule discovers a new parent service for a child service with an existing parent, you can specify whether you want to add the new parent. The set of fields at the top of the **Custom Auto-Population Configuration** window lets you set relationships when a new parent service is detected for a previously auto-populated child service.

Select the radio button for one of the options described in Table 37 on page 102.

Table 37. Radio button options

Option	Description
Add New Parent	This option adds the new parent service and keeps the existing parent service. This setting is the default.
Don't add New Parent	This option does not add the new parent service, but keeps the existing parent service.
Add New Parent and Remove Existing Parent	This option adds the new parent service and removes the existing parent service. It deletes all parent relationships that were created by auto-population. If the service has parents that were manually configured, those relationships will not be removed.

Settings for multiple service templates

This topic describes the settings for multiple service templates.

When your auto-population rule discovers a new service, the service is assigned to the service template specified in the auto-population rule. You can also assign the discovered service to additional service templates in the **Choose other templates to add to the auto-populated instance besides** *templatename* section of the **Custom Auto-Population Configuration** window.

The default service template is listed in the Selected Templates list. To choose other templates, select the service template you want from the Available Templates list and click the **>>** button.

Assign a group name to services

This topic describes how to assign a group name to services.

When your auto-population rule discovers a new service, the service is not assigned a group name. You can assign a group name for a service that allows users who are assigned to the group to view and edit the service instance. When you configure the groupname expression, TBSM assigns an additional property called `GroupForInstancePermissions` to the service instance and sets the value to the name of the group. When the permissions are determined for the service instance, if that additional property is set, TBSM uses that property in addition to any permissions stored in the Dashboard Application Services Hub.

If you want to create a group name for an auto-populated service, enter an expression in the **Group Name Expression** field. This expression can be the name of a field or a statement that uses the TBSM expression syntax to derive the group-name value from one or more fields in the data source.

For more information about expressions, see the *IBM Tivoli Business Service Manager: Customization Guide*.

Create policy for additional service properties

This topic describes how to start creating policies.

You can also use a custom policy to create additional properties for a given service. This feature is intended for experienced Netcool/Impact users. You should not attempt to edit this policy if you are unfamiliar with Netcool/Impact policies.

To use a custom policy to create custom service properties for the auto-populated services, select the **Use Policy For Service Properties** check box. Click the **Edit Policy** button to open the policy editor for this policy. Follow the instructions inside the policy editor.

For more information about editing this policy, see the *IBM Tivoli Business Service Manager: Customization Guide*.

Create policy for service names

This task explains how to use the generic policy and create a custom policy.

About this task

You can also create a custom policy to create the names for the auto-populated services. This feature is intended for experienced Netcool/Impact users. You should not attempt to use these policies if you are unfamiliar with Netcool/Impact policies.

You can use the generic policy that takes an SQL query as input or create a totally new policy to obtain the service names you want to create.

Using the generic policy

About this task

For the generic policy, you select a data source and create an SQL query that selects one row for each service you want to create. The **Instance Name Expression** in the **Auto-Population Rule** window is applied to each row selected by the query you enter here. To use the generic policy to obtain the service instance names, complete the following steps:

Procedure

1. Under the Use a policy to obtain instance names, select the **Use Policy** check box.
2. Select the **Use Generic Policy** radio button.

   The **Data source** and **Query** fields open.
3. Select the **Data source** you want to use for the generic policy query.
4. Enter an SQL query that returns exactly one row for each service instance you want to discover.

Creating a custom policy

About this task

This feature is intended for experienced Netcool/Impact users. You should not attempt to use these policies if you are unfamiliar with Netcool/Impact policies. To create a custom policy to obtain service instance names, complete the following steps:

Procedure

1. Under the Use a policy to obtain the instance names, select the **Use Policy** check box.
2. Select the **Use Custom Policy** radio button.
3. To open the policy editor, click on the **Edit Policy** button.
4. Follow the instructions inside the policy editor.

   For more information about editing this policy, see the *IBM Tivoli Business Service Manager: Customization Guide*.

Chapter 16. Query Builder

This section covers how to use Query Builder in IBM Tivoli Business Service Manager (TBSM).

Query Builder overview

This topic describes Query Builder.

The **Query Builder** helps you build SQL queries that retrieve data from databases supported by TBSM. The **Query Builder** can help you create queries when configuring a data fetcher and during template creation when you are adding ESDA rules.

From the Data Fetcher, the ESDA rule, and the Charts in Service Navigation creation windows, you have two options: you can enter queries manually, or use the **Query Builder** windows. However, unless you are familiar with the databases you will be working with, using the Query Builder is much easier than entering the statement manually. The **Query Builder** gives you access to the names of all the tables in the database and the fields in those tables that you can select to include in the query. It also helps you do an SQL `JOIN` between tables.

Building the query

This topic describes how to build queries using Query Builder.

About this task

You use the Query Builder to select a data source and then to help you build the query. You can access the Query Builder from either the Data Fetcher editor (see Chapter 11, “Data fetchers,” on page 39), the Templates editor, Create New ESDA Rule window (see Chapter 17, “ESDA rules,” on page 111), or from Charts in the Service Navigation drop-down list. The procedures in this chapter explain how the feature works.

Procedure

1. Click **Query Builder**.

   The first pane of the **Query Builder** wizard, *Choose Data Source and Tables*, opens.
2. Select the data source you want from list of existing data sources in the **Choose an existing Data Source** drop-down menu.

   You must create the data source before using Query Builder. For instructions on creating a data source, see Chapter 10, “Data sources ,” on page 35. When the data source has been created, reopen Query Builder and select the data source you created.
3. Select the data source and the **Available Tables** list will then automatically update with a list of the tables from the chosen data source.

   The **Available Tables** pane contains a list of all the tables in the data source.

   Note: The list of tables includes all user defined tables and all views for the database specified by the datasource. For some database types, this can mean numerous system views are included in the list. This happens because the system views are not distinguished from user defined views by the JDBC drivers or database management systems that provide the information.
4. Select one or more tables from the data source, depending on what data you need for your service model or template and click the **Add >** button.

   The tables move to the **Selected Tables** pane.

You can select more than one table at a time by holding down the Shift or Ctrl key and highlighting the table names.

To deselect a table, highlight it and click the **< Remove** button to move it back to the **Available Tables** pane.

In this example, two tables are selected, `ESDA_APPLICATION` and `ESDA_BUSINESSUNITS`.

Note: For an ObjectServer data source, you can only select one table because the ObjectServer does not support the `Join` operation.

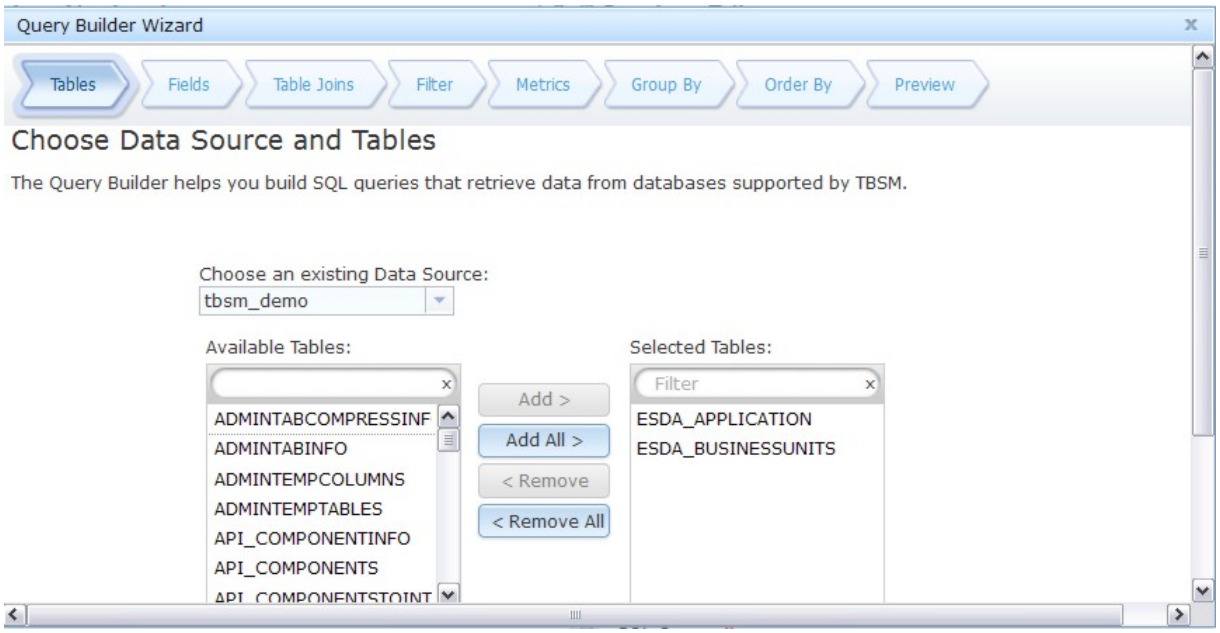


Figure 16. Query Builder - Select Tables from the Data Source window

5. Click **Next** to open the next pane in the wizard, **Choose Fields from the Selected Tables**.

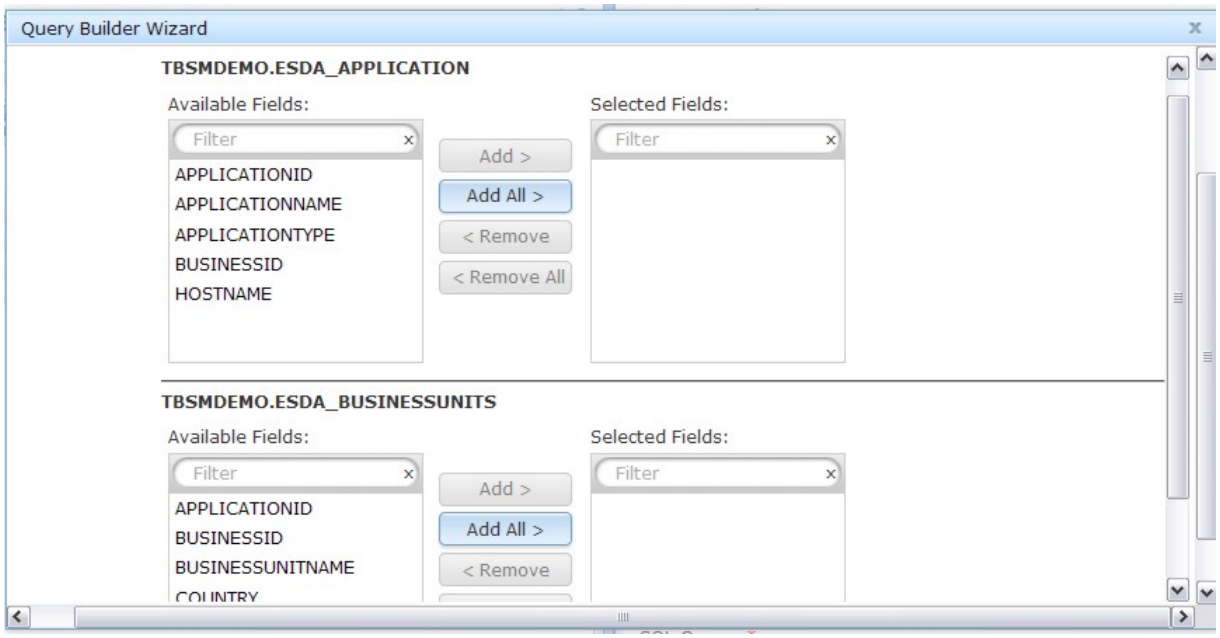


Figure 17. Choose Fields from the Selected Tables window showing two tables

The **Choose Fields from the Selected Tables** window contains two panes for each table that was selected in the **Select Tables from Data Source** window. In each pane on the left is a list of the fields that are available in those tables. The fields you select to use in the query move to the pane on the right.

6. Select the fields that you need from the tables you selected and click **>>**. You can select multiple fields at one time by holding down the Shift or Ctrl key while you select them.

   Select only the fields that you are interested in using for your service model or template.

   If you do not select any fields from these tables, the Query Builder returns all the fields from each table.
7. Click **Next** to open the next pane of the wizard, **Create Joins between Tables** window. This window displays only when you select more than on table.

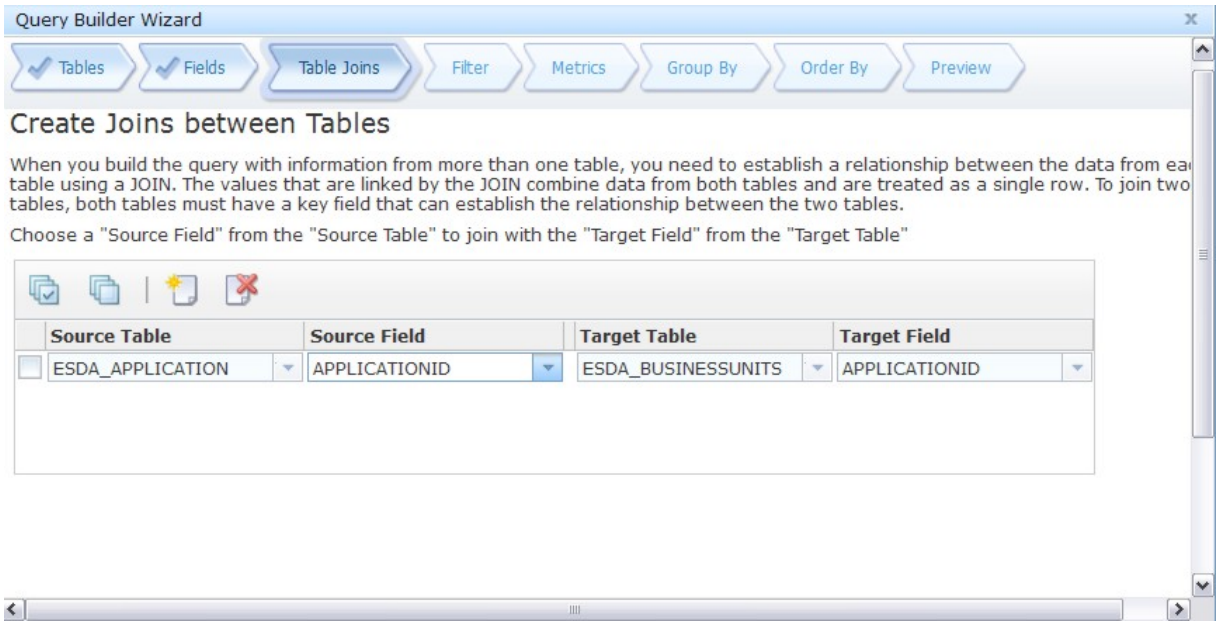


Figure 18. Create Joins between Tables window

   When you build the query with information from more than one table, you need to establish a relationship between the data from each table using a `Join`. The values that are linked by the `Join` combine data from both tables and are treated as a single row. To join two tables, both tables must have a key field that can establish the relationship between the two tables.

   In the example Figure 18 on page 107, the two tables are joined by `APPLICATIONID`, a field that exists in both tables.
8. Select a field from the source table.
9. Select a field from the target table.

   Note: If you create a complex query and the Query Builder locks up, you need to exit TBSM and close the browser window to unlock the Query Builder. If you want to save the complex query, increase the timeout value in the `server.timout` property on the TBSM Data Server `$TBSM_HOME/etc/TBSM_server.props` file.
10. You can select which field you want to filter, and then click **Append Field**. This will add the field name into the `WHERE` clause field. You can add multiple field names in the `WHERE` clause field. At any time, you can click **Apply** to see the results against the current query.

    For example, select the `TBSMDEMO.ESDA_BUSINESSUNITS` table and the `LOCATION` field. Click **Append Field** to add `LOCATION` to the **WHERE** clause. Enter `LOCATION='New York'` to and click apply to see all the rows for New York.
11. Click **Next**. The next pane of the wizard displays. The pane shows the results of your query.

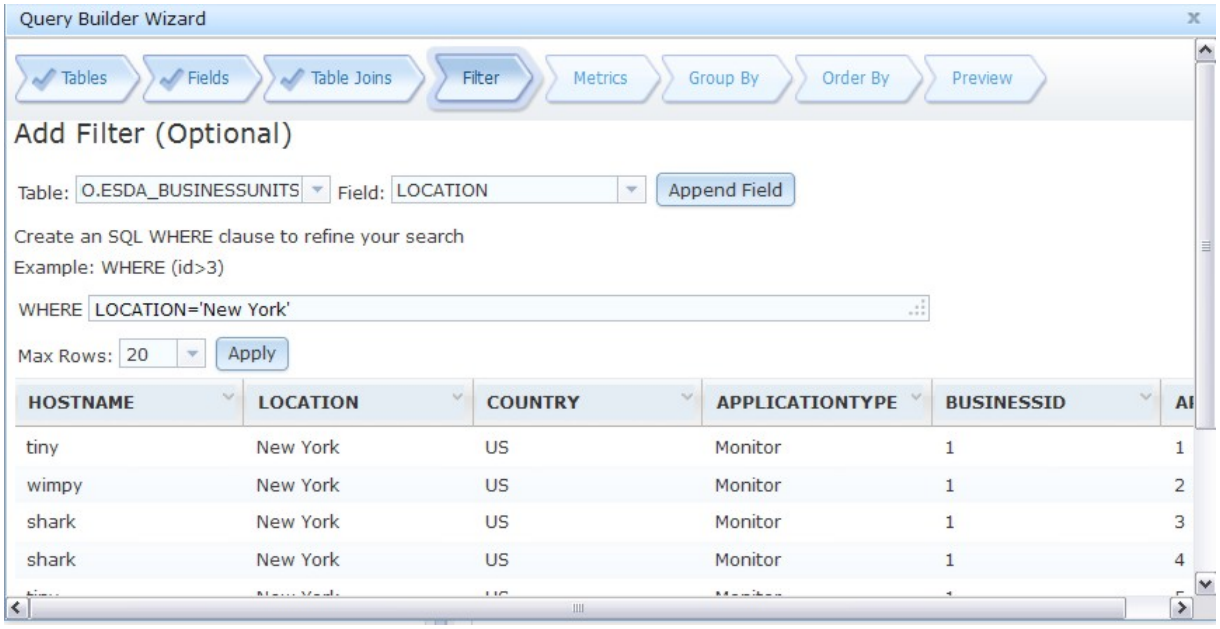


Figure 19. Query Builder filter results

If the data displayed in the **Add Filter** pane is not what you want, you can click the **Tables**, **Fields**, or **Table Joins** buttons at the top of the wizard to go back and correct your entries.

Results

If the query returns no data or is invalid, the query results would be: `No Data To Display Bad Query` or no data.

Filtering the query data

This task describes how to filter the query data.

About this task

When the correct data is displayed in the **Add Filter** pane, you can further refine the results by adding a filter in the SQL WHERE Clause section of the window.

For example, if you wanted to filter the data so that you see only the servers that have more than 20 seconds of downtime, you would complete the following steps:

Procedure

1. Select `fetcherappstatus` in the **Choose Data Source and Tables** drop-down list.
2. Select `fetcherappstatus.downtime` in the **Choose Fields from the Selected Tables** drop-down list.
3. Click **Append Field**. This will add the field name into the `WHERE` clause field.
4. Type `>20` as the operator.

   Valid operator characters: `<>, =, !=, >, <`.
5. Click **Apply** to see the results against the current query.

   You now see only the servers that have more than 20 seconds of downtime.

   Note: You can add multiple field names in the `WHERE` clause field.
6. To display the original data, delete the contents of the `WHERE` text field and click **Apply**.
7. Click **Next** to open the **Metrics** pane.

Add metrics

This task describes how to add query metrics in Query Builder.

About this task

If you are defining a data fetcher query in the Query Builder, you have the option of adding metrics.

In this example, metrics are created that show the average downtime for servers.

Clicking **Next** from the **Add Filter** pane takes you to the **Add Metrics** pane where the two fields that were selected from the tables display as the column names.

To add the average downtime for these servers, complete the following steps:

Procedure

1. To add a function, select one from the list. For example, AVG.
2. Select a field name. For example, select a field that shows downtime, such as FETCHERAPPSTATUS.DOWNTIME.
3. In the **As** field, replace expr1 with a column name that describes the metric you are adding. For example, add AVERAGEDOWNTIME.
4. Click **Apply** to see the results.

   When you create a metric, you need to use the fields you selected when you built your query.

Adding Group Bys

If you are defining a data fetcher query in the Query Builder, you have the option of adding grouping.

About this task

You can optionally group your query results.

Clicking **Next** from the **Add Metrics** pane takes you to the **Add Group Bys** pane.

To add grouping, complete the following steps:

Procedure

1. In the Select Fields to Groupby list, if you specified a metric and have not explicitly set any "Group By" fields - then the group by fields will automatically be added to the query by TBSM.

   For example, if you added SERVERNAME was added, the servers are now grouped, and the metric is shown for each server name.

   Note: If you select no fields on the **Choose Fields from the Selected Table** window, the query builder selects all fields for the table. In other words, you select * from the table. When you select all the fields from the table in this way and you want to use the Group by clause, the fields will not be automatically added to the Group by clause. In this case, you need to add the fields to the Group by clause by hand.
2. You can add more groups to the GroupBy by selecting another field and clicking the **Add All** button.

   The figure below shows the results after adding applicationname to the GroupBy.

   You can delete any field names that you add by clicking the **Remove All** button in the GroupBy toolbar, but you cannot delete the fields that were automatically added.

Adding Order Bys

If you are defining a data fetcher query in the Query Builder, you have the option of adding ordering.

About this task

You can order your results by a metric or numerical field.

Clicking **Next** from the **Add Group Bys** pane takes you to the **Add Order Bys** window where the two fields that were selected from the tables display as the column names.

To add ordering to the query results, complete the following steps:

Procedure

1. Change the order by selecting `asc` or `desc` in the **Direction** drop-down list.
2. Now scroll down to the `Available Fields` table.
3. Select the field you want to order by.
4. Click the **Add** button.
5. Click **Apply** to see updated query results.
6. When you have the query the way you want it, click **Next** to preview the query, or click the **Back** button to return to previous windows to modify the query.

   When you click **Finish** on the Preview Query pane, the Query Builder closes and you return to the original editor (Data Fetcher or ESDA rule). You can see the query you created in the SQL Query text box.

Chapter 17. ESDA rules

This section describes how to work with External Service Dependency Adapters (ESDAs).

ESDAs overview

This topic describes ESDAs.

ESDAs let IBM Tivoli Business Service Manager (TBSM) dynamically import service hierarchies from any data source you can configure for TBSM. You should only attempt to use ESDAs if you understand SQL queries, regular expressions, and the underlying data for your service model.

Note: By default, the ESDA functionality is disabled. To enable ESDA functionality, set `//impact.esda.enable=true` in the `/opt/IBM/JazzSM/profile/installedApps/JazzSMNode01Cell/isc.ear/sla.war/etc/RAD_sla.props` file.

Accessing the ESDA sample data

This topic describes how to access ESDA sample data.

The sample data for this section is part of the TBSMDEMO schema included the TBSM database. To access this data create a data source for the TBSM DB2 database.

The tables used for ESDA examples are:

ESDA_APPLICATION
Application table for the examples

ESDA_BUSINESSUNITS
Business unit table for the examples

ESDA_HOSTS
Hosts table for the examples.

Creating the data source

To create a datasource for the TBSMDEMO data:

1. Select **Administration > Service Configuration** from the task view.
2. Select **Data** from the **Service Navigation** drop-down list.
3. Click the **New** button.
4. Enter these values for the data source:

   Data source name
   `tbsm_demo`

   User name
   DB2 user name.

   Password
   DB2 password

   Host Name
   DB2 host name.

   Port
   DB2 port. Default is `50000`.

Database
TBSM

Failover policy
Disabled.

5. Click **Test Connection**

   A message window displays:

   ```
   CTGBA0025I The test connection is open and functioning.
   ```

   If an error message displays, check your settings and test the connection again.

About TBSM auto-population solutions

This topic describes two TBSM auto-population solutions.

TBSM provides two solutions for automatically configuring services based on template rules:

- Using a data fetcher in combination with auto-population rules.

  Auto-population rules are appropriate for simple models that do not constantly update. They work best when you can obtain all the data (ObjectServer or SQL data source) for the model from a single row in one table
- ESDA auto-population rules work better for more complex models that update data in real time, when the model structure changes quickly whenever underlying data changes.

ESDA rules and service structure

This topic describes ESDA rules and service structure. This topic also describes seed services.

You configure ESDA rules for the service templates in your service model. The ESDA rules consist of queries and naming expressions that discover and configure the child and parent services defined by the aggregation rules in the seed service template. For each of these service templates, the ESDA rule creates multiple services that share the common rules defined in the service templates. You can configure ESDA rules that discover services for each of the service templates in this service model. The seed template can be at any level, depending on your configuration.

Using the template structure shown in Figure 20 on page 113, you configure the ESDA rules in the seed service template: `BUSINESSUNIT`. Figure 20 on page 113 shows the service template structure where the `BUSINESSUNIT` is the seed service template, `CUSTOMER` is the parent service template, and `APPLICATION` is the child service template. Figure 20 on page 113 is an example of a possible configuration.

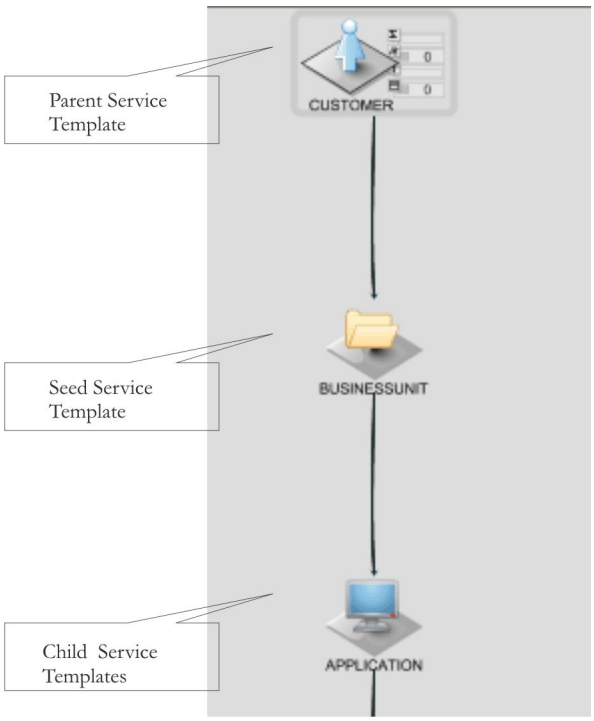


Figure 20. Example service template structure

Before you configure your ESDA rules, you should analyze your source data and create a service structure by defining the relationships between TBSM service templates. You should sketch these relationships on paper before you start your ESDA rule configuration.

Creating a seed service

After you create the service template structure, you need to create a seed service as a starting point for your ESDA rules. The rule cannot discover any new services until you create a seed service and assign it to template with an ESDA rule.

When a service has ESDA rules configured in its assigned service template, TBSM automatically executes the necessary query and logic to discover any new services related to the seed service. TBSM then combines the dynamically discovered services with any existing child or parent services of the seed service. As the data changes in the external data source, TBSM adjusts the service model to reflect the changes by adding and removing services and relationships in the service model.

Child and parent rules

This topic describes child and parent rules.

With an ESDA rule, you can create a query and naming expressions for both child and parent services of the seed service. Child rules discover child services for the seed service. Parent rules discover parent services for the seed service. The **ESDA Model Rule** window lets you select the seed service template and whether the rule discovers child or parent services.

Note: When you configure your ESDA rules, you should configure your child rules first, confirm that they work, and then configure your parent rules.

ESDA queries

This topic describes ESDA queries.

For each level on the service model, you create a query in the ESDA rule. The ESDA queries select data from an external data source and this data is used to build service instances. You can type in the query by hand, or select the **Query Builder** button to browse the data source and build your query from the **Query**

Builder window. You use the data from the query to select the data needed to discover the services for your service model.

Note: All `Select` statements should reflect the case sensitivity of the database being used.

Whenever a service is created with an ESDA rule, the field and value pairs retrieved from the data source are added as additional properties of the ESDA services that are created. You can use these additional parameters in custom policies for ESDA and other rules. The names of the additional properties will be exactly the name of the fields returned by the query. For more information about policies, see the *IBM Tivoli Business Service Manager: Customization Guide*.

ESDA service expressions

This topic describes ESDA service expressions.

For each level of your service model, you can configure ESDA rules that define the basic service attributes for each discovered service. You configure expressions that create a unique service name, a description, and a display name for a given service. These expressions use the values from the SQL query you created for your rule. You can use expressions to combine these values with data from multiple fields or with data from the seed service. You can also parse the data using Netcool/Impact functions and expressions. For more information about using expressions in TBSM, see the *IBM Tivoli Business Service Manager: Customization Guide*.

ESDA rule triggers

This topic describes ESDA rule triggers, On-Demand triggers, and automatic updates.

TBSM triggers ESDA rules on demand, or when new service data is discovered in the service template data source. The data source can be an ObjectServer or an SQL database. You can update the service model interactively, or you can set up the service model to update automatically based on new data.

On-Demand triggers

When you click a service with ESDA rules in the **Service Navigation** portlet, TBSM executes the ESDA rules, updates the service model, and renders the service model in the *Service Editor*.

When you click the **+** sign for a service with ESDA rules in the **Service Navigation** portlet, TBSM executes the ESDA rules, updates the service model, and renders the updated service model in the **Service Navigation** portlet.

Automatic updates

The ESDA rule can also be triggered when a new or updated event or database row is detected by TBSM. To enable automatic ESDA updates, you must configure an incoming-status rule for the service template at the bottom level in your service model. If the new data matches an incoming-status rule in the service template, it triggers the creation of a new service based on the new data. TBSM calculates the status propagation of all affected services and discovers the parents of the service recursively.

You configure rules for importing child and parent services in service templates assigned to the seed service. To discover the children or parents of a service, TBSM goes through all the templates that the service is assigned to and applies all the ESDA rules in the service templates.

RAD shell commands

RAD shell command "discoverHierarchy" can be used to recursively discover the ESDA hierarchy from a given starting instance. This is equivalent to expanding the seed service in the Service Tree. Run this command after you have run the invalidateServiceName command, for the seed service, to run ESDA rules.

A scripted invalidateServiceName() and discoverHierarchy() can be created for discovery of children services, and this script can be run at the same time as doing the XMLToolkit discovery. The Toolkit log can be checked for message indicating the successful processing of a IdML book. The message looks like:

[2016/07/14-00:25:54.380] com.ibm.tbsm.cltools.service.ASIRADObserver run [6] GTMCL5290I: Book <book name>.xml processed successfully.

The script can run every minute or two and checks if there has been a new GTMCL5290I message written to the Toolkit log, and thus runs the invalidation/discovery for the child services.

Limitation

1. The ESDA Automatic trigger is not designed for automated service model generation, and it is rather for dynamic service model generation on user action.

   Note: For automatic service model generation, users need to use Auto-Population Rule. Dynamic service model generation means, user clicking the plus icon on the Service Tree, clicking the showing levels up or down in Relationships/Basic Relationships view in Service Viewer portlet in Service Administration and Service Configuration portlets.
2. By clicking the + icon in Service Tree or levels up/down in the Service Viewer will not show fresh data unless the Service Instance being clicked was set to invalid. The "Invalidate" button in the Service Editor portlet can be clicked to invalid the Service Instance. Thus for Dynamic/Instance:

   Note: ESDA rule triggers on invalid Service Instances. For valid Service Instances, TBSM will show what's in the current memory.
3. The discovery of the data source and showing fresh data will work only for invalid service instances.

   Note: The relevant properties in this situation are impact.sla.runesdasonstartup and impact.sla.onlymanualinvalidation. That is, whether the ESDA will run on server startup and whether only manual invalidation is allowed.

Service persistence

This topic describes service persistence.

For each service template, you can specify whether services discovered with ESDA rules remain in the TBSM database based on the data retrieved in the SQL query.

Setting the service model structure

This topic describes how to set the service model structure using an example.

You cannot configure ESDA rules until you have defined the structure of your service model. You define your TBSM service model structure by defining the relationships between the service templates in your service model. Figure 20 on page 113 shows an example service template structure. The status of a service assigned to a parent service template depends on the status of services assigned to child service templates. You define these relationships by configuring aggregation rules in your service templates.

Example service model structure

This example shows how to define the relationships between the service templates for the CUSTOMER, BUSINESSUNIT, APPLICATION, and HOST service templates, as described in Table 38 on page 115.

Table 38. Example service model structure

Parent Service Template	Child Service Template	Condition
CUSTOMER	BUSINESSUNIT	The status of CUSTOMER services changes based on the status of BUSINESSUNIT services.
BUSINESSUNIT	APPLICATION	The status of BUSINESSUNIT services changes based on the status of APPLICATION services

Table 38. Example service model structure (continued)		
Parent Service Template	**Child Service Template**	**Condition**
APPLICATION	HOST	The status of APPLICATION services changes based on the status of HOST services

Figure 21 on page 116 shows the example service model structure in the **Service Navigation** portlet.

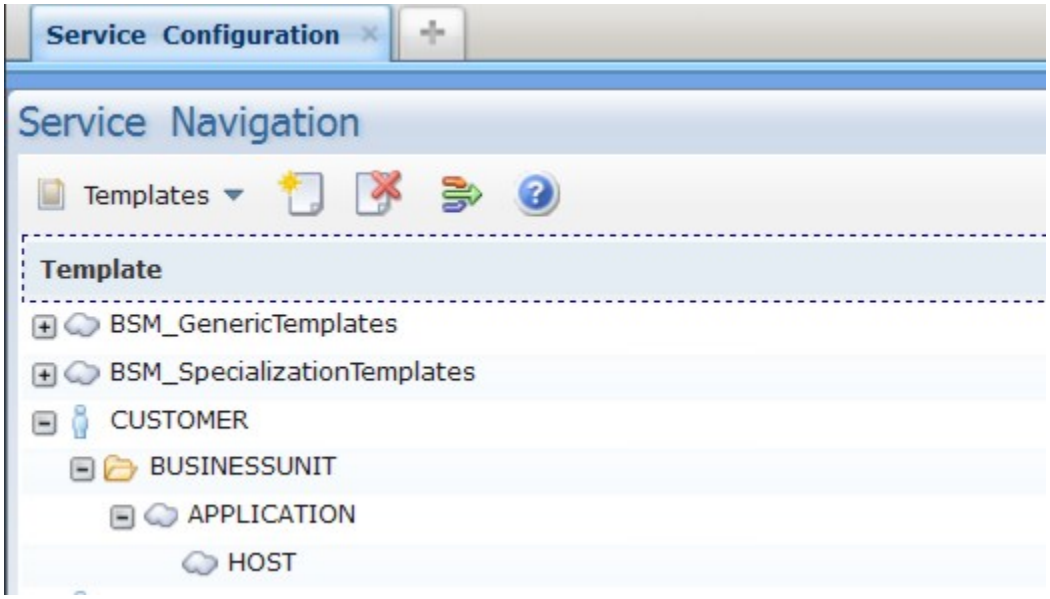


Figure 21. Example service model structure

Each of these service templates describes a type of TBSM service that shares common rules and attributes. For each of these service templates, you create many services that share common rules and attributes. Each service is uniquely defined by the **Service Name** attribute. The ESDA rules let you select and filter the data TBSM needs to create the individual services assigned to each service template.

Figure 22 on page 117 shows the settings you use to configure the BUSINESSUNIT service template as a child service template of the CUSTOMER service template. In this case, the parent CUSTOMER service's status changes to bad when any of it's child BUSINESSUNIT services status changes to bad.

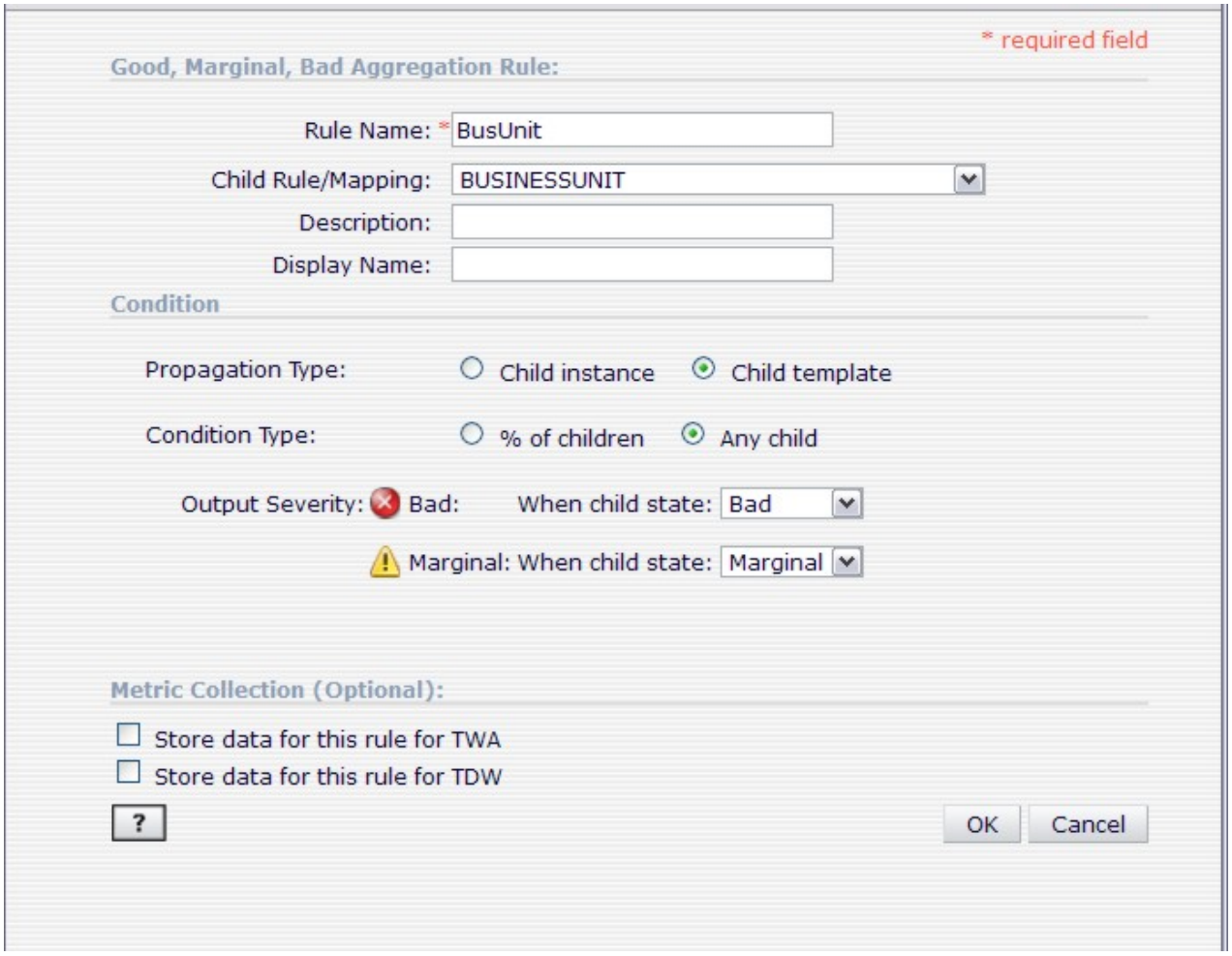


Figure 22. Good, marginal, bad aggregation rule settings

Selecting child or parent rules

This topic describes how to select child or parent rules.

About this task

When you configure an ESDA rule for a service template, you need to specify whether a given ESDA query is used to discover child or parent services. You should configure and test your child rules before you configure your parent rules. Before you start configuring your ESDA rules, you must create a seed service instance by hand. The seed service is a starting point where you can add the parent and child services with the ESDA rule. Typically, you assign your seed service for your top-level service template. For example, assign your seed service to the CUSTOMER service template.

To select the service type:

Procedure

1. From the **Service Navigation** drop-down menu, select **Templates**.
2. From **Templates**, click the service template you want for the new ESDA rule.

   For example, in Figure 23 on page 118 you could select the CUSTOMER template.

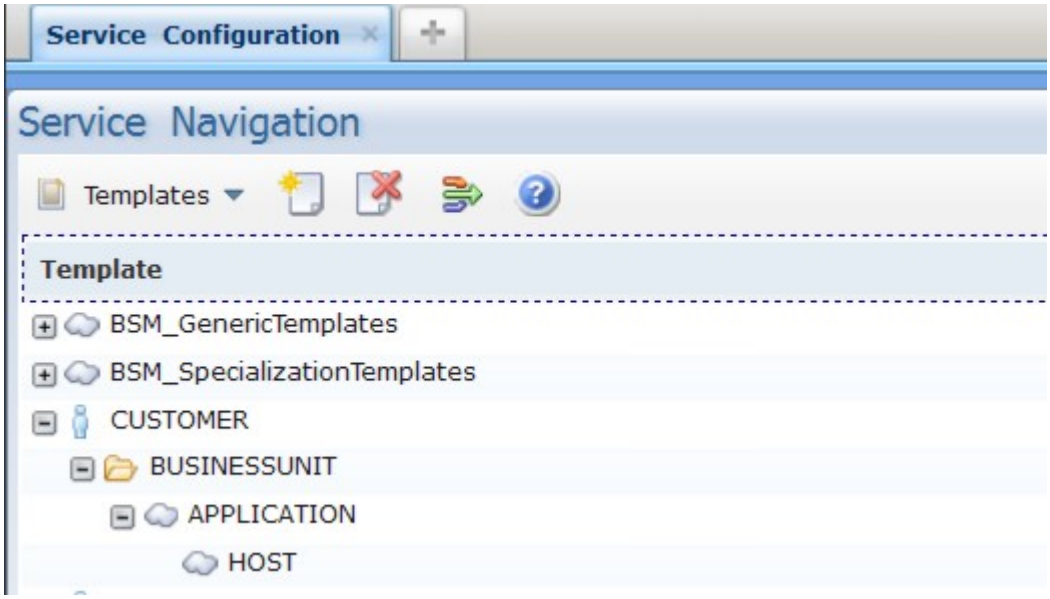


Figure 23. Example service template structure

3. If the **Edit Template '***Template Name***'** tab is not open, click the tab to open it.
4. Click the **Create ESDA Rule** button.

   The **ESDA Model Rule** window opens. The **ESDA Model Rule** window shows the service templates related to the selected template in the two columns: one on each side of the window. The values in these **Child Rules** and **Parent Rules** columns are determined by the structure of your service model.
5. Enter a rule name or accept the default. The rule name you select here will be added to each template where you configure a child or parent rule.
6. To configure a rule that discovers child services for a template, click the service template name in the **Child Rules** column on the left side of the window.

   The **ESDA Model Rule** window refreshes to show the settings you need to configure for this part of the ESDA rule. The settings let TBSM discover child services for the selected seed service template.

   In Figure 24 on page 119, the `BUSINESSUNIT` service template is selected in the **Child Rules** column. The query and expressions entered in this window are used to discover child services for the `BUSINESSUNIT` service template. In this case, the child services are assigned to the `APPLICATION` service template. That is, the seed service template is `BUSINESSUNIT`, and the child service template is `APPLICATION`.
7. To configure a rule that discovers parent services for a service template, click the service template name in the **Parent Rules** column on the right side of the window. You should configure all your **Child Rules** before you configure your **Parent Rules**.

   The **ESDA Model Rule** window rule refreshes to show the settings you need to configure for this part of the ESDA rule. The settings let TBSM discover parent services for the selected service template.

   For example, Figure 24 on page 119 shows similar settings for this section of configuration.

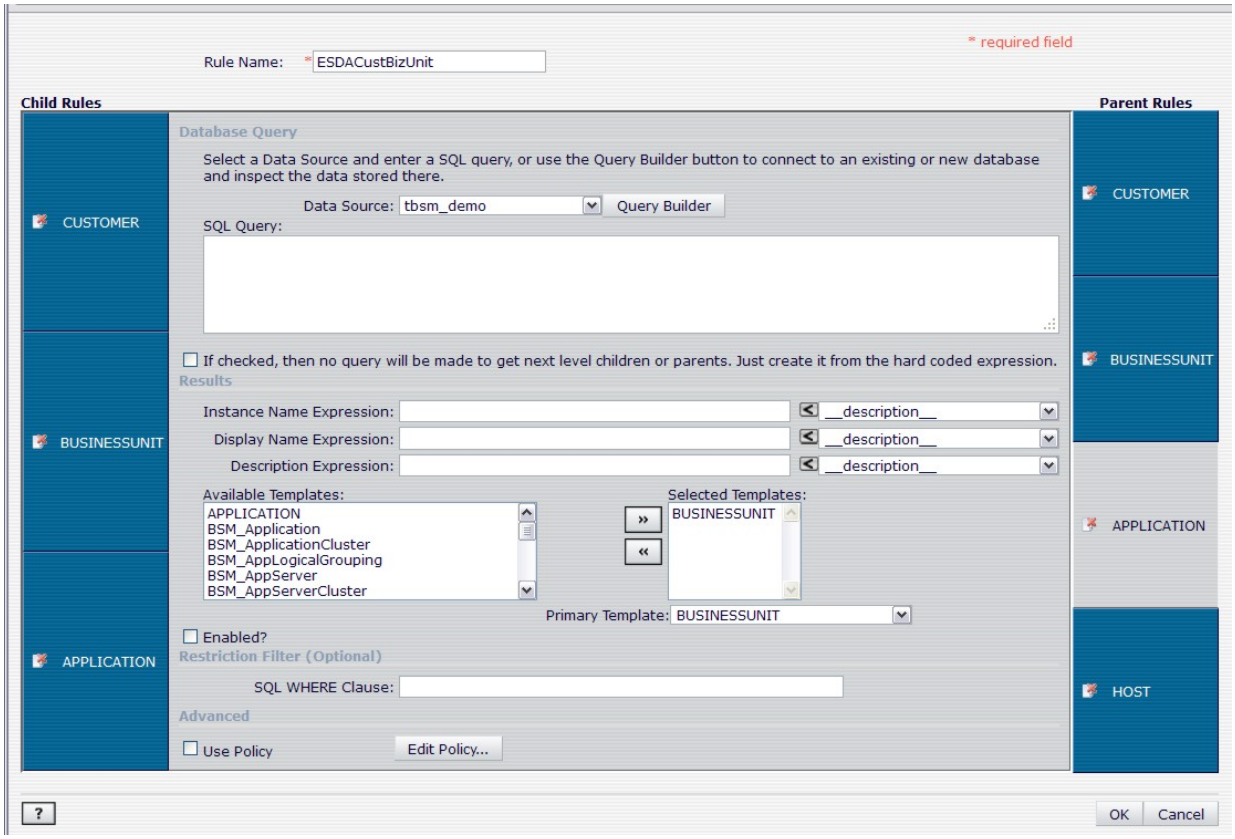


Figure 24. ESDA Model Rule window: parent rules

In Figure 24 on page 119, the APPLICATION service template is selected in the **Parent Rules** column. The query and expressions entered in this window are used to discover parent services of the APPLICATION service template. In this case, the seed service template is APPLICATION and the parent services are assigned to the BUSINESSUNIT service template.

8. When you have selected the service template for either parent or child rules, you will see the template in the **Child/Parents** rules column.

   If you have multiple templates for a given level, you will only see one template in the **Child/Parents** rules column.
9. To delete a rule, click the red **X** in the box for the rule you want to delete.

ESDA query concepts

This topic describes ESDA query concepts, ESDA data sources, and selecting only the data you need.

After you have selected the seed service template for either child or parent rules, you create an SQL query to select the service data from an external data source. The SQL query selects a set of data you use to obtain the service names for the child and parent services of a given seed service.

ESDA data sources

Before you create an ESDA query, you must configure the data source you want to use for TBSM. You can configure the data source for TBSM from the **Edit Data Source** tab.

Note: If you use the Query Builder to create your query, the fields you select with your query are added to the drop-down lists in the **Results** section of the window for the **Instance Name**, **Display Name**, and **Description** expression fields.

Important: select only the data you need

Whenever you create a query for an ESDA rule, it is important that you only select the tables and fields you need to identify unique service names for the services you want to discover. The `Select *` clause should only be used in cases where you need data from all the columns in a table. Otherwise, your queries will take more time and system resources to execute.

Figure 25 on page 120 shows the **Query Builder** window with the necessary fields selected from the `TBSM.DEMO.ESDA_APPLICATION` table. In this case, you do not need the `BUSINESSID` and `hostname` fields to define the services assigned to a service template for applications.

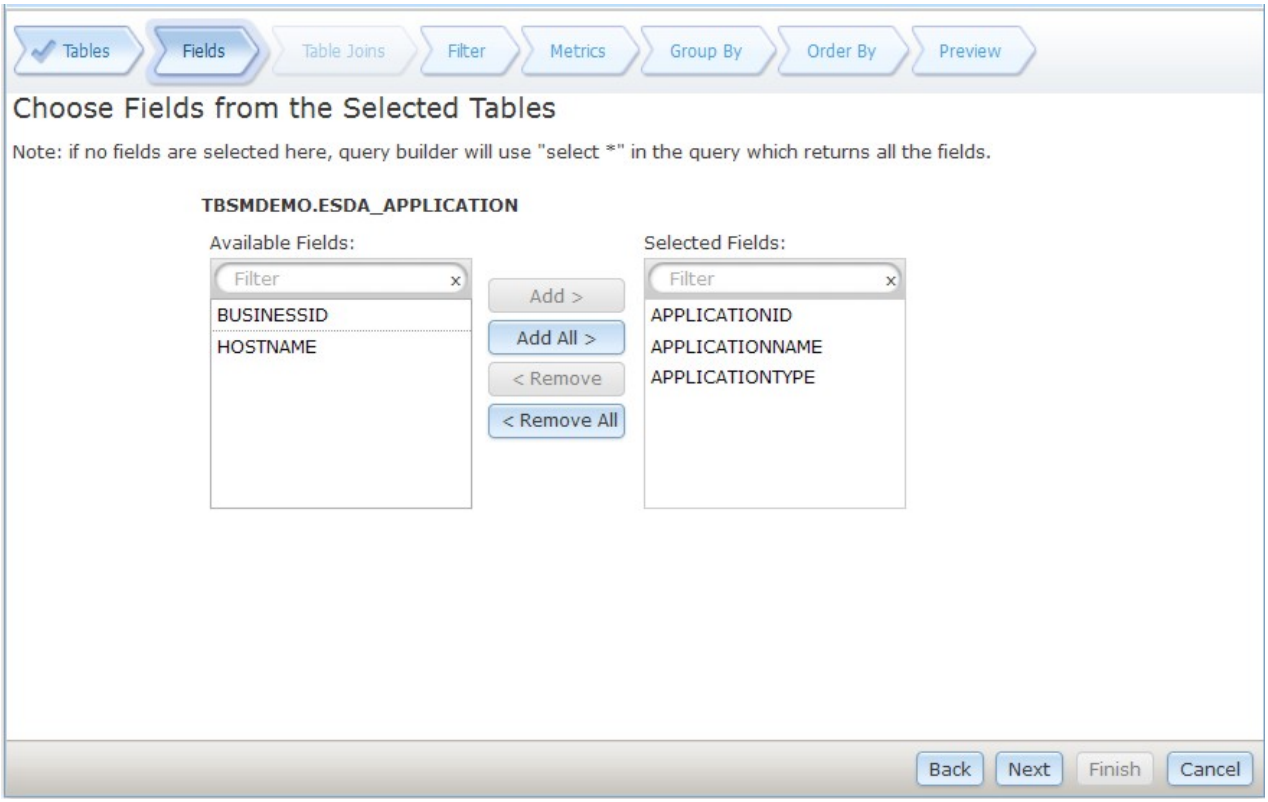


Figure 25. Selected Fields from TBSM.DEMO.ESDA_APPLICATION table.

In the following example, the query only selects the fields you need to identify the services from the `TBSMDEMO.ESDA_APPLICATION` table.

```
select APPLICATIONID,APPLICATIONNAME,APPLICATIONTYPE from TBSMDEMO.ESDA_APPLICATION
where BUSINESSID = __BUSINESSID__
```

This query also uses the *__fieldname__* notation. This notation lets you specify a variable field value. In this example, both the `TBSM.DEMO.ESDA_APPLICATION` table, and the `TBSM.DEMO.ESDA_BUSINESSUNIT` table contain a key field named `BUSINESSID`. The query only selects rows where the value of the `BUSINESSID` field (in `TBSM.DEMO.ESDA_APPLICATION` table) matches the field value (in `businessunits` table) for the seed service. This notation refers to the value for the seed service instance.

For the business services in the service model, this example query dynamically imports the data about the child services of a given business unit service. Each business unit service has a field called `BUSINESSID` which was set by the ESDA query that created the business unit services from data in the `TBSM.DEMO.ESDA_BUSINESSUNIT` table.

The data about the child service is in a table called `TBSM.DEMO.ESDA_APPLICATION`, which also has a `BUSINESSID` field that maps each application to a given business unit. The ESDA rule uses the data from the `TBSM.DEMO.ESDA_APPLICATION` table to import the corresponding applications as child services of

the matching business unit service. That is, when the BUSINESSID value matches in both tables, the ESDA rule creates a child application service for the matching business unit service.

Entering an ESDA query

This topic describes how to enter an ESDA query.

About this task

You can manually enter an SQL query in the **SQL Query** text field if you have already configured the data source you want to use for TBSM or you can use the Query Builder. To create a query in your ESDA rule:

Note: If you are not familiar with the tables and all the fields in the database you want to query, it will be much easier to use the Query Builder. The Query Builder lists all the tables and their fields and allows you to select them from the lists. If you use the Query Builder, you will avoid spelling and case sensitivity problems.

Procedure

1. Select a data source from the **Data Source** drop-down list.
2. Enter your query in the **SQL Query** text field.

   Figure 26 on page 121 shows an SQL query that selects the rows where the value of the BUSINESSID field matches the BUSINESSID field value (__BUSINESSID__) for the seed service. In this case, the ESDA is selecting a child service assigned to the APPLICATION service template for the seed services assigned to the BUSINESSUNIT service template.

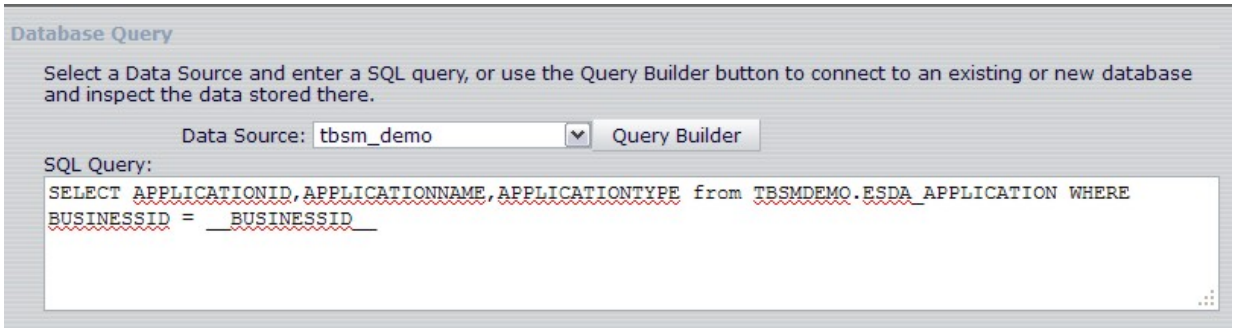


Figure 26. Example ESDA database query

3. Enter the **Instance Name**, **Description**, and **Display Name** expressions for the services.

Working with ESDA service expressions

This topic describes how to configure ESDA service expressions.

After you create an SQL query to select the service data you want from a data source, you need to configure expressions that create unique attributes for each discovered service. You configure expressions that create a unique name, description, and display name for a given service in the **Results** section of the **ESDA Model Rule** window.

Note: You must make sure that each **Instance Name Expression** creates a unique service name for each service you want to discover. If the expression results are the same for multiple rows, the ESDA rule will only create a service for the last row that matches the **Instance Name Expression**.

These expressions use the values from the SQL query you created for your rule. You can use expressions to combine these values with data from multiple fields or with data from the seed service. You can also parse the data using Netcool/Impact functions and expressions. For more information about using expressions in TBSM, see the *IBM Tivoli Business Service Manager: Customization Guide*.

To configure service expressions:

1. Select the seed service template and rule type you want.
2. Configure the SQL query to select the data you want to use for your ESDA rule. You enter expressions in these fields to obtain the service attributes from the data selected by your SQL query.
3. Enter the information in the service expression fields using the information in Table 39 on page 122.

   Note: When entering name expressions, select the field values from the drop-down menu whenever possible to avoid incorrectly typing the field names for the name expressions. Selecting the values eliminates spelling and case errors that cause your rule to malfunction. Some data sources, such as MS-SQL, have case-sensitive field names. If you use the Query Builder to build the query, all the fields for query will be in drop-down lists for the naming expressions In this way you can avoid spelling and case errors in the field names.

Table 39. ESDA rule configuration window

Field name	Description
Instance Name Expression	Enter an expression for the service name. For example, enter: `"Application" + '__serviceinstancename__' + applicationid)` to create the name from the word `Application`, + the name of the seed service, + the value of the `applicationid` field. The resulting name would be like: `ApplicationMIS24`. For more information about expressions, see the *IBM Tivoli Business Service Manager: Customization Guide*. **Important:** The result of the naming expression field for automatically created services must not contain these special characters: `" < > \ * ? | ;` The result of the naming expression must not be longer than 127 characters. Names longer than 127 characters can cause display and performance issues.
Display Name Expression	Optional. If you want a different display name other than the default name defined in the **Instance Name Expression** field, enter an expression in this field. This expression must be in the TBSM expression language. Otherwise, the name will be the same as the service name defined in the service template status rule. For example enter: `applicationtype + applicationid` to create the display name from the values of the `applicationtype` and `applicationid` fields. This name is displayed as the service name in the **Service Navigation** portlet tree.
Description Expression	Optional. If you want to include a description of the discovered service, enter an expression in this field.
Available/Selected Templates	Select the service template or templates you want assigned to the discovered services. Use the >> and << buttons to move the service templates between the Available Templates and Selected Templates lists.
Primary Template	If you have selected multiple templates for the discovered services, select the primary template you want assigned to the service from the **Primary Template** drop-down list field.

Table 39. ESDA rule configuration window (continued)	
Field name	**Description**
Enabled	Select the **Enabled** check box to activate the ESDA rule.
Restriction Filter (Optional)	Enter an SQL WHERE clause in this optional field if you want to apply additional conditions when the ESDA rule is run. This condition is evaluated against the fields of the seed service. For example, if you only want a service created if the `applicationid` field of the seed service has a value, type `applicationid <> ''` (`applicationid` field value not equal to empty string) in this field.

4. When you have finished configuring you service expressions, select the **Enabled** check box to enable the rule. The ESDA rule will not function until you select the **Enabled** option.
5. Click **OK** to save your settings and close the window.

Note: When you save an ESDA rule definition, a message warns you that the incomplete rule definitions will not be saved. The system displays this message if you opened a child or parent rule page, but did not specify all the required fields. When you click the **OK** button, the system reports these incomplete definitions for any ESDA child or parent rules. This includes the rule page that displays when you first open the **ESDA Model Rule** window. If you do not need the incomplete rules for your ESDA specification, then click **OK** for the message and only the completed rules will be saved. Otherwise, open the appropriate rule page and specify all the required fields.

Testing ESDA rules

This topic describes how to test ESDA rules, check an ESDA rule in the services tree, check an ESDA rule in the Service Editor, and check automatic updates.

By default, TBSM automatically checks for new services when you click a service that is assigned to a service template with ESDA rules.

Checking an ESDA rule in the services tree

To check that the ESDA rule is working, click the + for the seed service name in **Services** in the **Service Navigation** portlet. TBSM executes the ESDA rules for the service and displays the updated service model in the **Services Tree**.

Before you test the ESDA rule, you need to create the seed service and assign it to a service template.

Table 40. Example discovered services and their related templates	
Service	**Service Template**
Micromuse	CUSTOMER
Development	BUSINESSUNIT
Impact	APPLICATION
shark	HOST

Checking an ESDA rule in the Service Editor

To check that the ESDA rule is working, click the seed service name in **Services** in the **Service Navigation** portlet. TBSM executes the ESDA rules for the service and displays the updated service model in the Service Editor.

Checking automatic updates

If a service template with an ESDA parent rule also has an incoming-status rule, TBSM automatically updates the service model whenever a new or updated event for the service is detected. To check that the ESDA rule is working for automatic updates, send a test event or add a database row for the seed service. TBSM executes the ESDA rules for the service and displays the updated service model in the **Services Tree** and Service Editor.

Service persistence settings

This topic describes how to set persistence settings for a service template.

About this task

For each service template, you can specify whether services discovered with ESDA rules remain in the TBSM database based on the data retrieved in the SQL query. For example, you can specify that a service will be deleted if TBSM does not detect any activity for the service within a given time period.

Important: The search function in the Service Navigation portlet and Service Tree will only find services that are persisted. Services that are not persisted will not return as search results.

To set the persistence settings for a service template:

Procedure

1. Open the **Edit Template** '*Template Name*' tab for the service template you want to edit.
2. Click the **Edit Properties** link.

   The **Standard Template Properties** window opens.
3. Enter the settings for *Standard Template Properties* using Table 41 on page 124 as your guide.

Table 41. Edit Template Properties window

Field	Description
Make Auto-populated Instances Persistent	Accept the default of true if you have auto-population rules defined for this template and you want the automatically created service instances to be persisted in the database. If not, change the entry to false. Instances not persisted will be discarded when the TBSM server is shut down and will be recreated when another event drives the auto-population rules.
Make ESDA Instances Persistent	Accept the default of false or change the value to true if you have ESDA rules defined and want the service instances created by the ESDA rules to be persisted in the database. Instances not persisted will be discarded when the TBSM server is shut down and will be recreated when the ESDA rules are executed.
Invalidation Period (seconds)	Set a time period (in seconds) to check for updated service instance data. If the system does not detect any change in the service data for the specified time period, the service is invalidated.
Instance Lifetime (seconds)	Set a time period for the life time of the service data for invalidated service instances. After a service is invalidated, the system waits the specified time period before deleting the service instance.

4. Click **OK** to save the settings and close the window.

Chapter 18. Configuring services

This section describes how to configure IBM Tivoli Business Service Manager (TBSM) services.

Services overview

This topic describes services. This topic also describes the relationship between services and templates.

TBSM services represent services in your enterprise that are configured for TBSM. When you configure a service in TBSM, you create a representation of hardware or software in your company or organization. The service can also represent a customer or a collection of related services. For example, you can create a service that groups a collection of services that support your operations in a given geographical region.

Services and templates

When you create a service, you can assign it to one or more templates. The service template defines how the service responds to events, external data, and the status of other services. In other words, the services tagged with the rules from a template are the same type of service and therefore can use the same rules to evaluate service status.

Note: If you change rule or template names, you can cause issues with other features of TBSM. TBSM uses these names to reference the rule values for other rules, such as auto-population or ESDA rules and for custom displays such as service trees and view definitions.

If you want to change a rule or template name, change the name before you use the rule or template elsewhere in TBSM.

For example, when you create numerical and text-based rule, you need to create a custom service tree or view definition to display the output of the rule in the TBSM console. The trees and view definitions use the template and rule names to map the rule values to a display object. As a result, if you change the name of a rule or a template used in a custom display, the custom tree or view definition will no longer display the rule values properly.

Service configuration

This topic describes how to create and edit services.

About this task

When you have configured your service templates, you can add services and tag them with the rules for a given service template or templates.

Creating a service

You create services in the **Edit Service** tab of the Service Editor.

About this task

To create a service, click the **New** button in **Services** in the **Service Navigation** portlet.

The **Edit Service** tab is displayed in the Service Editor.

Service name restrictions

Important: TBSM checks the Service Name field for invalid characters. The Name field for services must not contain these special characters:

```
" < > \ * ? | ;
```

TBSM will not create a service when the Name field contains any of these invalid characters. If the object does not save, remove any special characters from the name field.

Service names longer than 127 characters can cause display and performance issues.

Service names longer than 127 characters will be truncated in the Service Details portlet display. This does not affect the functioning of TBSM, but the truncated service name will appear in the Service Details portlet.

If you use the Node column in Netcool/OMNIBus to create or match your service name: The node column is limited to 64 characters. If the service name is greater than 64 characters, it will be truncated and TBSM will not receive any events for the service.

You can either change the incoming status rule to use different service instance naming fields, or you can change the identification fields for a service with a long name. The identification field or fields can be different than the service naming field or fields. For example, if you have a service name that uses the Node column for its name, and the name is longer than 64 characters, the service will not receive events. To fix this, you can specify another field, such as a field that contains the IP address as the Identification field for the service.

Editing an existing service

You edit services in the **Edit Service** tab of the Service Editor.

About this task

To edit an existing service:

Procedure

Click the service name you want to edit in **Services** in the **Service Navigation** portlet.

The Service Editor is populated with the selected service.

Results

Table 42 on page 126 describes the elements of the **Edit Service** tab.

Table 42. Edit Service tab elements

Element	Description	References
Toolbar	The toolbar options let you save and refresh the service settings. You can also display online help.	"Edit tabs" on page 29
Service Properties	The **Service Properties** fields let you set the basic service information and select an SLA service level. You can also create and edit maintenance schedules for the service.	"Basic service properties" on page 127 "Maintenance schedules overview" on page 139 Chapter 23, "Service level agreement configuration," on page 165

Table 42. Edit Service tab elements (continued)		
Element	**Description**	**References**
Templates tab	The **Templates** tab allows you to tag the service with a template or templates. You only see the templates you have permission to view.	"Assigning templates to a service" on page 129
Identification Fields tab	If the service has incoming status rules configured, the settings in this tab let you view and change the identification field values for the service. For each service, TBSM searches incoming events for the service name values specified in the identification fields. If the service has no associated incoming status rules, this tab is blank.	"Service Identification fields" on page 129
Dependents tab	The settings in this tab let you select child services for the service you are editing. The list of available dependent services can be filtered by template and service name.	"Service dependency configuration" on page 131
Additional tab	This tab allows you to set the sort order of the service in the Service Navigation portlet, set GIS properties for map based views, and set values for any additional properties that have been configured for this service. If the service has SLA settings, you can also set the penalty amount in dollars for each hour of downtime.	"Additional service settings" on page 132
Security tab	The settings in this tab let you assign user and group permissions for viewing and editing the service. You only see this tab if you have the `tbsmServiceAdmin` user role for the service.	Chapter 21, "Granting user and group permissions to templates and services," on page 145

Basic service properties

This topic describes basic service properties, and how to set them.

About this task

The Service Properties section of the **Edit Service** tab lets you specify the following information for a service:

- Service name, description, and display name

  Note: The Name field for services must not contain these special characters:

  ```
  " & < > \ / * ? | ( ) : ;
  ```

  Otherwise, these objects may not be saved properly in the TBSM database. If the object does not save, remove any special characters from the name field.

  Note: Service names and events: If your service name is longer than 64 characters, the name will be truncated in Netcool/OMNIBus ObjectServer events due to field length limits in the alerts.status Node field. As a result, the service name in the status and TBSM-generated events will not match the service name in the TBSM database. These events cannot be correlated with the service they affect.
- SLA level for services with SLA settings

- Maintenance schedules for the service

Note: The Name field for rules, data sources, maintenance schedules, view definitions and other objects must not contain these special characters:

```
" & < > \ / * ? | ( ) : ; $ ! %
```

Names must not contain spaces. Otherwise, these objects may not be saved properly in the TBSM database. If the object does not save, remove any special characters or spaces from the name field.

Setting basic service properties

About this task

After you have created a service and saved it, refresh the **Service Navigation** portlet to see it displayed in the service list. The service has an icon displayed next to it, which appears with the service throughout the system to more easily identify it.

The display icon selected depends on the template it is attached to. If the service is attached to more than one template, the primary template determines the service icon. If you do not attach the service to a template, the service icon from the default template is assigned to the service.

To select basic service properties:

Procedure

1. Type a unique service name in the required **Service Name** field. If a service has incoming status rules, the service name you enter here is used as the default value for the identification fields. If you have more than one identification field, you may need to change the default.

   Important: TBSM checks the Service Name field for invalid characters. The Name field for services must not contain these special characters:

   ```
   " < > \ * ? | ;
   ```

   TBSM will not create a service when the Name field contains any of these invalid characters. If the object does not save, remove any special characters from the name field.

   Service names longer than 127 characters can cause display and performance issues.

   Service names longer than 127 characters will be truncated in the Service Details portlet display. This does not affect the functioning of TBSM, but the truncated service name will appear in the Service Details portlet.
2. If the service has SLA rules defined, select an SLA service level in this drop-down list.
3. If you want the service display name to be different from the service name, enter the name in the **Display Name** field. If you leave this field blank, the service display name will match the value of the **Service Name** field.
4. If the service has scheduled maintenance times, select or set the times with the drop-down list or buttons for the **Maintenance Schedule** field.
5. Click **Save** in the toolbar.
6. In the **Service Navigation** portlet, the new service with a default icon should appear in the list automatically. If it does not, select the **Refresh** button on the portlet title bar.

Assigning templates to a service

This topic describes how to assign templates to a service.

About this task

When you assign a template to a service, you *tag* the service with the template. Using templates eliminates the necessity of creating the same rules for a service type more than once. If you do not tag a service with a template, TBSM assigns the default template.

To tag a service with service templates:

Procedure

1. From the **Edit Service** tab, click the **Templates** tab.

   The **Templates** tab opens. The **Templates** tab shows all the templates you have permission to assign in the **Available Templates** list. The **Selected Templates** list shows all the templates assigned to the service.
2. To assign a template to a service, highlight the template or templates you want to add from the **Available Templates** list.

   Press **Ctrl** or **Shift** while clicking to select multiple templates from the list.
3. Click the **>>** button.

   The highlighted templates move to the **Selected Templates** list.
4. To remove templates from the **Selected Templates** list, highlight the template or templates you want to remove from the **Selected Templates** list.
5. Click the **<<Remove** button.

   The highlighted templates move to the **Available Templates** list.
6. If the service has multiple templates assigned, select a main template from the **Primary Template** drop-down list. This setting sets the display icon for the **Services** tree and service-dependency view in the Service Editor.

   If the assigned service templates have SLA settings configured, the **Primary Template** tells the system which SLA settings apply to the service.
7. When you have finished assigning templates, click **Save** in the toolbar.

Service Identification fields

This topic describes service identification fields.

If the service has incoming status rules, the settings in this tab let you view and change the identification field values for the service. For example, you may want to have TBSM check events for a different value than the **Service Name** field or check for multiple **Identification Field** values for a single incoming status rule.

When TBSM receives new data from an event or other data source, it checks for incoming status rules which match the event or data. For each incoming status rule in a template, TBSM checks the rule instance name field values for a match against the Identification fields of all the services assigned to the template. If you do not configure the Identification fields, the rule instance name value will be matched against the service instance name.

impact.sla.replayeventsonidentificationfieldchange

6.2.0.3

FP3 introduces the following new property:
`impact.sla.replayeventsonidentificationfieldchange`

This is a boolean property and the default value is `false`, which preserves the old functionality.

Up until FP2

When the `Identification Field` values for a TBSM service change, events which had previously been matched to a service instance will continue to affect the service instance status until they are deleted from ObjectServer, despite not matching any current `Identification Field` values.

From FP3

Setting `impact.sla.replayeventsonidentificationfieldchange=true` will ensure that TBSM re-evaluates the event matches for the changed service. If previously matched events no longer apply, then they are removed from the service.

To enable this function, you need to add following line to this file on the backend server `<INSTALL DIR>/etc/TBSM_sla.props`:

```
impact.sla.replayeventsonidentificationfieldchange=true
```

The property change requires a server restart for it to take effect.

Editing Identification fields

This task describes how to edit identification fields.

About this task

To edit the identification fields for a service:

Procedure

1. From the **Edit Service** tab click the **Identification Fields** tab.

   By default, this tab shows the value entered in the **Service Name** field. If the service template has multiple incoming status rules configured, this tab will show the identification field values for each rule.
2. Configure the identification fields using Table 43 on page 130 as your guide.

Table 43. Identification Fields tab elements

Tab Element	Description
Select All button 	Click **Select All** to select all the identification fields for deletion.
Deselect All button 	Click **Deselect All** to deselect all the identification fields selected for deletion.
Delete button 	Click **Delete** to delete the selected identification fields.
Rule Name column and Add button 	This column lists all the incoming status rules configured for the service's template. If you want to add another field value for the rule, click **Add** for the rule and a new row appears for the rule.

Table 43. Identification Fields tab elements (continued)	
Tab Element	**Description**
Select column	Select the check boxes in this column to mark identification fields for deletion.
Field Values column	This column shows the field name and value for each incoming status rule. By default, this tab shows the value entered in the **Service Name** field for the service.

3. When you have finished configuring the identification fields, click **Save** in the toolbar.

Service dependency configuration

This topic describes how to configure service dependencies.

The settings in the **Dependents** tab let you select child services for the service you are editing. The status of the service you are editing depends on the status of the dependent services you select here. The list of available dependent services can be filtered by template and service name.

Configuring service dependencies

To configure service dependencies:

1. From the **Edit Service** tab, click the **Dependents** tab.

   The **Dependents** tab opens.

2. Select the service dependencies using Table 44 on page 131 as your guide.

Table 44. Dependents tab elements	
Tab element	**Description**
Show services for templates	This list shows all the templates you have permission to view. If you want to only display the services for a single template, click the template name you want from this list.
Search the Available Services	Enter a string in the search field to search for services that contain the string you entered. The system displays the search results in **Available Services** list. Select the service you want from the list and click the **>>** button.
Available Services list	This list shows all the services you have permission to view in your system. You can press **Ctrl** or **Shift** while clicking to select multiple services in this list. The services you select from this list are added to **Selected Services** list when you click the **>>** button.
Search	Click this button to search the **Available Services** field.
Maximum results	This field limits the amount of data displayed in the Available Services List. If there are more services to be displayed than the maximum limit, then the data is sorted and the limit is applied. You can use the filter to narrow your search for services. The default setting is 100.
>> Add button << Remove button	Highlight a service or services and click the **>>** button to move services from the **Available Services** list to the **Selected Services** list. Click the **<<** button to remove the highlighted services back to the **Available Services** list.

Table 44. Dependents tab elements (continued)	
Tab element	**Description**
Selected Services list	This list shows all the services that you have selected as dependent (child) services. You only see services you have permission to view in system. You can press **Ctrl** or **Shift** while clicking to selected multiple services in this list.
Add from Service Component Repository	This button is available only if you install the `Service Component Repository` (SCR) templates. Click this button to open a window that lists the SCR services. See the *IBM Tivoli Business Service Manager: Administrator's Guide* for more information about managing these services.

3. When you have finished selecting dependent services, click **Save** in the toolbar.

Additional service settings

This topic describes additional service settings.

The settings in the **Additional** tab let you configure the following settings:

- GIS map coordinates of the service location
- Sort order of the service in the **Service Navigation** portlet
- If the service has SLA settings, the penalty amount in dollars for each hour of downtime
- Any additional properties that have been configured for this service

For information about how to create a key performance indicator (KPI) for a service, see "Specifying key performance indicators for services" on page 101.

Setting the GIS position for a service

This topic describes how to set the GIS position for a service.

About this task

When you set the GIS coordinates for a service, the service location can be displayed on a map in the **View Service** tab of *Service Editor*. To set the GIS coordinates for a service location:

Procedure

1. From the **Edit Service** tab, click the **Additional** tab.

   The **Additional** tab opens.
2. Enter the GIS coordinates using Table 45 on page 132 as your guide.

Table 45. Additional tab elements: GIS positioning	
Tab element	**Description**
Use GIS Positioning check box	Select this check box to enable the GIS positioning option.
City Locations drop-down lists and **Set** button	These drop-down lists contain the country and city locations preinstalled into TBSM. Select the country and city of your service and click the **Set** button. The GIS coordinates for the city you selected display in the **Longitude** and **Latitude** fields. You can also type the coordinates by hand.

Table 45. Additional tab elements: GIS positioning (continued)	
Tab element	**Description**
Longitude field	The longitude coordinate for your service location. If your service location is not in the **City Locations** drop-down lists, you can enter this coordinate by hand.
Latitude field	The latitude coordinate for your service location. If your service location is not in the **City Locations** drop-down lists, you can enter this coordinate by hand.

3. When you have finished selecting your service coordinates, click **Save** in the toolbar.

What to do next

For more information about using a custom GIS map, see the *IBM Tivoli Business Service Manager: Customization Guide*.

Setting SLA penalty calculations

This topic describes how to set the SLA penalty calculations for a service.

About this task

If a service's template has cumulative time-based SLA settings, you can set a cost estimate for each hour of downtime caused by an outage. If there is an hourly penalty set for the service's template, this template value is the default for the **HourlySLAPenalty** field.

To set the SLA hourly penalty calculations for a service:

Procedure

1. From the **Edit Service** tab, click the **Additional** tab.
2. Enter the dollar amount for the hourly penalty in the **HourlySLAPenalty** field.
3. Click **Save** in the toolbar.

Setting the sort order for a service

This topic describes how to set the sort order for a service.

About this task

You can change the sort order of the services in **Services** in the **Service Navigation** portlet by setting the service sort order in the **Additional** tab. If you do not enter a value in the **Order** field, the services are listed in alphabetic order in the **Services** tree. To set the sort order of a service:

Procedure

1. From the **Edit Service** tab, click the **Additional** tab.
2. In the Additional Parameters section, enter a number or a letter-number combination in the **Order** field. The lowest number is displayed at the top of the service tree.

   For example, a service with a 3 in the **Order** field displays closer to the top of the tree than a service with a 4. You can also enter a combination of letters and numbers. For example, 3a displays closer to the top of the tree than 3b.

   If two services have the same number in the **Order** field, they display in relative alphabetic order. That is, a service starting with A is closer to the top of the tree than a service starting with B.
3. When you have finished setting the sort order for a service, click **Save** in the toolbar.
4. To see how the sort order affects the services tree, click **Refresh** in **Service Navigation** portlet.

Note: You can also set the sort order for services in the Service Editor. However, you can only use numeric ordering. Alphabetic ordering is not supported for the Service Editor.

Note: The resources displayed in the *Service Viewer* and *Service Editor* are not sorted. They are positioned, so as to optimize the layout of the view in the space provided.

Adding user and group permissions

On the Security tab, you can assign access privileges to a given service in TBSM. To control access for a service, you assign roles to users and groups.

Chapter 19. Service Details portlet

This topic describes the Service Details portlet and the portlet toolbar buttons.

Note: This feature will not be part of subsequent TBSM releases.

The **Service Details** portlet shows information about events, rules, and SLAs that affect a given service-dependency model. The **Service Details** portlet is displayed below the Service Editor by default. It has the following tabs:

- **Cumulative SLAs**
- **Events**
- **Rules**

  Note: Clicking a **Service Details** tab does not update the data displayed in the tab. Click the service or chart as described in the following sections for each tab.

Table 46 on page 135 describes the **Service Details** portlet toolbar buttons for the **SLA**, **Events**, and **Rules** tabs.

Table 46. Service Details portlet toolbar buttons

Button	Click this button to...
	Refresh the data in this tab.
	Freeze/update this portlet.
	Select/deselect all rows.
	Access help for the tab.

Cumulative SLAs tab

This topic describes the Cumulative SLAs tab in the Service Details portlet.

The **Cumulative SLAs** tab shows the SLA status for services that have SLA cumulative-time settings defined. When a service is `marginal` or `critical`, you can view the details of the time and duration calculations of the outage.

To display the data in the **Cumulative SLAs** tab, click a service with SLA settings in the Service Tree or the Service Navigation portlet. The data will displayed in the Service Viewer and Editor.

Note: Only SLA cumulative, time-based violations are shown on the Cumulative SLAs tab.

Events tab

This topic describes the Events tab in the Service Details portlet.

The **Events** tab shows ObjectServer events that affect the selected service in table format. There are three ways to display events in the **Events** tab.

- Right-click a service in the Service Viewer or Service Editor **View** tab and select **Show** > **Service Affecting Events (Table)** or select **Show** > **Root Cause Events**.
- Double-click the service image.

- Click a service in the Service Tree or **Service Navigation** portlet. The service details updates.

If you double-click the event in the **Service Details** portlet, the *Alert Status* window opens. It shows all the field values for the event.

Rules tab

This topic describes the Rules tab in the Service Details portlet.

The **Rules** tab shows the status of the event or dependency rules that caused a service instance's status to change to `marginal` or `critical`.

To display the rule status for a service, right-click a service in the Service Tree, Service Navigation portlet, Service Viewer, or Service Editor and select **Show** > **Show Rule Status** from the menu that is displayed.

Displaying events

This topic describes how to display service-affecting events, display root cause events, and display active events list.

About this task

You can display information about the events that affect a given service in a table or in an Active Event List (AEL).

Note: The Service Viewer will be removed in TBSM 6.1.1 Fix Pack 6.

Displaying service-affecting events table

About this task

You can display data on ObjectServer events that affect a given service in the **Events** tab of the **Service Details** portlet in the following ways:

Procedure

1. From the Service Tree, the Service Navigation portlet, Service Viewer, or the Service Editor **View Service** tab, right-click the service where you want to see events.
2. Select **Show** > **Service Affecting Events (Table)** from the menu that appears.

   The **Events** tab in the **Service Details** portlet shows a table of events that affect the service. Only the events related to the service are displayed. The columns in this table are Netcool/OMNIbus event list fields.

Displaying root cause events

About this task

If a service has an outage, you can display only the root-cause events for the outage in the **Service Details** window as follows:

Procedure

1. Right-click the service instance in the Service Tree, the Service Navigation portlet, Service Viewer, or the Service Editor to display the options menu.
2. Select **Show** > **Show Root Cause Events** from the menu that appears.

   An event table is displayed in the **Service Details** window. However, this table will only display the data for the events that caused the outage.

Display active events list

About this task

You can also display service-affecting events in an Active Event List (AEL). You must have the Java plug-in installed on your client machine to display an AEL. To open an AEL for a service:

Procedure

1. Right-click the service instance in the From the Service Tree, the Service Navigation portlet, Service Viewer, or the Service Editor to display the options menu.
2. Select **Show** > **Show Service Affecting Events (AEL)** from the menu that appears.

   A portlet page is opened containing an AEL portlet shown in context of the service. If an AEL page is already opened, it will be reused.

   Only the events related to the service are displayed. The AEL contains many of the same features as the Netcool/OMNIbus event list.

   Note: If you want users to be able to modify ObjectServer events, those users must exist in the ObjectServer user store and the appropriate roles must be assigned.

   Note: By default, the AEL tools in the Tools menu do not function in TBSM. You need to edit the appropriate tool configuration file to point to the PERL home directory on the TBSM host. The default location of PERL may not be correct for your system. For more information about enabling AEL tools, see the *IBM Tivoli Business Service Manager: Administrator's Guide*.

Show rule status

This topic describes how to show rule status in Service Details.

About this task

You can display a summary of the TBSM rule status for a service in the **Service Details** window. To show the rule status for a service, complete the following steps:

Procedure

1. Right-click the service in the Service Tree, the Service Navigation portlet, Service Viewer, or the Service Editor to display the options menu.
2. Select **Show** > **Show Rules Status** from the menu that appears.

   A list of service-affecting rules is displayed in the **Rules** tab of the **Service Details** window. The rule information displayed here is the same as the service status events that TBSM sends to the ObjectServer when the service's status changes.

   You can show the details of each event by double-clicking on an event in the list.

Chapter 20. Maintenance schedules

This section describes how to create, edit, and remove maintenance schedules.

Maintenance schedules overview

This topic describes maintenance schedules.

TBSM includes a maintenance feature that allows you to schedule maintenance periods for your services.

During maintenance, TBSM displays the following maintenance information:

- The service status in the **Services** tree is in maintenance (blue).
- The service instance in the **View Service** tab turns blue and has a construction-cone icon to indicate the service is undergoing maintenance.

If an event comes in that clears the service, the service instance and status in the **Services** tree turns green.

To schedule a maintenance period for a service, you associate a maintenance schedule with the service. You can assign one or more time windows to a single maintenance schedule. There are two types of time windows: a recurring schedule and an absolute, one-time only time period. A combination of each type can be assigned to a single maintenance schedule.

You can enter maintenance schedules in two ways:

- From the **Edit Service** tab, you can configure maintenance schedules for a service. You can configure both recurring schedules and one-time only (Absolute) time windows and assign them to a maintenance schedule.
- From both the **View Service** tab in the Service Editor and the **Services** tree in the Service Navigation portlet, you can use the right-click menu options to quickly add or remove maintenance schedules. Using the quick option, you can schedule maintenance times in the following ways:
  - For the selected service only
  - For the selected service and all its descendents
  - For all the children but not for the selected service

Note: There is no way to delete a maintenance schedule from the schedule selection list in the TBSM console. If you need to delete a maintenance schedule, see the *TBSM Administrator's Guide*.

Configuring maintenance schedules

This task describes how to configure maintenance schedules in the **Edit Service** tab.

About this task

Maintenance schedules contain one or more maintenance time windows. The time windows are the periods when the service is in maintenance mode. There are two types of time windows:

- Recurring time windows are maintenance periods that occur on a weekly basis. For example, you can schedule a maintenance period for every Monday between 1 am and 3 am.
- Absolute time windows are maintenance periods with a set date and time. They do not repeat on a regular basis. For example, you can schedule a maintenance period for March 30, 2007 between 2 am and 5 am.

Note: If a schedule has multiple time windows, these time windows cannot overlap.

Use the options on the **Edit Service** tab to add and edit maintenance schedules for a given service.

Note: When you create a service instance, you must save it before you add a maintenance schedule for it. If you do not save the service instance first, all other settings you have made for the instance will disappear.

Table 47 on page 140 describes the **Edit Service** tab maintenance elements.

Table 47. Edit Service tab: maintenance elements

Window element	Description
Maintenance Schedule list	If there are maintenance schedules configured on your system, select the schedule you want from this list and click the **Save** button in the toolbar.
Edit button	If you want to change the settings for a schedule, select a schedule from the **Maintenance Schedule** list and click the **Edit** button. The **Maintenance** window opens in a new browser window.
New button	If you want to create a maintenance schedule, click the **New** button. The **Maintenance window** opens in a new browser window.

To edit or create a maintenance schedule for a service, complete the following steps:

Procedure

1. In the Service Navigation portlet drop-down menu, select **Services**. Click the service you want.
2. To edit an existing maintenance schedule, select a schedule from the **Maintenance Schedule** drop-down list and click the **Edit** button.

   To create a schedule, click the **New** button.

   The **Maintenance** window opens in a new browser window.
3. Enter the maintenance schedule data using Table 48 on page 140 as your guide.

Table 48. Maintenance window elements

Window element	Description
Schedule Name field	The name of the schedule you are creating or editing. If you click the **New** button, this field is blank.
Select All button 	Click the **Select All** button to select all the time windows for deletion.
Deselect All button 	Click the **Deselect All** button to deselect all the time windows selected for deletion.
Delete button 	Click this button to delete the selected time windows.
Recurring Time Window list and Add button 	This field lists all the recurring-time windows configured on your system. If you want to add an existing time window to a service, select a time window and click the **Add** button. A new row for the time window appears in the time window table.

Table 48. Maintenance window elements (continued)	
Window element	**Description**
Create New Recurring Time Window button 	Click this button to create a recurring-time window. The **New Recurring Time Window** window opens in a new browser window.
Absolute Time Window list and Add button 	This field lists all the absolute-time windows configured on your system. If you want to add an existing time window to a service, select a time window and click the **Add** button. A new row for the time window appears in the time window table.
Create New Absolute Time Window button 	Click this button to create an absolute-time window. The **New Absolute Time Window** window opens in a new browser window.
Select column	Select the check boxes in this column to mark time windows for deletion.
Time Window column	This column shows a summary for each time window.
Type column	This column shows whether the time window is `Absolute` or `Recurring`.

4. When you have finished configuring the schedule, click **OK**.

   If this schedule is new, it is listed in the Maintenance Schedule drop-down list in the **Edit Service** tab.

Configuring recurring time windows

This task describes how to configure recurring time windows.

About this task

To create a recurring time window:

Procedure

1. Click the **New** button for recurring-time windows.

   The **New Recurring Time Window** window opens in a new browser window.
2. Select the day of the week for the time window from the **Start Day** and **End Day** drop-down lists. The maintenance period will begin and end on these days every week.
3. Select the time of day for the time window from the **Start Time** and **End Time** fields. The maintenance period will occur between these times on the scheduled day(s) of the week.
4. Click **OK**.

   A new row for the time window appears in the time window table.

Configuring absolute time windows

This task describes how to configure absolute time windows.

About this task

To create an absolute time window:

Procedure

1. Click the **New** button for absolute-time windows.

   The **New Absolute Time Window** window opens in a new browser window.

2. Click the **Start Date** button to open a calendar. Click the start date.

   The date is displayed in the **Start Date** field.
3. Select the time of day for the time window from the **Start Time** field. The maintenance period will occur between this time and the end time on the scheduled day.
4. Click the **End Date** button to open a calendar. Click the end date.

   The date is displayed in the **End Date** field.
5. Select the time of day for the time window from the **End Time** field. The maintenance period will occur between this time and the start time on the scheduled day.
6. Click **OK**.

   A new row for the time window displays in the time window table.

   Time and date hints

   When you enter an invalid time or date, this hint appears: `The value entered is not valid`.

   The time and date fields will be formatted based on the format defined for the locale chosen for the browser. A new time or date can be selected from the menu shown when the entry field is clicked. If a custom time or date is required, select a value from the window close to the wanted value, and edit the entry field manually using the completed format as a guide.

Maintenance right-click tools overview

This topic describes right-click tools for maintenance.

You can quickly add and remove maintenance schedules for services using the right-click menus. You access these menus from either the Service Navigation portlet or the Service Editor by right-clicking the service that you want for maintenance schedules updates. When you add or remove maintenance schedule with the right-click tool, you can choose the scope of the schedule as:

- the selected service only
- for this service and all dependent services
- for only the dependant services, but not the selected service

Adding maintenance schedules quickly

This topic describes how to add maintenance schedules.

About this task

To add or edit a maintenance schedule with a right-click menu option, complete the following steps:

Procedure

1. Open the Service Administration page.
2. Navigate to the service you want, either in **Services** in the Service Navigation portlet drop-menu, or the **View Service** tab in the Service Viewer.
3. Right-click the service you want and select **Maintenance window tools** from the list.
4. Select **Schedule Maintenance** to open the Schedule Maintenance window.
5. In the window that opens, click **OK** to accept the defaults, or complete the following steps:
6. Select the scope:
   - Select to add the schedule for this service only (the default)
   - Select to add the schedule to this service and all its children

- Select to add the schedule to all the children but not to the service

7. Enter the duration.
   - You can accept the default of **Now** radio button, for one hour (the default), or change the duration.
   - You can select the **Schedule** radio button. After selecting this option, you can select an existing schedule, select an existing schedule and edit it, or add a new schedule. Selecting any of these options opens the Schedule Maintenance windows that are accessible from the **Edit Service** tab in the Service Viewer.
8. Click **OK** to save the schedule and close the window.

Removing maintenance schedules quickly

This topic describes how to remove maintenance schedules.

Before you begin

To remove a maintenance schedule from a service using the right-click option, complete the following steps:

Procedure

1. Open the Service Administration page.
2. Navigate to the service you want, either in **Services** in the Service Navigation portlet drop-menu, or **Services** in the Service Administration portlet drop-down menu.
3. Right-click the service.
4. Select **Maintenance window tools** from the list.
5. Select **Remove from Maintenance** from the list.
6. Select the scope:
   - Remove the schedule for the selected service only (the default)
   - Remove the schedule for the selected service and all its dependents
   - Remove the schedule for the service's dependents but not for the selected service
7. Click **OK** to remove the schedule.

Chapter 21. Granting user and group permissions to templates and services

This section describes how to add user privileges at the service and service template levels.

Assigning privileges overview

You assign access privileges by assigning roles to users and groups. You can grant access privileges in the following ways:

- Globally
- Per service
- Per template

For examples on how to assign user and groups roles in IBM Tivoli Business Service Manager (TBSM), see the *TBSM Administrator's Guide*.

Assigning roles globally

This describes global user roles.

When you assign a role globally to a user or group, that role is effective for all the associated TBSM objects. For example, if you assign the `tbsmEditService` role to a user, that user can edit any service instance objects currently defined in TBSM. Similarly, if you assign the `tbsmEditTemplate` role to a user, that user can edit any template object. You assign a role globally to users and groups using the Dashboard Application Services Hub console, as described in the *TBSM Administrator's Guide*.

To manage user roles, from the left navigation pane, click **Settings**. For more information, see the console help.

Assigning roles per service

This describes how to assign roles per service.

About this task

When you assign a role per service, that role is effective only for the service for which it is assigned. For example, if you assign the `tbsmEditService` role to a user in association with a particular service, that user is only given permission to edit that service. You assign roles per service when you edit services using the TBSM console.

To assign roles per service:

Procedure

1. In the left navigation pane, expand **BSM Menu**, and click **Service Administration** or **Service Configuration**. The **Service Administration** or **Service Configuration** page opens on the right.
2. In the **Service Navigation** portlet, select **Services** from the drop-down menu, click the service that you want to assign roles to.
3. Open the **Edit Service** tab if it is not already open.
4. Click the **Security** tab. The **Security** tab opens.

5. Select an item from the **Users/Groups** list and then use the **Privileges** lists to add or remove roles from the specified user or group.

   Table 49 on page 146 describes the privilege levels:

Table 49. Available user privileges	
Privilege	**Description**
tbsmServiceAdmin	Users can add and delete services as well as view and edit them.
tbsmEditService	Users can edit existing services as well as view them.
tbsmViewService	Users can view service only.

Assigning roles to templates

This describes how to assign roles to templates.

About this task

When you assign a role per template, that role is effective only for the template for which it is assigned. For example, if you assign the `tbsmEditTemplate` role to a user in association with a particular template, that user is only given permission to edit that template. You assign roles per template when you edit templates using the TBSM console.

To assign roles per template:

Procedure

1. In the left navigation pane, expand **BSM Menu**, and click **Service Administration** or **Service Configuration**. The **Service Administration** or **Service Configuration** page opens on the right.
2. Select **Templates** from the drop-down list in the **Service Navigation** portlet, and click the template that you want to add roles to.
3. From the Service Editor, **Edit Templates** tab, click the **Security** tab. The **Security** tab opens.
4. Select an item from the **Users/Groups** list and then use the **Privileges** lists to add or remove roles from the specified user or group.

Results

Table 50 on page 146 describes the privilege levels.

Table 50. Available user privileges	
Privilege	**Description**
tbsmTemplateAdmin	Users can add, edit, delete, and view the template.
tbsmServiceAdmin	Users can add, edit, delete, and view specific services that are tagged with the template.
tbsmEditTemplate	Users can edit the template as well as view it.
tbsmViewTemplate	Users can view the template only.
tbsmEditService	Users can edit specific services that are tagged with the template.
tbsmViewService	Users can view the service only.

Accessing TBSM-specific roles

This topic describes how to access TBSM-specific roles.

About this task

The Dashboard Application Service Hub **Settings** > **Roles** page provides a list of TBSM-specific roles.

Use the following procedure to access the **Security** page:

Procedure

1. From the left navigation pane, expand **Settings**.
2. Select **Roles** from the list.

   The TBSM-specific roles are prefixed with `tbsm`. Check all three pages to see them all.

   Table 51 on page 147 describes the TBSM-specific roles:

Table 51. TBSM user roles

Role	Description
tbsmAdminUser	A user assigned to this role can access and modify all the TBSM features except AEL event modification. See the *IBM Tivoli Business Service Manager: Administration Guide* for more information.
tbsmSLAChartViewVisible	This role is 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.
tbsmAVSaveCanvasLayoutForGroup	Create, edit, and delete custom canvases for a group.
tbsmAVSaveCanvasLayoutForUser	Create, edit, and delete custom canvases for a user.
tbsmTemplateAdmin	Add or edit any template
tbsmServiceAdmin	Add or edit service instances
tbsmCreateTemplate	Add service templates
tbsmEditTemplate	Edit service templates
tbsmViewTemplate	View service instances
tbsmCreateService	Create service instances
tbsmEditService	Edit service instances
tbsmViewService	View service instances
tbsmDataSourceAdmin	Add, edit, or delete data source configurations.
tbsmCreateDataSource	Add data source configurations.
tbsmEditDataSource	Edit data source configurations.
tbsmViewDataSource	View data source configurations.
tbsmDataFetcherAdmin	Add edit, or delete data fetcher configurations.
tbsmCreateDataFetcher	Add data fetcher configurations.
tbsmEditDataFetcher	Edit data fetcher configurations.

Table 51. TBSM user roles (continued)	
Role	**Description**
tbsmViewDataFetcher	View data fetcher configurations.
tbsmChartAdmin	Create, edit, view, and delete TBSM charts.
tbsmCreateChart	Create, view, and edit TBSM charts.
tbsmEditChart	Create view, and edit TBSM charts.
tbsmViewChart	View TBSM charts.
tbsmReadOnlyUser	This role is required to make readonly users operational on TBSM and must be assigned to all readonly users. This role is assigned by default to the tbsmReadOnly group; users assigned to that group automatically have this role.
tbsmViewDefinitionAdmin	Create, edit, view, and delete TBSM view definitions.

Chapter 22. Editing policies

This section explains how to work with the IBM Tivoli Business Service Manager (TBSM) Policy editor.

About the Policy editor

This topic describes the Policy editor.

Policies consist of a series of function calls that manipulate events and data from your supported data sources. A policy is a script that contains a set of instructions to automate alert management tasks, for example, defining the conditions for sending an e-mail to an administrator, or sending instructions to the ObjectServer to clear an event. For example, TBSM policies define conditions that determine whether the status of a service is good, marginal, or bad.

Pre-defined policies overview

This topic describes pre-defined policies.

TBSM lets you customize the functions for the policies used for numeric aggregation rules and rules used to create services automatically. You can modify these policies to better meet the needs of your service environment. You should be familiar with Netcool/Impact policies before you attempt to customize these policies. For more information about Netcool/Impact policies, see the Netcool/Impact *User*, *Policy Reference*, and *Solutions* guides.

You can edit the following policies from within the TBSM console:

- Numeric-Aggregation Policy
- Numerical Formula Rule Policy
- Auto-Population Rule Policies
- ESDA Model Rule Policy

For more information about these policies, see the *IBM Tivoli Business Service Manager: Customization Guide*.

Opening the Policy editor

This topic describes how to open the policy editor.

You can access the Policy editor by clicking the **Edit Policy** button in the following windows:

- *Numerical Aggregation Rule*
- *Numerical Formula Rule*
- *ESDA Model Rule*
- *Custom Auto-population Rule Configuration*
  - Click **Custom Configuration** in the **New Auto-population Rule** window.
  - Click one of the **Use Policy** check boxes in the **Custom Auto-population Rule Configuration** window to make the **Edit Policy** button available.

For more information about accessing and customizing the Policy editor, see the *IBM Tivoli Business Service Manager: Customization Guide*.

Policy editor toolbar controls

An overview of the policy editor toolbar controls.

Table 52. Policy Editor toolbar options

Icon	Description
	The **Save** icon saves the current policy. Use the **Save with comments** option to save your policy with comments. To save a policy with a different file name click **Save as...**. **Remember:** If you use UTF-8 characters in the policy name, check that the locale on the Impact Server where the policy is saved is set to the UTF-8 character encoding.
	Restore your work to its state before your last action, for example, add text, move or, delete. **Undo** works for one-level only.
	Restore your work to its state before you selected the **Undo** action. **Redo** works for one-level only.
	Cut highlighted text. In some instances, due to browser limitations, the **Cut** icon cannot be activated. Use the keyboard short cut `Ctrl + x` instead.
	Copy highlighted text. In some instances due to browser limitations, the **Copy** icon cannot be activated. Use the keyboard short cut `Ctrl + c` instead.
	Use this icon to paste cut, or copied text to a new location. In some instances due to browser limitations, the **Paste** icon cannot be activated. Use the keyboard short cut `Ctrl + v` instead. To copy and paste rich text formatted content, for example from a web page or document file: 1. Paste the content into a plain text editor first to remove the rich text formatting. 2. Copy the content from the plain text editor into the policy editor.
	Use this icon to find and replace text in a policy. Search for a text string. Type the text that you want to find, choose if you want to run a case-sensitive search, and choose the direction of the search. Search for text and replace it with a text you specify. Type the text that you want to search for. Type the replacement text. Choose if you want to run a case-sensitive search, and choose the direction of the search.
	Click the **Go To** icon to show a **Go To Line** field in the policy editor. Type the number of the line you want the cursor to go to. Click **Go**.
	Insert a selected function, an action function, or a parser function, in your policy. Add additional parameters for the function if required. The toolbar selection lists provide you with a set of functions to use in your policy.

Table 52. Policy Editor toolbar options (continued)	
Icon	**Description**
	Access a list of data types. The **Data Type Browser** icon simplifies policy development by showing available data types and details including field name and type information. You do not have to open the data type viewer to get the data type information.
	The **Check Syntax** icon checks the policy for syntax errors. If there are errors, the error message locates the error by the line number. If there are no errors, a message to that effect is shown.
	Optimize the policy. .
	Click the **Run Policy** icon to start the policy. After removing all syntax errors, you can run the policy to ensure that it produces the result you wanted. To run your policy with additional parameters, use the **Run with Parameters** option. You can use this option after you configure user parameters for your policy.
	Use this icon to set the user parameters for the policy.
	Click the **View Version History** icon to view the history of changes made to policies, and compare different versions of policies. **Important:** The **View Version History** icon is disabled for new and drafted policies and it becomes active after the policy is committed to server. This option is supported only with the embedded SVN version control system.
	Click this icon to view the policy logs in the log viewer.
	The **Graphic View** is not available for JavaScript policies.
	Click this icon to manually enable or disable the syntax highlighter.

Using the Undo, Redo, Cut, Copy, and Paste options

This topic describes the Undo, Redo, Cut, Copy, and Paste options.

About this task

This section explains how to use the **Undo**, **Redo**, **Cut**, **Copy**, and **Paste** options.

Undo

To restore your work to the state it was in before your last action, for example, adding text, deleting text, replacing text, and so on, click the **Undo** button.

This function is a one-level Undo only.

Redo

To restore your work to the state it was in before an Undo, for example, adding text, deleting text, replacing text, and so on, click the **Redo** button.

This function is a one-level Redo only.

Cut

To cut text that you want to either delete or move to a new location, highlight the text and click the **Cut** button.

Copy

To copy text so that you can place it in an additional location without retyping it, highlight the text and click the **Copy** button.

Paste

To paste cut or copied text to a new location, place your cursor where you want it to start and click the **Paste** button.

Using the Find and Replace option

This topic describes how to use the Find and Replace option.

About this task

Select the **Find and Replace** button to open the **Find and Replace** window.

Table 53 on page 152 explains the window elements.

Table 53. Policy Script Editor Find and Replace window

Window Element	Element Type	Description
Find What	Required text field	Enter the text that you want to find or replace.
Replace With	Required text field	Enter the new text.
Match case	Check box	Select to consider case.
Direction: Backwards	Check box	By default, the direction of the search is forward - beginning to end. You can check **Backwards** to change the direction of the search - end to beginning.
Replace	Button	Click to replace the text at the first occurrence.
Replace All	Button	Click to replace all occurrences of the text string.
Find	Button	Click to highlight the requested text.
Close	Button	Click to close the window.

Using the Goto Line option

This topic describes how to use the Goto Line option.

About this task

This option moves your cursor immediately to a specified line in a policy.

Procedure

1. Select the **Goto Line** button to access the **Goto Line** window:

2. Click the **Line Number Spin** arrows to find the appropriate line number, or type in the number that you want.
3. Click **Go**.

   To close the window without searching for a line number, click the **Close** button.

Checking the policy syntax

This topic describes how to check the policy syntax.

About this task

While you are creating your policy, you can check to ensure that the syntax is correct.

When you select the **Check Syntax** button, the system displays a list of errors at the bottom of the *Policy* editor. If there are no errors in the policy, the following message is displayed:

```
Syntax check successful. No error found.
```

If the checker finds errors, you will see a table at the bottom of the editor that shows the error type, the line where it occurred, and a message.

The **Type** column contains an error indicator, either Warning or Error.

The **Line** column contains the line number where the error occurred. To find the error, click the line number. The editor scrolls to that line in the script.

Optimize your policy

This section describes how to optimize a policy.

About this task

After you have written, named, and saved your policy, you can check to see whether there is a way to improve it by selecting the **Optimize It** button

To optimize your policy, click the **Optimize It** button.

Results

A message box opens either with suggestions or a message that no improvements are necessary.

Adding functions to policy

Use this procedure to add a function to a policy.

Procedure

1. Click the **Insert function** icon and select one of the functions.
2. Enter the required parameters in the new function configuration window.

   Note: When entering a string, check that all string literals are enclosed in quotation marks ("*string* "), to distinguish them from variable names, which do not take quotation marks.

List and overview of functions

A list of all functions with a short overview.

Table 54. List of functions

Name	Type	Description
Activate	Policy	The Activate function runs another policy.

Table 54. List of functions (continued)

Name	Type	Description
ActivateHibernation	Policy	The `ActivateHibernation` function continues running a policy that was previously put to sleep using the `Hibernate` function. You must also run the `RemoveHibernation` function to remove the policy from the hibernation queue and to free up memory resources.
AddDataItem	Database, Internal	The AddDataItem function adds a data item to a data type.
BatchDelete	Database	The BatchDelete function deletes a set of data items from a data type.
BatchUpdate	Database	The BatchUpdate function updates field values in a set of data items in a data type.
BeginTransaction	Database	The BeginTransaction is a local transactions function that is used in SQL operations.
CallDBFunction	Database	CallDBFunction calls an SQL database function.
CallStoredProcedure	Database	The CallStoredProcedure function calls a database stored procedure.
ClassOf	Context	The `ClassOf` function returns the data type of a variable.
CommandResponse	Systems	Use the `CommandResponse` function to run interactive and non-interactive programs on both local and remote systems.
CommitChanges	Database	Used only in connection with GetByFilter, and GetByKey functions to force updates in a database.
CommitTransaction	Database	The CommitTransaction function is a local transactions function that is used in SQL operations.
CurrentContext	Context	The `CurrentContext` function returns the current policy context.
Decrypt	String	The Decrypt function decrypts a string that has been previously encrypted using Encrypt or the nci_crypt tool.
DeleteDataItem	Database, Internal	The DeleteDataItem function deletes a single data item from a data type.

Table 54. List of functions (continued)

Name	Type	Description
Deploy	Miscellaneous	The Deploy function copies data sources, data types, policies, and services between server clusters.
DirectSQL	Database	The `DirectSQL` function runs an SQL operation against the specified database and returns any resulting rows to the policy as data items.
DataItems	Keys	Provided for backward-compatibility only.
Distinct	Array	The `Distinct` function returns an array of distinct elements from another array.
Encrypt	String	The Encrypt function encrypts a string.
Escape	String	This function escapes special characters in an input string in a policy.
Eval	Context	The `Eval` function evaluates an expression using the given context.
EvalArray	Array, Context	The EvalArray function evaluates an expression using the given array.
Exit	Policy	You use the `Exit` function to stop a function anywhere in a policy or to exit a policy.
Extract	String	The Extract function extracts a word from a string.
FindFilters	Database	Provided for backward-compatibility only.
Float	Numeric	The Float function converts an integer, string, or Boolean expression to a floating point number.
FormatDuration	Time	The FormatDuration function converts a duration in seconds into a formatted date/time string.
GetByFilter	Database, Internal, ITNM, LDAP, XML	The GetByFilter function retrieves data items from a data type using a filter as the query condition.
GetByKey	Database, Internal, LDAP	The `GetByKey` function retrieves data items from a data type using a key expression as the query condition.
GetByLinks	Database, Internal, XML	The `GetByLinks` function retrieves data items in target data types that are linked to one or more source data items.

Table 54. List of functions (continued)

Name	Type	Description
GetByXPath	XML	The `GetByXPath` function provides a way to parse an XML string or get an XML string through a URL specified as parameter.
GetClusterName	Variables	You use the GetClusterName function inside a policy to identify which cluster is running the policy.
GetDate	Time	The GetDate function returns the date/time as the number of seconds expired since the start of the UNIX epoch.
GetFieldValue	Java	Use this function to get the value of static, or non-static fields. For non-static fields, use the variable *FieldName* for a Java class or *TargetObject* for a Java object. For a static Java class field, use the variable *ClassName*.
GetGlobalVar	Variables	This function retrieves the global value saved by previous `SetGlobalVar` calls.
GetHTTP	REST	You can use the `GetHTTP` function to retrieve any HTTP URL or to post content to a web page.
GetHibernatingPolicies	Policy	The `GetHibernatingPolicies` function retrieves data items from the Hibernation data type by performing a search of action key values.
GetScheduleMember	Time	The GetScheduleMember function retrieves schedule members associated with a particular time range group and time.
GetServerName	Variables	You use the GetServerName function inside a policy to identify which server is running the policy.
GetServerVar	Variables	You use this function to retrieve the global value saved by previous SetServerVar.
Hibernate	Policy	The Hibernate function causes a policy to hibernate.
Illegal	String	If the input in the policy has malicious content, the Illegal function throws an exception in a policy.
Int	Numeric	The Int function converts a float, string, or Boolean expression to an integer.
JRExecAction	Systems	The JRExecAction function executes an external command using the JRExec server.

Table 54. List of functions (continued)

Name	Type	Description
JavaCall	Java	You use this function to call the method `MethodName` in the Java object `TargetObject` with parameters, or, to call the static method `MethodName` in the Java class `ClassName` with parameters.
Keys	Context	The `Keys` function returns an array of strings that contain the field names of the given data item.
Length	Array, String	The Length function returns the number of elements or fields in an array or the number of characters in a string
Load	JavaScript	You use this function to load a JavaScript library into your JavaScript policy.
LocalTime	Time	The LocalTime function returns the number of seconds since the beginning of the UNIX epoch as a formatted date/time string.
Log	Policy	The Log function prints a message to the policy log.
Merge	Context	The `Merge` function merges two contexts or event containers by adding the member variables of the source context or event container to the those of the target.
NewEvent	Context, Database	The NewEvent function creates a new event container.
NewJavaObject	Java	The `NewJavaObject` function is used to call the constructor for a Java class.
NewObject	Context	The NewObject function creates a new context.
ParseDate	Time	The ParseDate function converts a formatted date/time string to the time in seconds since the beginning of the UNIX epoch. 1st January 1970 00:00:00 (UTC).
PassToTBSM	TBSM	Use the `PasstoTBSM` function to send event information from Netcool/Impact to TBSM.
Random	Numeric	The Random function returns a random integer between zero and the given upper bound.
RDFModel	String	You can use the RDFModel function to create an RDF model without any runtime parameters.

Table 54. List of functions (continued)

Name	Type	Description
RDFModelToString	String	You can use the RDFModelToString function to export an RDF model to a string in a particular language.
RDFModelUpdateNS	String	You can use the RDFModelUpdateNS function to insert, update, or remove a namespace from an RDF model.
RDFNodeIsResource	RDFNode	You can use the RDFNodeIsResource function to help other functions read and parse objects that are also an RDF resource.
RDFNodeIsAnon	RDFNode	You can use the RDFNodeIsAnon function to assist in reading and parsing an RDF.
RDParse	String	You can use the RDFParse function to help other functions read and parse an RDF object
RDFRegister	String	You can use the RDFRegister function to help you to register service providers or OSLC resources with the registry server.
RDFUnRegisgter	String	To remove the registration record of a service provider or resource from the registry server, use the RDFUnRegister function to supply the location of the registration record, the Registry Services server username and password, and the registration record that you want to remove.
RDFSelect	String	You can use the RDFSelect function to assist in reading and parsing an RDF. To retrieve statements based on an RDF model, you call the RDFSelect function and pass the RDF model that is created by the RDFParse function. You can filter based on subject, predicate, and object.
RDFStatement	String, Boolean	You can use the RDFStatement function to create and add statements to an RDF model.
RemoteTBSMShell	TBSM	A stand-alone implementation of Netcool/Impact can run `RADShell` commands from a policy in Tivoli Business Service Manager.
RExtract	String	The RExtract function uses regular expressions to extract a substring from a string.
RExtractAll	String	The RExtractAll function uses regular expression matching to extract multiple substrings from a string.
ReceiveJMSMessage	JMS	The ReceiveJMSMessage function retrieves a message from the specified Java Message Service (JMS) destination.

Table 54. List of functions (continued)

Name	Type	Description
RemoveHibernation	Policy	The `RemoveHibernation` function deletes a data item from the Hibernation data type and removes it from the hibernation queue.
Replace	String	The Replace function uses regular expressions to replace a substring of a given string.
ReturnEvent	Policy	The ReturnEvent function inserts, updates, or deletes an event from an event source.
RollbackTransaction	Database	The RollbackTransaction function rolls back any changes done by an SQL operation.
SendEmail	Notifications	The SendEmail function sends an email that uses the email sender service.
SendInstantMessage	Notifications	The SendInstantMessage function sends an instant message using the Jabber service.
SendJMSMessage	JMS	The SendJMSMessage function sends a message to the specified destination by using the Java Message Service (JMS) DSA.
SetFieldValue	Java	Use the SetFieldValue function to set the field variable in the Java class to some value.
SetGlobalVar	Variables	The SetGlobalVar function creates in a policy a global variable which can be accessed from any local functions, library functions, and exception handlers in a policy.
SetServerVar	Variables	The SetServerVar function creates a server-wide global variable in a policy.
SnmpGetNextAction	SNMP, Sytems	The `SnmpGetNextAction` function retrieves the next SNMP variables in the variable tree from the specified agent.
SnmpGetAction	SNMP, Systems	The SnmpGetAction function retrieves a set of SNMP variables from the specified agent
SnmpSetAction	SNMP	The SnmpSetAction function sets variable values on the specified SNMP agent.
SnmpTrapAction	SNMP	The SnmpTrapAction function sends a trap (for SNMP v1) or a notification (for SNMP v2) to an SNMP manager.
Split	String	The Split function returns an array of substrings from a string using the given delimiters.

Table 54. List of functions (continued)

Name	Type	Description
String	String	The `String` function converts an integer, float, or boolean expression to a string.
Strip	String	The Strip function strips all instances of the given substring from a string.
Substring	String	The Substring function returns a substring from a given string using index positions.
Synchronised	Policy	Use the Synchronized function to write thread-safe policies for use with a multi-threaded event processor using IPL or JavaScript.
ToLower	String	The ToLower function converts a string to lower case characters.
TBSMShell	TBSM	This topic describes the TBSMShell action function which lets you put RADshell commands in a policy. With the TBSMShell function, you can change the TBSM configuration in a policy.
ToUpper	String	The ToUpper function converts a string to upper case characters.
Trim	String	The Trim function trims leading and trailing white space from a string.
URLDecode	String, REST	The URLDecode function returns a URL encoded string to its original representation.
URLEncode	String, REST	The URLEncode function converts a string to a URL encoded format.
UpdateEventQueue	Database	The `UpdateEventQueue` function updates or deletes events in the event reader event queue.
WSDMGetResourceProperty **[Important: This feature is deprecated.]**	Web Services	The `WSDMGetResourceProperty` function retrieves the value of a management property that is associated with a Web Services Distributed Management (WSDM) managed resource.
WSDMInvoke **[Important: This feature is deprecated.]**	Web Services	The `WSDMInvoke` function sends a web services message to a Web Services Distributed Management (WSDM) managed resource.
WSDMUpdateResourceProperty **[Important: This feature is deprecated.]**	Web Services	The WSDMUpdatetResourceProperty function updates the value or values of a management property that is associated with a Web Services Distributed Management (WSDM) managed resource.

Table 54. List of functions (continued)		
Name	**Type**	**Description**
WSInvoke	Web Services	Provided for backward-compatibility only.
WSInvokeDL	Web Services	The `WSInvokeDL` function makes web services calls when a Web Services Description Language (WSDL) file is compiled with `nci_compilewsdl`, or when a web services DSA policy wizard is configured.
WSNewArray	Web Services	The WSNewArray function creates an array of complex data type objects or primitive values, as defined in the WSDL file for the web service.
WSNewEnum	Web Services	The WSNewEnum function returns an enumeration value to a target web service.
WSNewObject	Web Services	The WSNewObject function creates an object of a complex data type as defined in the WSDL file for the web service.
WSNewSubObject	Web Services	The WSNewSubObject function creates a child object that is part of its parent object and has a field or attribute name of ChildName.
WSSetDefaultPKGName	Web Services	The WSSetDefaultPKGName function sets the default package that is used by WSNewObject and WSNewArray.

For more details about each of these functions, see the *Policy Reference Guide*.

Triggering the policy

This describes how to trigger a policy.

About this task

To trigger a policy, complete the following steps:

Procedure

1. Click the **Policy Trigger** button to the open the **Policy Trigger** window. The fields you see in the window depend on whether you have specified any runtime parameters for the policy and what the runtime parameters were set to.
2. If you do not want to use the default values, then enter appropriate parameter values for the fields that are displayed.
3. Click **Execute** to trigger the policy.

Setting policy runtime parameters in the editor

Use this procedure to set the runtime parameters for your policy in the policy editor.

Procedure

1. In the policy editor toolbar, click the **Configure Runtime Parameters** icon to open the policy runtime parameter editor.
2. Click **New Runtime Parameter** to open the **Create a New Policy Runtime Parameter** window.

   Enter the information in the new runtime parameter configuration window. Required fields are marked with an asterisk (*).
3. To edit an existing runtime parameter, select the check box next to the parameter and select **edit** in in the corresponding cell of the **Edit** column.
4. Click **OK** to save the changes to the parameters and close the window.

Displaying the graphical view

This topic describes how to display the graphical view of the policy.

About this task

Click the **Graphical View** button to open a graphical view of the policy in a window.

Procedure

1. Remove the check mark in the **Show: Assignments** check box to remove assignments from the view.
2. Enter a check mark in the **Logging** check box to show logging details.

   Note: If you use JavaScript to write a policy, the graphical view option is not available.

Editing a policy

This topic describes how to edit a policy.

About this task

Use the following procedure to edit a policy:

Procedure

1. To open a policy for editing, click the **Edit Policy** button, which is accessible from any of the following TBSM windows:
   - *Numerical Aggregation Rule*
   - *Numerical Formula Rule*
   - *ESDA Model Rule*
   - *Custom Auto-population Rule Configuration*
     - Click **Custom Configuration** in the **New Auto-population Rule** window and then
     - Click one of the **Use Policy** check boxes in the **Custom Auto-population Rule Configuration** window to make the **Edit Policy** button available.

   You can edit the policy as required in the same way you created it: entering new text, new action and parser functions, checking the syntax and optimizing the policy.

Note: In general, do not use special characters in policy names.

2. When you are finished, click **Save**.

Saving a policy

You can save a policy that is open in the **Policy** editor by clicking the **Save** button in the *Policy* editor toolbar. You can also add comments when you save a policy by selecting the **Save with comments** option.

Chapter 23. Service level agreement configuration

This section explains how to configure and view the SLA data with the optional SLA features of IBM Tivoli Business Service Manager (TBSM).

SLAs overview

This topic describes service level agreements (SLA).

When you have created your service template hierarchy and added services, the basic service configuration is complete. Optionally, you can add service level agreement rules to your templates. This section describes how to apply service level agreement (SLA) rules to service templates with TBSM and how to display SLA data. SLA rules define when a given service violates the parameters of an SLA. Each template has its own set of service level agreements. For example, you can configure a rule that triggers an SLA violation whenever a service's up time is less than 99% of the time in a given week.

SLA settings tab

This topic describes the SLA settings tab in the Service Editor.

The **SLA** tab in the **Service Editor: Edit Template** tab lets you set the SLA rules for each service template. For each service template, you have three options:

- You can accept the default values for the standard SLA settings that have already been set for you.
- You can modify the settings for the standard level of service, including:
  - Cumulative, time-based violations and warnings (for the month, day, hour, and minute)
  - Duration-based violations and warnings
  - The number of violation incidents in a given time period
- You can also create multiple levels of service to replicate the various service level agreements you have with your customers. For example, you could create gold, silver, and bronze service levels.

Note: Only the SLA cumulative, time-based violations are shown on the Service Details: SLA tab.

SLA data in the dependency view

This topic describes SLA data in the dependency view and in service instances.

You can display detailed SLA data from the Service Editor dependency view. This SLA time display shows you the status of the SLA and lets you view the information about outages that have occurred during the SLA monitoring period. The SLA report only displays if an outage has occurred during the monitoring period.

Displaying the SLA tab

This topic describes the Cumulative SLAs tab in the Service Details portlet. This topic also describes how to display the SLA tab.

About this task

Note: This feature will not be part of subsequent TBSM releases.

The **Cumulative SLAs** tab in the **Service Details** portlet shows the SLA status for services that have SLA settings defined. When the service status is bad, you can view the details of the cumulative-duration calculations for the outage in the **Service Details** portlet.

To display the **Cumulative SLAs** tab in the **Service Details** portlet:

Procedure

1. From the **Services** in the **Service Navigation** portlet or the Service Tree, click the service you want to view.

   The dependency view for the service is displayed in the Service Editor.
2. Click a service with SLA settings.
3. Click the **Cumulative SLAs** tab in the Service Details portlet.
4. When the status refreshes, click the service again to see updated data in the **Cumulative SLAs** tab.

Results

The updated data is organized in the list on the tab. The table below describes each column of data.

Table 55. Time computations display

Column name	Description
Service Name	The name of the affected service.
Best Case %	Best case projections for the selected time period for the service.
Downtime	Calculation for how long the service has been down.
Time Left	Calculation for how much time is left before the SLA is in violation.
TWin (Time Window)	The time range selected for tracking the outage.
Penalty	The estimated cost of the outages in dollars.

6.2.0.3 Disable SLA tracking

This topic describes how to disable SLA tracking.

If any SLA level is enabled in a template (for example, Cumulative Duration), all service instances created using the template with Standard SLA will begin tracking Standard SLA.

To specify that no SLA tracking should occur for a specified service instance, the Service Level for the service can be set to None. To do this, set the property below to `true` in the `sla.props` file on the Data Server and on the DASH Server.

```
impact.sla.noslatracking=true
```

Chapter 24. Viewing service models

You view and monitor your service-dependency model in the IBM Tivoli Business Service Manager (TBSM) Service Editor. This section explains how to work with the Service Editor.

Note: The Java plug-in for web browsers relies on the cross platform plugin architecture NPAPI, which has been supported by all major web browsers for over a decade. However, Mozilla intends to remove support for most NPAPI plugins in Firefox by the end of 2016.

Oracle has announced that it is to depreciate the Java browser plug-in technology with the next release of the Java platform (JDK9). Because of these industry moves, TBSM is announcing the end of support for the Service Viewer (and custom canvases). These features will be removed from the product in Fix Pack 5. Customers are advised to replace custom canvas usage with IBM Dashboard Application Services Hub widgets, as soon as possible.

See link: http://www.ibm.com/support/knowledgecenter/SSSPFK_6.1.1.3/com.ibm.tivoli.itbsm.doc/adminguide/bsmu_jazz_c_overview.html?lang=en

Detailed information on the TBSM Datasets available for use with DASH widgets is available in the "Advanced Topics" section of the Tivoli Business Service Manager Wiki at https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli+Business+Service+Manager1/page/Advanced+Topics.

The “Topology” Dataset, available since in 6.1.1.0, provides similar visualization to the legacy Concentric views. The “Business Impact” Dataset, introduced in 6.1.1.4, provides similar visualizations to the legacy Business Impact All and Business Impact views and the “Map” Dataset, introduced in 6.1.1.4, provides similar visualization to the legacy GIS views.

Service Administration Overview

This topic describes how to use the Service Administration page to view services.

Services in the **Service Navigation** portlet lets you view your service-dependency models and monitor how ObjectServer events or other data affect your services.

Note: The Service Administration page will not be part of subsequent TBSM releases. Use the Service Configuration page to configure your services models.

To access the Service Administration page:

1. In the left navigation pane, click the **BSM Menu**.
2. Click **Service Administration**. The page loads on the right.

From these service views you can do the following:

- View your services, subservices, and service health based on simple color-coded indicators (red, yellow, green, and so on)
- Display related events in an Event List in the **Service Details** portlet or in an Active Event List (AEL)
- Display the status of the rules you configured in a services template
- Display a computation of optional SLA settings (see Chapter 23, “Service level agreement configuration,” on page 165)
- Display an event summary chart of any service.

Service Availability Overview

This topic describes how to use the Service Availability page to view services.

The Service Availability page lets you view your service models and related data. The **Service Tree** portlet lets you view your service-dependency models and monitor how ObjectServer events or other data affect your services.

The Service Tree and Urgent Services portlets can be customized. Portlet preferences can be accessed by clicking the **Edit Options** button on the portlet.

To access the Service Availability page:

1. In the left navigation pane, click the **BSM Menu**.
2. Click **Service Availability**. The page loads on the right.

From these service views you can do the following:

- View your services, subservices, and service health based on simple color-coded indicators (red, yellow, green, and so on)
- Display related events in an Event List in the **Service Details** portlet or in an Active Event List (AEL)
- Display the status of the rules you configured in a services template
- Display a computation of optional SLA settings (see Chapter 23, “Service level agreement configuration,” on page 165)
- Display of all the services in the urgent (red) state in a table form.

Note: There are no buttons on the Availability page, because no configuration is done on this page. Your permissions control which page you are able to view and use.

Service Configuration Overview

This topic describes how to use the Service Configuration page to view services.

Services in the **Service Navigation** portlet lets you view your service-dependency models and monitor how ObjectServer events or other data affect your services.

To access the Service Configuration page:

1. In the left navigation pane, click the **BSM Menu**.
2. Click **Service Configuration**. The page loads on the right.
3. **Templates** is open by default in the Service Navigation portlet. Select **Services** in the drop down list to view your services.

From these service views you can do the following:

- View your services, subservices, and service health based on simple color-coded indicators (red, yellow, green, and so on)
- Display the status of the rules you configured in a services template
- Display a computation of optional SLA settings (see Chapter 23, “Service level agreement configuration,” on page 165)

Creating a freeform custom page

This topic describes how to create a page using the freeform option.

Before you begin
To create a custom page, you need to be assigned the `administrator` role.

About this task

Using the freeform option gives you more freedom to arrange your widgets the way you want. Your options include all the TBSM default widgets, as well as some Netcool/Impact information:

Table 56. Lists some of the common types of TBSM information you can choose to display on a custom page.

TBSM
Service Tree
Service Navigation
Service Viewer
Service Editor
Service Details
Urgent Services
Charting
Widgets
Time Window Analyzer

You can also add pre-configured, customized widgets to a custom page. If you want a custom widget on your new page, you must do any customizing before you start creating a new page or the custom widget will not be accessible in the catalog.

Procedure

1. Log in to the DASH Console as a user with administrator roles.
2. In the task list, select **Settings > Pages**.

   The **Pages** tab is displayed.
3. In the **Pages** tab, click the **New Page** button.

   The **Page Settings** tab is displayed.
4. In the **Page name** field, enter a name for you custom page.
5. In the **Page Layout** field, select **Freeform**.
6. Click **OK**.

   A palette tab for your custom page is displayed.
7. Click **OK** to close the Information window.

   You can now start adding widgets to your custom page.
8. Use the **Search** field to display the available widget types that can be added to the page.
9. Drag a widget from the **Search Result** area to the grid below.

   You can now select a dataset to populate the widget.
10. Click on the drop-down button on the top right corner of the widget and select **Edit**.

    The **Select a Dataset** window is displayed.
11. Use the **Search** field to display the available datasets.
12. Scroll through and select a dataset.
13. Specify your required **Visualization Settings**.
14. Click on **OK**.

    You can add as many widgets as you wish. You can also move and resize the widgets on the grid to suit your requirements.

15. Click **Save**.

Services in the Service Navigation portlet

This topic describes Services in the Service Navigation portlet. This topic also describes how to display the dependency view.

The **Services** tree in the **Service Navigation** portlet contains trees of service-dependency models that have been configured for TBSM. The service instances you see depend on your user and group permissions.

To see the full tree, click the plus symbols to the left of the services. A list of services instances is displayed.

Displaying the dependency view

Clicking a service in the **Services** navigation tree loads a dependency view of the service and its subservices in the **View Service** tab of the Service Editor.

"Service Administration Overview" on page 167 shows the dependency view for `ABCBankeBanking` and its subservices.

Note: If you change rule or template names, you can cause issues with other features of TBSM. TBSM uses these names to reference the rule values for other rules, such as auto-population or ESDA rules and for custom displays such as service trees and view definitions.

If you want to change a rule or template name, change the name before you use the rule or template elsewhere in TBSM.

For example, when you create numerical and text-based rule, you need to create a custom service tree or view definition to display the output of the rule in the TBSM console. The trees and view definitions use the template and rule names to map the rule values to a display object. As a result, if you change the name of a rule or a template used in a custom display, the custom tree or view definition will no longer display the rule values properly.

Services elements

This topic describes the elements in Services in the Service Navigation portlet.

The color coded indicators to the right of each service name and subservice name change color as the status of the service changes. The status is automatically refreshed every 60 seconds. If the services have been added or removed from the service model, the service model is automatically updated with the changes.

Table 57 on page 170 describes the **Services** elements.

Table 57. Services elements

Services element	Element type	Description	Page
	**New** button	Click this button to create a service instance. When you click this button, a blank **Edit Service** tab opens in the Service Editor. Enter the parameters for the new service instance and click **Save**.	• Service Administration • Service Configuration

Table 57. Services elements (continued)

Services element	Element type	Description	Page
	**Delete Services** button	Click this button to select the services you want to delete. When you click this button, the **Delete Instances** tab opens in the Service Editor. Select the services you want to delete and click the **Delete** button in the **Delete Instances** tab.	• Service Administration • Service Configuration
	**Find Instance** button	On the **Service Administration** and **Service Configuration** pages, clicking the **Search** button opens a new window that allows you to search for service instances. Enter a search string in the **Search for** field, and click **Search**. The matching results are shown in the table below. For example, if you want to find all the service instances with names like `webserver1`, click the search icon. In the new window, enter `webserver1` in the **Search For** field and click the **Search** button. The table will be populated with matching service instance. If you double-click the instance in the list, the service editor will be populated with that service. On the Service Availability page, when you click one of the instances in the list, it will be displayed and selected in the Service Tree.	• Service Administration • Service Configuration • Service Availability
	**Tree Template Editor** button	Click this button to open an editor where you can create, design, and edit tree templates.	• Service Administration • Service Configuration
	**Help** button	Click this button to access help for the tab.	• Service Administration • Service Configuration
Service	Column	This column lists the names of services instances configured for TBSM. This column is sorted alphabetically by default, but you cannot selectively sort it. **Note:** If your service name is too long, the Service Tree on the Service Availability page automatically wraps the text to the next line.	• Service Administration • Service Configuration • Service Availability

Table 57. Services elements (continued)

Services element	Element type	Description	Page
State	Indicator Column	The **State** indicator shows the status with up-to-the-minute information about the service as a whole, depending on the status of the subservices. The color indicators are: • Green = `Good` • Yellow = `Marginal` • Purple = `unknown` (for events only) • Blue = `Maintenance` • Red = `Bad` You can sort the services according to the status by clicking this column heading.	• Service Administration • Service Configuration • Service Availability
Time	Indicator Column	If a service has cumulative-time SLA settings, the **Time** indicator column shows the status of the SLA. If the service does not have cumulative-time SLA settings this indicator is gray. The color indicators are: • Green = `Good` • Yellow = `Marginal` • Red = `Bad` You can sort the services according to the status by clicking this column heading.	• Service Administration • Service Configuration • Service Availability
Events	Indicator Column	The **Event** indicator shows the highest severity level of related events for the service. The indicator colors match the severity-level colors for ObjectServer events. For example, if the highest severity level is warning, this indicator is blue. You can sort the services according to the event status by clicking this column heading.	• Service Administration • Service Configuration • Service Availability
	Clear settings button	Use the **Clear settings** button to restore the default column-based settings in the Service Tree. The **Clear settings** button will not restore all default portlet settings.	Service Availability

Note: You can customize the Indicator columns to meet your needs.

Services in the Service Tree

This topic describes Services in the Service Tree. This topic also describes how to display the dependency view.

The **Service Tree** contains trees of service-dependency models that have been configured for TBSM. The service instances you see depend on your user and group permissions.

To see the full tree, click the plus symbols to the left of the services. A list of services instances is displayed.

Displaying the dependency view

Clicking a service in the **Service Tree** loads a dependency view of the service and its subservices in the Service Viewer.

"Service Availability Overview" on page 168 shows the dependency view for ABCBankeBanking and its subservices.

Note: If you change rule or template names, you can cause issues with other features of TBSM. TBSM uses these names to reference the rule values for other rules, such as auto-population or ESDA rules and for custom displays such as service trees and view definitions.

If you want to change a rule or template name, change the name before you use the rule or template elsewhere in TBSM.

For example, when you create numerical and text-based rule, you need to create a custom service tree or view definition to display the output of the rule in the TBSM console. The trees and view definitions use the template and rule names to map the rule values to a display object. As a result, if you change the name of a rule or a template used in a custom display, the custom tree or view definition will no longer display the rule values properly.

Service Tree preferences

This topic describes Service Tree preferences.

The **Service Tree preference** window allows you to control the default presentation of a tree and set a default service instance.

Note: Any changes that you make in the Service Tree preferences window are not saved when you close your login session.

To access Service Tree preferences:

1. Click **Personalize Page** option in the Service availability page actions.
2. Select **Personalize**.

   The service tree preferences window opens. The **General** tab shows by default.
3. Set your preferences using Table 58 on page 173 as a guide.

Table 58. Service Tree preferences guidelines

Tab	Description
General	This tab lets you specify the portlet title and the refresh interval of the tree (30 seconds by default).

Table 58. Service Tree preferences guidelines (continued)	
Tab	**Description**
Context	This tab lets you set: • The default starting service instance • Whether the tree accepts context from another page or from other portlets on the same page. See "Set service context" on page 174 for more information.
View	This tab lets you select which Tree Template you want to view in the Service Tree portlet. Select the **Use Depth Shading** option to color each level expanded in the tree to distinguish it from its parent and children. If you do not configure TBSM for OSLC, this option is disabled. This tab also allows you to set the visibility of services in the Service Tree portlet. For more information, see "Service Tree preferences View tab" on page 175 for more information.

4. To save your new preferences, click **Save**.

Note: The **Restore defaults** button lets you restore the preferences to the system defaults.

Set service context

This topic describes the elements on the Service Tree preferences **Context** tab. This topic also describes how to set the options on this tab.

About this task

The **Context** tab options let you specify the default top-level service for your tree, or whether the tree changes in response to context passed from other portlets or pages.

The **Set Starting Instance** option lets you select a default service from a tree of your service models. This service displays as the default in the Service Tree portlet. You can set the starting service instance for your user ID only or for one of the groups you belong to.

The **Change starting instance when Node clicked on or launch page events are received** option lets you specify if the tree accepts context from another page or from other portlets on the same page.

To set the context for a Service Tree portlet, complete the following steps:

Procedure

1. From the **Context** tab in the Service Tree preferences, choose from these options:

   Change starting instance when Node clicked on or launch page events are received.
   When you select this option, the tree accepts context from another page or from other portlets on the same page. When the service tree receives context, it shows either the service passed as context or its child services as the default view. For example, if you select a service in another tree on the same page, the context of this Service Tree changes to show the selected service.

   By default, it is not selected. If it is not selected, the tree does not receive context from other portlets or pages.

   Show starting node in tree
   To use this option, select the service you want to display by default from the Service Tree. Then select this option to show the service as the only root node in the Service Tree.

By default, it is not selected. If it is not selected, the immediate children of the starting instance become the root nodes of the tree.

Broadcast service instance to page after tree loaded
When you configure this option, after the tree is loaded, it broadcasts service context data to the other portlets on the page. This enables the initial content for the page. To enter the broadcast service instance, select one of the following values:

- - 1: Indicates that there is no automatic selection of the initial row.
- 0: Indicates that the first row is automatically selected.
- 1: Indicates that the second row is automatically selected.
- 99: Indicates that the 100th row is automatically selected.

2. Click **OK** to save your preferences.

Service Tree preferences View tab
This topic describes the options in the Service Tree preferences **View** tab. You can also set the visibility for services from this tab.

Display options

You can set these display options in the Service Tree preferences **View** tab:

- Use the **Tree Template** drop-down list to select the tree template where you want to change the hover preview and depth shading options.
- If TBSM is not configured for OSLC, the checkbox is disabled. Before you can use the option, TBSM and the Jazz for Service Management Registry must be configured for hover preview as described in the TBSM *Customization Guide > OSLC hover preview and TBSM*.
- **Use Depth Shading** option colors each level expanded in the tree to distinguish it from its parent and child services.

Service Visibility options

Access the Service Visibility option in the Service Tree preferences **View** tab to do the following tasks:

- Selectively show services in the Service Tree portlet
- Remove services from display in the Service Tree
- Show selected services always

For any services you want to control, you must specify a numeric rule. Each service in the model is then checked for the existence of this rule. If the rule exists and the output is zero, the service is considered invisible and is not displayed in the Service Tree. If the rule does not have an output of zero, the service is displayed in the Service Tree. When a rule does not exist for a service or is not specified, the tree shows the default that you select on the View Tab. All services without a rule are invisible or visible depending on your selection.

Note: Numeric formula rules with a hardcoded numerical value cannot be used to make services always visible because the rule doesl not recalculate its value when the data server is restarted. Use a rule that is driven by an event or data fetcher. Service visibility can be made static by using a numeric incoming status rule that uses a data fetcher to obtain data from a static database table.

If you do not have any numeric rules, you must create a new numeric rule to control visibility for each template in the Service Editor. Otherwise you cannot use this preference option. To create a numeric rule in the Service Editor, see the topic: “Create and edit numerical incoming status rules” on page 73.

To set the Service Visibility option on the Service Tree preferences, follow these steps:

1. Type the rule name you created in the **Visibility Rule Name** field. The name is case-sensitive. This rule name controls what your Service Tree looks like.

Important: Rules are case-sensitive, and you must type the rule name exactly. If the name does not exactly match the rule you created in the Service Editor, the name does not exist to the TBSM program.

2. Select either **Invisible** or **Visible** to specify the default visibility state when the rule does not exist for a particular service.

   Note: If your rule is not found, and you selected **Invisible**, all your services will become invisible.

   Note: If the rule is present on more than one template a service is tagged with, the Service Tree does not work correctly. Create a rule only for the primary template.

3. Click **OK** to save your preferences.

   Note: To change your current settings, return to the Service Tree preferences and edit the settings.

   Restriction: The Service Tree will not dynamically update when template changes are made. For example, tagging or un-tagging a service with a template containing a visibility rule would require the Service Tree to be reloaded by reopening the page.

Note: Deprecation for Jazz™ for Service Management Registry Services is due with version 1.1.3 and thus the Open Services Lifecycle Collaboration (OSLC) Hover Preview functionality, which uses this feature, is also deprecated, when used with versions or Jazz™ for Service Management equal to or greater than version 1.1.3.

Customizing the Service Tree

This topic describes how to customize the Service Tree by customizing columns, using sort options, and the Find tool.

TBSM allows you to customize columns in the Service Tree directly.

Note: Any column customizations you make will be saved until you click **Clear Settings** on the toolbar.

Customizing Service Tree columns

Resizing columns

You can resize columns by clicking column header borders, then dragging to the size you want.

Reordering columns

You can reorder columns by clicking the column headers and dragging them to the wanted place.

Column visibility

You can control what columns are visible in the Service Tree.

To hide any columns from view:

1. Right-click on any column.
2. Select **Column Visibility**.
3. Clear the column you want to hide. The Service Tree automatically refreshes to reflect the change.

   Note: Repeat steps 1 and 2 for any additional columns you want to hide.

To make hidden columns visible, following these steps:

1. Right-click on any column.
2. Select **Column Visibility**.
3. Check that the column you want to see. The Service Tree automatically refreshes to reflect the change.

Locking Columns

This function allows you to lock certain columns and keep them in view horizontally while the remaining columns scroll (left and right). To lock a column:

1. Right-click on the column you want to lock.
2. Select **Lock Column**.

To unlock a column:

1. Right-click on the column you want to lock.
2. Select **Unlock Column**.

Service tree sort options

You can click the column headers in the Service Tree to change the order of the displayed information. When you click the column header, it toggles the sort mode between ascending and descending.

For example, on the initial column click, the Service Tree sorts the services in ascending order. On the second click, the services sort in descending order, and on the third click, the service tree reverts to the original sort order.

Note: By default, the Service name column is automatically sorted in ascending order. The State, Time, and Events columns are sorted in descending order by default.

Service tree Find

When you click the **Find** button on the toolbar, the **Find Service** window opens. To find a service or display name:

1. Enter the service or display name you want to find.

Wildcard Character	Use
*	This wildcard is used to match zero or more characters.
?	This wildcard is used to match any single character.

   Note: There is an implied * wildcard at the beginning and end of the filter by default.
2. Click **Find**. A list of instances pops up.
3. Find the service you want on the list and click it. The **Find Service** window closes, and your service is highlighted in the Service Tree.

Service Viewer preferences

This topic describes how to use the Service Viewer preferences.

Note: This feature will not be part of subsequent TBSM releases.

The **Service Viewer preference** window allows you to control the default presentation of the Service Viewer and set a default service instance.

To access Service Viewer preferences:

1. To access the service viewer preferences, open the Service Availability page.
2. Select **Edit Shared Settings** from the Edit drop-down menu.

   The service viewer preferences window opens. The General tab shows by default.
3. Set your preferences using Table 59 on page 177 as a guide.

Table 59. Service Viewer preferences guidelines

Tab	Description
General	This tab lets you specify a name for the portlet.
Context	This tab lets you set the starting instance. See “Service Viewer preferences Context tab” on page 178 for more information.

Table 59. Service Viewer preferences guidelines (continued)	
Tab	**Description**
Viewer	This tab lets you set view definitions and toolbar options. See “Service Viewer preferences Viewer tab” on page 178 for more information.

4. To save your new preferences, click **Save**.

   Note: If you want to return to the system defaults for the portlet, click **Restore defaults**.

Service Viewer preferences Context tab

This topic describes the elements on the Service Viewer preferences Context tab. This topic also describes how to set the options on this tab.

About this task

The Set Starting Instance option on the Context tab allows you to select a default service from a tree of your service models. This service displays as the default when you log in to TBSM. You can set the starting instance for your userid only or for one of the groups you belong to. This section describes the steps to select a default service instance to be displayed when you log in. To select the service instance you want to open when a user logs in to TBSM, complete the following steps:

Procedure

1. From the Context tab in the Service Viewer preferences, select the service you want to display by default from the list of services.
2. Click **OK** to save the preferences.

Service Viewer preferences Viewer tab

This topic describes the elements on the Service Viewer preferences View tab. This topic also describes how to set the options on this tab.

The service viewer preference window allows you to control these elements of a Service Viewer portlet:

- View Definition
- Levels up & down
- Visibility of Controls

The defaults for each option in the viewer tab are described in the table below.

Table 60.		
Option	**Description**	**Default Setting**
View Definition	This option controls which view definition you see by default.	The view definition is set to Relationships.
Levels up & down	This option controls how many levels appear in your service model hierarchy.	Levels up: 1. Levels down: 3.
Visibility of Controls	You can choose what controls are visible. You can choose to display the View Definition toolbar, the Standard toolbar, or the menu bar. You can select all the options or any combination of these options you want.	Menu bar, Standard toolbar, and View Definition toolbar are all selected.

To change these options, follow these steps:

1. Select the view definition you want to display from the list of available view definitions.
2. To change the levels, select the check box **Override levels up and down with:**
3. Use the **Up** and **Down** fields to specify the number of services levels you want to see above and below your selected service.

   Note: To show no ancestors, specify 0 in the **Up** field. To show no descendents, specify 0 in the **Down** field.
4. Select menu bar, Standard toolbar, View Definition toolbar, or all to get the toolbar settings you want. Table 61 on page 179 describes these options.

Table 61. Service Viewer toolbar options

Toolbar option you want to view	Settings
Menu bar	Make sure only the. menu bar is selected
Standard	Make sure that only the. Standard check box is selected
View Definition	Make sure that only the. View Definition checkbox is selected
All selected	This setting is the. Make sure the menu defaultbar, the Standard toolbar, and the View Definition toolbar check boxes are all selected.

   Note: This feature can also be found in the Service Editor **View** tab menu and in the Service Viewer menu. Select **View ► Toolbars**. Select or deselect **Standard** and **View Definition** to get the settings you want.
5. To save your new preferences, click **OK**.

   Note: If you want to return to the system defaults for the portlet, click **Restore defaults**.

View Service toolbar and menus

This topic describes the elements of the View Service toolbar and menu.

The options on the View Service toolbar and menus let you customize and save the views in the **View Service** tab.

The figure shows the toolbar.

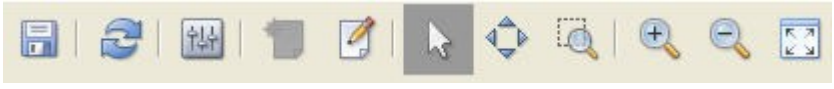


Figure 27. View Service toolbar

Save

Clicking the **Save** button allows you to save a custom canvas.

Figure 28. Save button

Viewer Preferences tool

The **Viewer Preferences** tool allows you to set your display preferences for the **View Service** tab.

Figure 29. Viewer Preferences tool

Click the **Viewer Preferences** button in the toolbar to open the **Viewer Preferences** window.

You can change the following settings in the **Viewer Preferences** window:

- If you have `Save Canvas Layout for Group` permissions, whether the views you save and delete affect your entire user group.
- The default visual elements for service instance images.

If you have the `Save Canvas Layout for Group` role, the **Save** tab appears in the **Viewer Preferences** window. The **Save Custom Views** option lets you set whether the views you save affect your entire user group.

To set your view saving preference, complete the following steps:

1. From the **Save Custom Views** option in the **Viewer Preferences** window, select the radio button for the option you want. If you select **For Group**, any views you save and delete will be saved for the selected group.
2. Click **OK**.

Create Custom Canvas tool

Clicking the **Create Custom Canvas** button allows you to create a custom canvas.

Figure 30. Create Custom Canvas button

Palette tool

Clicking the **Palette** button displays the custom indicator and decoration palette. For more information about creating custom views, see the *IBM Tivoli Business Service Manager: Administrator's Guide*

Figure 31. Save button

Refresh tool

The **Refresh** tool allows you to refresh the status of the services displayed in the dependency view.

Figure 32. Request Status/Instances Refresh button

Use the **Refresh** tool to get an updated view, including changes in status and service dependencies. Changes in status are indicated by changes in the color of the service's visual elements.

Select tool

The **Select** tool allows you to select an individual service or visual element in custom views. You must be in a custom view to move or delete a visual element.

Figure 33. Select tool button

Pan tool

The **Pan** tool allows you to grab all the items in the dependency view and move them around the window.

Figure 34. Pan tool button

To use this tool, complete the following steps:

1. Click the **Pan** button. The cursor turns into a figure of a hand.
2. Click any service in the dependency view.
3. Move the displayed services to a new point in the pane.
4. Click the **Pan** icon again to turn off the tool.

Zoom Box tool

The **Zoom box** tool allows you to select a service, several services, or part of a service rectangle in the dependency view to zoom in on the area.

Figure 35. Zoom Box tool

To use this tool, complete the following steps:

1. Click the **Zoom box** tool.
2. Draw a rectangle around the service, services, or a portion of the service, that you want to zoom in on.
3. To make the view smaller, click the **Zoom out** button until the view is restored to the size that you want.

   You can return the view to its original position in the pane by using the **Fit to View** tool.

Zoom In tool

The **Zoom in** tool allows you to enlarge the whole view.

Figure 36. Zoom In tool

To use this tool, complete the following steps:

1. Click the **Zoom in** button.

   The entire view becomes larger.
2. To restore the view to its original size, use the **Zoom out** or **Fit to View** buttons.

Zoom Out tool

The **Zoom out** tool allows you to reduce the whole view.

Figure 37. Zoom Out tool

To use this tool, complete the following steps:

1. Click the **Zoom out** button.

   The entire view becomes smaller.
2. To restore the view to its original size, use the **Zoom in** or **Fit to View** buttons.

Fit Contents in View tool

The **Fit Contents in View** tool allows you to make the view fit to the size of the pane. By default, service models are set to display the entire model in the view.

Figure 38. Fit Contents in View tool

Click the **Fit Contents in View** button. This view enlarges or reduces to fit the pane.

View Service menus

This topic describes the View Service menus.

The menus give you additional options that are not on the toolbar.

File menu

The **File** menu are described in the table below.

Table 62.

Toolbar button	Description
Reload Style sheets	Selecting this option will allow you to reload the style sheets.
Clear Instances Cache	Selecting this option will allow you to clear the instances cache.

Edit menu

The **Edit** menu contains the options as described in Table 63 on page 183.

Table 63. Edit menu Options	
Toolbar button	**Description**
Cut	Selecting **Cut** lets you cut the selected service indicator. To use this tool: 1. Select an indicator with the Arrow tool. 2. Select Cut from the Edit menu to remove the selected indicator. You can paste the item using the Paste option in the Edit menu.
Paste	Selecting **Paste** lets you paste a cut service indicator. To use this tool: 1. Cut an indicator. 2. Change views to another custom view. 3. Select the Paste option from the Edit menu.
Delete	Selecting this option lets you delete the service indicator you select. To use this tool: 1. Select an indicator with the Select tool. 2. Select Delete from the Edit menu to remove the selected indicator.
Bring to Front	When you are creating a custom canvas, if you have overlapping service indicators or decorations, selecting the **Bring Visual Element to Front** option sends the item to the front.
Send to Back	When you are creating a custom canvas, if you have overlapping service indicators or decorations, selecting the **Send Visual Element to Back** option sends the item to the back.
Fill Background	You can choose the color of the background of your service model by selecting this option. Clicking the **Reset** button at any time returns the background to default settings.

View menu

The **View** menu contains the options as described in Table 64 on page 183.

Table 64. View menu options	
Toolbar button	**Description**
Grid	Selecting **Grid** displays a grid in the `Custom View`.

Table 64. View menu options (continued)	
Toolbar button	**Description**
Magnifier	This option allows you to magnify one area of your service model. To use this tool: 1. From the **View** menu, select **Magnifier**. 2. Click the area you want to see. 3. Click again and hold, and you can see the "snapshot" of the area that you wanted magnify. 4. To turn off this feature, deselect **Magnifier** in the **View** menu.
Overview	Use the Overview window to pan across a dependency view when it is magnified. The Overview window is especially useful for viewing complicated service models with many service images. To learn how to use this tool, see “Overview tool” on page 184 for more information.
Inspector	Selecting this option opens the Inspector.
Toolbars	You can set toolbar settings to display the View Definition toolbar, the Standard toolbar, or both. Select or deselect Standard and View Definition to get the settings you want.

Overview tool
This topic describes the overview tool.

About this task

The **Overview** tool gives you a more flexible **Overview** window. Use the **Overview** window to pan across a dependency view when it is magnified. The **Overview** window is especially useful for viewing complicated service models with many service images.

Use this tool as follows:

Procedure

1. Select the **Overview** tool from the **View** menu.

   The **Overview** window is displayed in a new browser window.

   The red box within the Overview window is the overview area for the service model. This overview area matches the viewable area of the service model in the **View Service** tab. That is, the portion of the service model highlighted in the overview area is displayed in the **View Service** tab.

   For example, if you click the **Zoom-in** button in toolbar, the overview area shrinks in proportion to the zoom level. You can also change the overview area size from within the **Overview** window.
2. Use the **Zoom-in**, **Zoom-out**, or **Zoom box** buttons in the toolbar to change the size of the overview area.
3. To change the size of the overview area from the **Overview** window, click and drag one of the corner-selection blocks.
4. To change the position (focus) of the overview area, click and drag the red box to the part of the service model you want to view.

Quick Find

This topic describes the Quick Find function of the Service Editor/Viewer.

About this task

Quick Find allows you to search for services by name in the Service Viewer/Editor portlet. You can navigate between individual matches using the **Previous** and **Next** buttons.

To use the Quick Find:

Procedure

1. Enter the name you want to find in the Quick Find field. See the table below for valid wildcard characters.

Option	Description
*	This wildcard is used to match zero or more characters.
?	This wildcard is used to match any single character.

2. Press **Enter**. The view automatically zooms to show all matches. You may change the scope of the view by adjusting the zoom using buttons on the toolbar.
3. Click **Previous** or **Next** to find what you want.

   Note: If no service names match, the Quick Find field will turn red.

Editing service configuration data

This topic describes how to edit service configuration data.

If you have permission to edit a service, you can display and edit the following configuration information for a service from the **Services** tree.

Note: These options are only available on the Service Administration and Service Configuration pages.

Editing service instance data

This topic describes how to edit service instance data.

About this task

If you have permission to change the configuration of a service, you can open the **Edit Service** tab as follows:

Procedure

1. Right-click the service in the services tree or the Service Editor.
2. Select **Edit Service Instance** from the menu that appears. You can also display this tab by clicking the **Edit Service** tab in the Service Editor.

Editing service template data

This topic describes how to edit service template data.

About this task

If you have permission to change the configuration of a service's template, you can open the **Edit Template** tab as follows:

Procedure

1. Right-click the service from the services tree or the Service Editor.

2. Select **Edit Member Templates** > **Template Name** from the menu that appears, where **Template Name** is the name of the service template you want to edit.

   The **Edit Template** tab is displayed in the Service Editor.

Removing child service dependencies

This topic describes how to remove child services from dependency models.

About this task

You can remove child-service dependencies from the services tree or the *Service Editor* using the right-click menu options.

This procedure removes the service from the dependency model, but does not delete the service from the TBSM database. To remove a child service from a service-dependency model, complete the following steps:

Procedure

1. Right-click the service instance you want to remove.
2. Select **Children** > **Remove as Child** from the menu that appears.
3. Select the template you want to remove.

   The selected service is removed from the service-dependency model.

Sending test events

This topic describes how to send test events.

About this task

If a service has incoming status rules configured, you can send test ObjectServer events from the **Service Configuration** > **Services** drop-down menu or the services tree. These test events let you see how the dependency model responds to a given event or events. To send test events, complete the following steps:

Procedure

1. Right-click the service.
2. Select **Show** > **Send Test Event** from the menu that appears.

   Note: If you have more than one incoming status rule, **Send Test Event** will send events that match the filters in the first rule only. If you have multiple rules, use the `rad_sendevent` script, as described in the *TBSM Administrator's Guide*.

   The **Event Fields** window is displayed in a dialog that pops up. The fields in this window match the identification and output threshold fields defined in the services template.
3. Enter the field values you want for your test event.
4. Click **OK**.

   The test event is sent to the ObjectServer.

Chapter 25. Urgent Services portlet

The Urgent Services portlet displays a list of all services that have a status of critical.

Urgent status threshold

By default, the Urgent Services portlet displays services with a status of bad (5). You can set your preferences to filter Urgent Services by severity or tagged template. For information about changing the urgent status threshold, see the *TBSM Administrator's Guide*.

Urgent Services update frequency

By default, the Urgent Services portlet updates every 30 seconds. You can set your preferences to use a different update frequency. For information about changing the urgent services update frequency, see the *TBSM Administrator's Guide*.

Adding the Urgent Services portlet

By default the Urgent Services portlet only displays on the Service Availability page; it does not display in the Service Administration page on the TBSM console. To add this portlet to a page, an administrator needs to add the Urgent Services portlet to a page, such as the Service Administration page. For information about adding the Urgent Services portlet, see the *TBSM Administrator's Guide*.

Showing an urgent service in other portlets

You can display one of the listed services in context in the other portlets by clicking anywhere on the row of the service name.

Urgent Services portlet elements

The Urgent Services portlet lists the services with an urgent status and the last time the service status changed.

Table 65. Urgent Services portlet elements

Element name	Description
Service column	The service names display in this column. You can sort the service names alphabetically when you click **Service** in the title row for this column. You can switch between ascending and descending sort order when you click the up/down arrow. To display a service in the Service Viewer or update other portlets with data from a service, click the service name in this column.
State column	The status of the services displays in this column. You can sort the service names by status when you click **State** in the title row for this column. You can switch between ascending and descending sort order when you click the up/down arrow.
Last Changed column	This column shows the last time the status of the service changed. You can sort the service names by time when you click **Last Changed** in the title row for this column. You can switch between ascending and descending sort order when you click the column name again.
Selectable Rows	To show a service in the service viewer, click the service name.

Urgent Services preferences

This topic describes how to use the Urgent Services preferences.

The **Urgent Services preference** window allows you to filter the services in the Urgent Services portlet. You can set your preferences to filter services by severity or template.

To access Urgent Services preferences:

1. Open a page containing an Urgent Services portlet.

   Note: The Service Availability page contains the urgent services portlet by default.
2. Click **Edit Page** in the **Service Availability** page.
3. Select **Edit** in the **Urgent Services** portlet.

   The Urgent Services preferences window opens. The General tab shows by default.
4. Set your preferences using the table below as a guide.

Table 66. Urgent Services preferences guidelines	
Tab	**Description**
General	This tab lets you specify a name for the portlet. This tab also allows you to set the refresh interval (in seconds) for the portlet.
Filtering	This tab lets you filter by severity and template. See "Urgent Services preferences Filtering tab" on page 188 for more information.

5. To save your new preferences, click **Save**.

   Note: If you want to return to the system defaults for the portlet, click **Restore defaults**.

Urgent Services preferences Filtering tab

This topic describes the elements on the Urgent Services preferences Filtering tab.

The Urgent Services preference window allows you to filter services by:

- Severity
- Template

The defaults for each option in the Filtering tab are described in the table below.

Table 67.		
Option	**Description**	**Default Setting**
Filter by Severity	This option will filter the portlet results with services that are only Bad, only Marginal, or both.	**Bad** is selected.
Filter by Template	This option will filter the portlet results to only show services tagged with the templates you select.	No templates are selected.

To change these options, follow these steps:

1. Select the severity filter in the urgent services portlet, select **Bad**, **Marginal**, or both to get the setting you want. The table below describes these options:

Table 68. Severity options	
Service state you want to view	**Settings**

2. To filter by template, select a template from the list of templates. The table below describes template options:

Table 69. Template options		
Services you want to view	**Description**	**Settings**
Services tagged by template		Select the. You can choose templates youmore than one want the portlettemplate to to be filteredmatch. with
Primary template only	The template. selected from the list must be the primary template for the services you want to see	

3. To save your new preferences, click **Save**.

   Note: If you want to return to the system defaults for the portlet, click **Restore defaults**.

Chapter 26. Right-click menus

Right-click menus give you quick access to many of the editing functions in IBM Tivoli Business Service Manager (TBSM). The menus are accessible from the **Service Navigation** portlet and the **Service Editor**.

About the right-click menus

This topic describes right-click menus.

Many of the TBSM functions that are available in many parts of the system are also available by right-clicking a service either in the **Service Navigation** portlet, Service Tree, the Service Editor, or the Service Viewer. Not all right-click menu options are available on every portlet.

Note: The set of menu options are reduced if a Service Editor portlet is not on the page. Certain actions that require the Service Editor portlet will not be available.

TBSM has the following default right-click menu options:

- Edit Service Instance
- Delete Service Instance
- Children

  The **Children** option has a submenu containing the **Remove as Child** option.
- Edit Member Templates

  The **Edit Member Templates** option has a submenu containing the templates that can be edited.
- Show

  The **Show** option has the following submenu selections:

  – Service Affecting Events (Table)
  – Service Affecting Events (AEL)
  – Show Rule Status
  – Show Root Cause Events
  – Send Test Event
  – Show 30-Day History
  – Compare Service Metrics
- Launch to

  The **Launch to** option has a submenu containing launch options to other external applications. This list is dynamic, and what is shown varies from resource to resource. The **Launch to** option may have the following submenu selections:

  – Show Managed Systems (Tivoli Enterprise Portal)
  – Show Application Server (ITCAM for SOA)
  – Show HOP View (ITNM)
  – Show Physical Topology (Tivoli Application Dependency Discovery Manager)
  – Show Change History (Tivoli Application Dependency Discovery Manager)
  – View Incidents (SRM)
  – Open Process Request (CCMDB)
  – View Changes (CCMDB)
  – Show Storage Topology (Tivoli Storage Productivity Center)

- Maintenance window tools

  The **Maintenance** option has the following submenu selections:

  – Schedule Maintenance
  – Remove from Maintenance

You can add to and edit these options, called "Right-Click Actions," for custom views of your service configuration. The actions you add or modify will only be visible from the defined custom view. They will not be available from the default view or when right-clicking the services in the **Service Navigation** portlet.

Edit Service Instance action

This topic describes Edit Service Instance action.

The Edit Service Instance action opens the Service Editor for the selected service.

Children actions

This topic describes Children actions.

From the **Children** menu option, you can select the **Remove as Child** option.

Remove as Child action

This menu option allows you to remove the selected instance as a child of a service. To remove a child service, follow these steps:

1. Select this menu option. A submenu with a list of parent services for the selected service opens.
2. Choose the parent service you want the selected child service removed as a dependent. The child service is removed from the parent-specific service dependency model.

Edit Member Templates action

This topic describes Edit Member Templates action.

Selecting the **Edit Member Templates** menu option allows you to edit any templates that this service has been tagged with.

When you select a template, the **Edit Template** window opens in the Service Editor.

Show actions

This topic lists the actions available from the Show right-click menu.

From the **Show** menu options, you can select the following actions:

- Service Affecting Events (Table)
- Service Affecting Events (AEL)
- Show Rule Status
- Show Root Cause Events
- Send Test Event
- ITCAM for Internet Service Monitoring Report Viewer
- Show 30-Day History

Service affecting events (table) option

This topic describes Service affecting events (table) option.

When you select the Service Affecting Events (Table) option from the **Show** menu (when you have event-based incoming status rules set up for the selected service), the events display in table format in the **Events** tab of the **Service Details** portlet. This selection does not work for incoming status rules based on an SQL data fetcher.

Service Affecting Events (AEL) option

This topic describes Service affecting events (AEL) option.

When you select the Service Affecting Events (AEL) option from the **Show** menu (when you have event-based incoming status rules set up for the selected service), the events display in an Active Events List. This option does not work for incoming status rules based on an SQL data fetcher.

Show Rule Status option

This topic describes Show Rule Status option.

When you select the Rule Status option from the **Show** menu, the rules display in table format in the **Rules** tab of the **Service Details** portlet. The TBSM events are generated any time the status of a service changes. It changes the status based on formula and aggregation rules as well as incoming status rules.

Show Root Cause Events

This topic describes Show Root Cause Events option.

When you select the Root Cause Events option from the **Show** menu (when you have event-based incoming status rules set up for the selected service), the root cause events display in table format in the **Events** tab of the **Service Details** portlet. The root cause events are the events that cause the service's status to change.

Send test event

This topic describes Send test event option.

You can send a test event when you want to test your service configuration and their related incoming status rules. Do not use this option if you have more than one event-based incoming status rule set for the services tagged with a template. Instead, use the `rad_sendevent` script. For more information about this script, see the *IBM Tivoli Business Service Manager: Administrator's Guide*.

Show 30-day history

This section describes the option that lets you launch the Service Outage Detail report.

This option launches the Service Outage Detail report for the selected service in a new browser window.

Compare Service Metrics

This topic describes Compare Service Metrics option.

This option launches the Time Window Analyzer **Compare** frame. The service you selected is automatically selected in the **Compare** frame to view in the chart, and history for that service.

Launch to other Tivoli consoles

This topic describes the **Launch to** option.

The **Launch to** menu has launch options to other external applications such as Tivoli Enterprise Portal and Tivoli Application Dependency Discovery Manager consoles. This list of options is dynamic, and what is shown varies from resource to resource. This menu option is only applicable to services that have been imported with the Service Component Repository. All other services will not have this option enabled in the Service Navigation portlet.

If a service contains data about a Tivoli Enterprise Portal and Tivoli Application Dependency Discovery Manager server, these options will be enabled in the **Launch to** menu. The CCMDB options (Show CI details and Open Service Request) will not initially appear on the submenu. You need to set the CCMDB host data in the `RAD_sla.props` file and recycle the dashboard server for those options to appear. They will only be enabled for those services that have been discovered with the Discovery Library Toolkit from the Tivoli Application Dependency Discovery Manager. When you select one of these options, the Tivoli Application Dependency Discovery Manager and Tivoli Enterprise Portal console open in a new browser window.

For more information about these launch options, see the topic "Launching to and from Applications" in the *IBM Tivoli Business Service Manager: Customization Guide*.

Maintenance window tools

This section describes the options that let you change the maintenance status for a service.

The maintenance window tools let you perform the following actions:

- Quickly change the maintenance period for a service. You can also set the scope of the schedule by choosing to set the schedule for all its related services as well, or for the related services only.
- Quickly remove the maintenance status from the service. You can set the scope of the removal for this option as well.

Chapter 27. TBSM reports

TBSM reports can be viewed through Cognos Analytics Server V11.x.

You can load the TBSM reports using Cognos Administration as described in the Installing the Historical Reports section of the *IBM Tivoli Business Service Manager Installation Guide*. Cognos provides options to view and save the report output in a range of formats, for example HTML and PDF. For more information about configuring reports, and for information about the full range of tasks that you can perform, see the Cognos Analytics Server V11.x Knowledge Center or the Custom Reports section of the *IBM Tivoli Business Service Manager Customization Guide*. Cognos reports can be customized with the Reporting application. See the Cognos V11.x Knowledge Center for more information about report customization.

Note: BIRT reports are now out of support.

Run TBSM reports

This section describes how to open the TBSM report package and how to run reports.

Before you begin
The TBSM reports include reports developed with Cognos technology. Cognos provides additional functions that enable you to customize the Cognos reports. To access the TBSM reports, the reporting package must be installed on the Cognos Analytics server as described in the *TBSM Installation Guide*.

About this task
Log in to the Cognos Console where you have imported the TBSM reports.

Procedure

1. In the navigation pane on the top-left side of the window, click on **Team content**.
2. In the expanded navigation pane, click **IBM Tivoli Business Service Manager History Agent**.

   There are three report sets: Detail History Reports, Summary History Reports, Top Level History Reports.
3. Click the report set that you require. In the same pane, a list of reports is populated.
4. To display the report in HTML format, click the report name and provide the requested parameters.

   To display a menu of other output formats, click **Run with options** button to the right of the report name.

   You can select a report option to schedule the report. For more information about report options, see the *IBM Cognos Analytics - Reporting User Guide*.
5. To specify a specific date range for the Cognos reports, select the **Date Range** option from the list and then specify the start and end date in the date widgets.

   If you want to select a date option other than a specific date, select an option in the Date Range pull-down.
6. Click **Finish**. The report is formatted and displayed.

What to do next

For information about how you can customize the reports shipped with TBSM or create your own custom reports and use them alongside the TBSM reports, see the *TBSM Customization Guide*.

TBSM Average Outage Duration report

This report shows the average duration of outages for each top-level service.

Description

This report shows the average duration of outages for each top level service, a service with no parent. The average is shown for the period including the 7 days, 30 days, and 90 days prior to the end date of the report.

Parameter

When you run this report, you can specify this parameter:

Report End Data
Use this parameter to select the end date for the report.

TBSM Average Daily Outage Count report

This report shows the average number of outages per day for each top-level service.

Description

This report displays the number of outages per day for each top-level service that experienced an outage. Top-level services are services that do not have a parent. The value displayed is the number of service outages divided by the number of days in the time period. The value displayed is rounded to 1 decimal place. For example, if a service had 3 outages in the previous 7 days, the value displayed would be 3/7 = 0.4.

Parameter

When you run this report, you can specify this parameter:

Report End Date
Use this parameter to select the end date for the report.

TBSM Service Outage Average Duration chart

This report shows the average duration of outages recorded by TBSM during the report period.

Description

The chart shows a bar representing the average duration for the total number of outages next to the name of the affected service. The table displayed for this report lists the affected services and durations grouped by the status level.

Parameter

When you run this report, you can specify this parameter:

Date Range
Use this parameter to select the time interval for the report. Select a time interval from the list provided. If you choose **Date Range (below)**, you must specify values in the **Start Date** and **End Date** fields.

TBSM Service Outage Count Summary report

This report shows the number of outages recorded by TBSM during the report period.

Description

The report displays a count of outages for each service that has a status of bad. A chart displays the count for each service. Click on any service listed in the chart to review the Service Outage Details report for that service. The date range shown in the Service Outage Details report reflects the range chosen for the TBSM Service Outage Count Summary report.

Parameter

When you run this report, you can specify this parameter:

Date Range
Use this parameter to select the time interval for the report. Select a time interval from the list provided. If you choose **Date Range (below)**, you must specify values in the **Start Date** and **End Date** fields.

TBSM Service Path Outage Summary report

This report shows the number of outages recorded by TBSM during the report period for each service.

Description

This report shows the number of outages for each service in your model based on the parameters you specify.

Parameters

When you run this report, you can specify these parameters:

Date Range
Use this parameter to select the time interval for the report. Select a time interval from the list provided. If you choose **Date Range (below)**, you must specify values in the **Start Date** and **End Date** fields.

Top Level Path
You can limit the report to the subset of services under a portion of the service hierarchy tree by specifying the path, starting from the top of the tree, to the portion that you want to include in the report. The default value of `All` shows outages for all the paths. A list of the service hierarchy paths is provided.

Severity
You can limit the status events displayed based on the severity. Choosing 0 displays all status changes. A Marginal status severity is determined by a value of 3 and critical status severity is determined by a value of 5.

Note: To display a report for a specific service, use the TBSM Service Outage Detail report. To see the TBSM Service Outage Detail Report for a given service, click the status time within the list for the service.

TBSM Top Level Path Status Changes report

This report displays a list of service status changes, recorded by TBSM for the specified Top Level Path, ordered by date.

Description
Each row in the table shows the service affected, the time of the status change, the new status, the previous status, and a path from the root of the service tree to the affected service. The duration of the outage is show if the outage has ended, that it to say, if the current status is good.

Parameters

When you run this report, you can specify these parameters:

Date Range
Use this parameter to select the time interval for the report. Select a time interval from the list provided. If you choose **Date Range (below)**, you must specify values in the **Start Date** and **End Date** fields.

Top Level Path
The Top Level Path parameter specifies the service hierarchy path for the displayed services. Select a specific path or select **All** to view all paths.

TBSM Service Outage Duration report

This report displays a list of service outages that have ended for the specified Top Level Path. The service outages are ordered by duration.

Description
Each row in the table shows the service affected, the time of the status change, the new status, the previous status, the duration of the outage, and a path from the root of the service tree to the affected service.

Parameters

When you run this report, you can specify these parameters:

Date Range
Use this parameter to select the time interval for the report. Select a time interval from the list provided. If you choose **Date Range (below)**, you must specify values in the **Start Date** and **End Date** fields.

Top Level Path
The Top Level Path parameter specifies the service hierarchy path for the service affected by an outage. Select a specific path or select **All** to view all paths.

TBSM Service Affecting Events report

This report in shows the events that might have triggered service outages reported in TBSM for the specified service.

Description

The table lists the time of the event, the time its state changed, the event identifier and the event summary message.

Parameters

When you run this report, you can specify these parameters:

Date Range
Use this parameter to select the time interval for the report. Select a time interval from the list provided. If you choose **Date Range (below)**, you must specify values in the **Start Date** and **End Date** fields.

Service Name
Specify the service name. Select the service name you want from the list provided.

Top Level Path
The `Top Level Path` parameter specifies the service hierarchy path for the selected service. Select a specific path or select **All** to view all paths.

TBSM Service Outage Details report

This report shows the outages recorded by TBSM for a single service.

Description

The table shows the details of the status changes, including the time at which the status change occurred in TBSM, the new status, the previous status and the duration of the outage if this row represents a change from bad status. You can click the status time value in a row to display a report showing the events that might have triggered the status change for this service. When you display the Service Affecting Events report this way, it only includes events that occurred at about the same time as the outage time.

Parameters

When you run this report, you can specify these parameters:

Date Range
Use this parameter to select the time interval for the report. Select a time interval from the list provided. If you choose **Date Range (below)**, you must specify values in the **Start Date** and **End Date** fields.

Service Name
Specify the service name. Select the service name you want from the list provided.

Top Level Path
Select the service hierarchy path for the service name.

TBSM Top Service Outage Average Duration chart

This report shows the services with the highest average outage duration recorded by TBSM.

Description

The chart shows a bar representing the total outage duration next to the name of the affected service. The tabular part of the report lists affected services and outage duration totals grouped by status level. You can click on the service in the tabular data to view the Service Outage Detail for the service.

Parameters

When you run this report, you can specify these parameters:

Date Range
Use this parameter to select the time interval for the report. Select a time interval from the list provided. If you choose **Date Range (below)**, you must specify values in the **Start Date** and **End Date** fields.

Row Limit
Select the number of services shown in the report. The number of services shown is determined by the Row Limit. However, any services that have the same average outage duration as the last item defined by the Row Limit are also displayed.

TBSM Top Service Outage Count Summary chart

This report shows the services with the highest number of outages recorded by TBSM.

Description

The chart shows a bar representing the count next to the name of the affected service. The tabular part of the report lists affected services and counts grouped by status level.

Parameters

When you run this report, you can specify these parameters:

Date Range
Use this parameter to select the time interval for the report. Select a time interval from the list provided. If you choose **Date Range (below)**, you must specify values in the **Start Date** and **End Date** fields.

Row Limit
Select the number of services shown in the report. The number of services shown is determined by the Row Limit. However, any services that have the same outage count as the last item defined by the Row Limit are also displayed.

Chapter 28. Custom service trees

This section describes how to create custom Service trees in the **Service Navigation** portlet.

About custom service trees

This topic describes custom service trees in the Service Navigation portlet.

IBM Tivoli Business Service Manager (TBSM) lets you modify the columns in custom trees in both **Service Navigation** and **Service Tree** portlets.

Note: This topic only describes the **Service Navigation** portlet custom trees. See "Customizing the Service Tree" on page 176 for information about customizing the **Service Tree**.

The default **Service Navigation** portlet **Services** tree has three columns: State, Time, and Events. You can modify, delete, and add tree columns with the *Tree Template Editor*. The *Tree Template Editor* is available from the **Services** toolbar in the **Service Navigation** portlet. For each custom column, you use the *Tree Template Editor* to specify the rule data you want to display in the column.

You cannot change what Tree Template is assigned to the service tree in the **Service Navigation** portlet. You can assign a different tree template to trees in the **Service Tree** portlet by editing the portlet preferences. This allows you to create pages with multiple service tree portlet instances, each with its own custom tree template.

Figure 39 on page 201 shows a **Services** tree customized to display a fourth column called Tickets. This column displays the number of trouble tickets for a service, based on the output of numerical rules in the template assigned to the service.

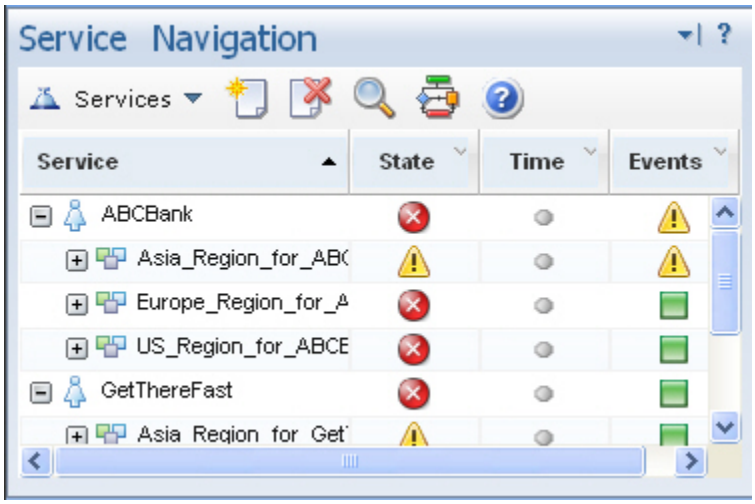


Figure 39. Example custom tree

Before you edit tree templates

Before you edit the tree templates for TBSM, you must configure the data sources, data fetchers, and service templates with numeric aggregation rules to collect the data you want to display in your custom service-tree columns.

When you set up your data source, your data fetcher to select the data you want, and configure your service templates and rules, you can configure a custom service tree to display the rule data.

Note: If you change rule or template names, you can cause issues with other features of TBSM. TBSM uses these names to reference the rule values for other rules, such as auto-population or ESDA rules and for custom displays such as service trees and view definitions.

If you want to change a rule or template name, change the name before you use the rule or template elsewhere in TBSM.

For example, when you create numerical and text-based rule, you need to create a custom service tree or view definition to display the output of the rule in the TBSM console. The trees and view definitions use the template and rule names to map the rule values to a display object. As a result, if you change the name of a rule or a template used in a custom display, the custom tree or view definition will no longer display the rule values properly.

Getting started with the Tree Template editor

This topic describes the Tree Template editor configuration settings. This topic also describes the features of the Tree Template editor: column configuration, service template selection, service template rule mapping, and how to view custom trees.

The *Tree Template Editor* allows you to modify the appearance of the columns for trees in TBSM that have been assigned the tree template being edited in the **Service Navigation** portlet. This section describes how to open the *Tree Template Editor* and modify the settings within the editor.

Tree template configuration settings

The **Tree Template Name** drop-down list in the *Tree Template Editor* lets you specify the tree template you want to edit. By default, TBSM includes three tree templates:

- The `ServiceInstance` tree template defines layout of the **Services** in the **Service Navigation** portlet. This tree template defines the `State`, `Time`, and `Events` columns.

  Note: If you modify this tree template, you will be modifying the view of all TBSM trees (unless they have been assigned a different template).
- The `ServiceTemplate` tree template defines layout of the **Templates** in the **Service Navigation** portlet. This tree template contains no column definitions.
- The `ServiceComponentRepository` tree template defines the layout for the Service Component Repository tree in the **Service Navigation** Portlet. This tree template contains no column definitions and should not be edited.

Column configuration

The TBSM *Tree Template Editor* lets you add, change, and delete the columns in a given tree template.

Service template selection

The TBSM *Tree Template Editor* lets you specify the service templates that you want to use for your service tree-column data. The `Default Tag` template contains the rules for the default **Services** tree.

Service template rule mapping

When you add a rule to a column, you select the service template, the rule, and the service-tree column that displays the rule data.

Viewing custom trees

After modifying a tree template for a given tree, you need to refresh the portlet that contains the tree.

Opening the Tree Template Editor

This topic describes how to open the Tree Template Editor.

About this task

To open the Tree Template Editor:

Procedure

1. From the **Service Navigation** portlet, select **Services** from the drop-down menu.
2. Click the **Tree Template Editor** button.

   The *Tree Template Editor* opens as shown in Figure 40 on page 203.

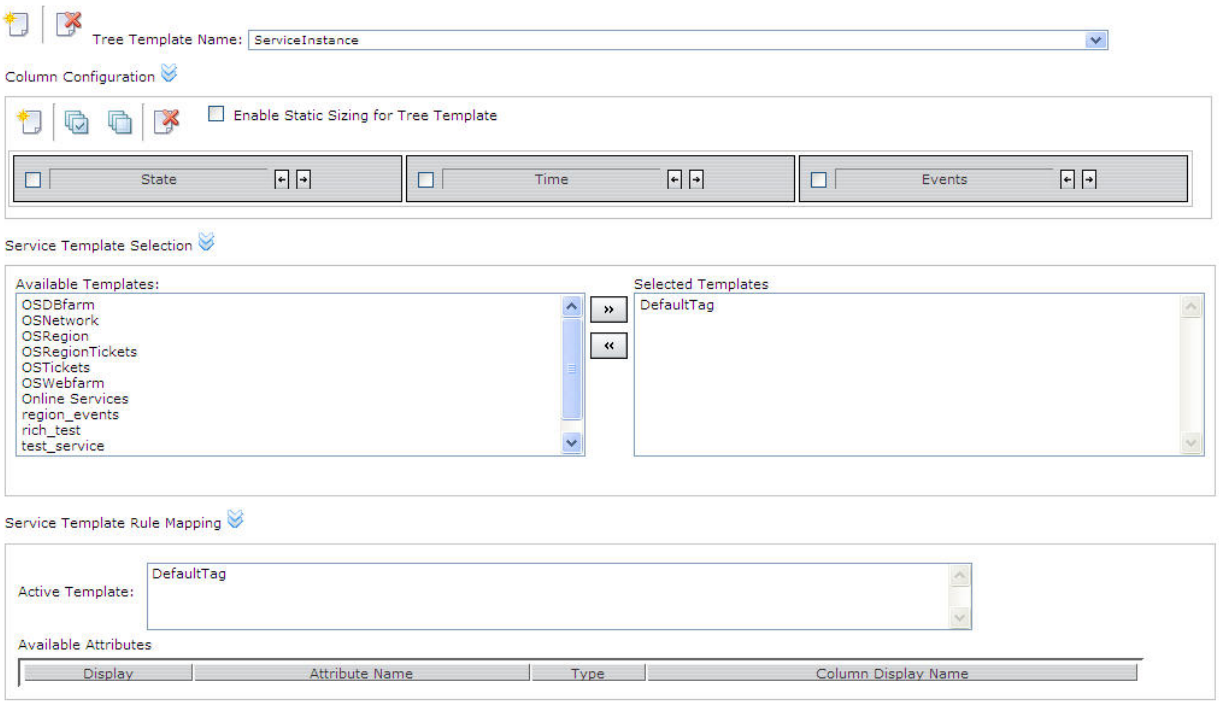


Figure 40. Tree Template Editor

Working with tree templates

This topic describes how to select, add, delete, and display a tree template.

About this task

The Tree Template Name list in the Tree Template Editor lets you specify the tree template you want to edit. You can select, add, and delete tree templates.

Selecting a tree template

About this task

To select a tree template, choose the tree template you want to modify from the **Tree Template Name** drop-down list.

Adding a tree template

About this task

To add a new tree template:

Procedure

1. Click the **Add New Tree Template** button.

   The **Create New Tree Template** window opens.

2. Type the name you want for the new tree template.
3. Click **Apply**.

   The **Create New Tree Template** window closes and the new tree template appears in the **Tree Template Name** drop-down list.

Deleting Tree Template

About this task

To delete a tree template:

Procedure

1. Select the tree template you want to delete from the **Tree Template Name** drop-down list.
2. Click the **Delete Tree Template** button.

Displaying the Tree Template

About this task

To display a custom tree, you need to add a service tree portlet to a page, and select the custom tree template.

Procedure

1. Create or edit a page using the Page Management feature.

   For more information about creating a custom page using the Page Management feature, see the *IBM Tivoli Business Service Manager: Scenarios Guide*.
2. Add a Service Tree portlet to the page:
   a) Click a custom page in the left navigation pane. The page displays on the right.
   b) From the **Select Action** drop-down list, click **Edit Page**.
   c) From the icons on the Service Tree title bar, click the Vertical Split button. The Portlet list displays.
   d) Select **Service Tree** as the **Portlet Entity Title** and click **OK**.
   e) Click **Save**. The Page Management page displays.
   f) Click **Save**. The updated custom page displays.
3. Click **Edit Options** in the title bar and select **Edit Shared Settings** from the menu that displays.
4. Select a tree template from the **Tree Template** drop-down list on the **View** tab.
5. Optional, select a default Starting Instance for the portlet.

What to do next

For more information about how to display a user-defined tree in TBSM, see the *IBM Tivoli Business Service Manager: Scenarios Guide*.

About tree columns

This topic describes tree columns. This topic also describes the Column Configuration toolbar.

Working with tree columns

The TBSM *Tree Template Editor* lets you add, change, and delete the columns in a given tree template. Before you can proceed with this step, you need to select the tree template you want to modify. To see

the changes made to the columns, you must save the tree template and refresh the **Service Navigation** portlet after you have saved the changes.

Column Configuration toolbar

Table 70 on page 205 describes the buttons in the Column Configuration toolbar.

Table 70. Column Configuration toolbar buttons

Button	Button name	Description
	Add New Tree Column	Adds a new tree column.
	Select All	Selects all columns for deletion.
	Deselect All	Clears the selection check box for all columns.
	Delete Selected	Deletes all the selected columns.
	Enable Static Sizing for Tree Template	By selecting the **Enable Static Sizing for Tree Template** checkbox, you can specify the size of each individual column. By default, the checkbox is not selected. Columns are sized initially to fit within the Service Tree boundary. As you resize the columns, those values are applied to the columns anytime you access the instance of the Service Tree portlet. After you select the **Enable Static Sizing for Tree Template** checkbox, click the **Launch Column Sizing Tool** button. From the **Column Width Sizing tool** dialog, you can configure the static size configuration of the columns in either pixels or percentages. **Note:** If the pixel width specified is greater than the width of the portlet, it will not be possible to view or scroll through the rows in the Service Tree properly.

Adding a tree column

This topic describes how to add a tree column.

About this task

To add a tree column, complete the following steps:

Procedure

1. From the Column Configuration section, click the **Add New Tree Column** button.

A blank column is added to the Column Configuration table.

2. Type the name you want to use in the blank field for the new column.
3. Click **OK** to save the new column.

   The new column is saved and the *Tree Template Editor* closes. When you open the *Tree Template Editor*, it shows the new column.

Changing a tree column name and position

This topic describes how to change a tree column name and position.

About this task

You can change the name and the position of a given tree column.

To change the column position, use the arrows next to the column name text field to move the columns left and right.

Changing the column name

About this task

1. To change the name of a column, click the text field for the column name and edit the text.
2. When you are finished editing the text, click **OK** to save your changes.

Deleting columns

This task describes how to delete tree template columns in TBSM.

About this task

You can delete tree template columns by selecting the columns you want to delete and clicking the **Delete Selected** button.

To select and delete columns, complete the following steps:

Procedure

1. To select a single column, click in the check box next to the column name. To select all the columns for deletion, click the **Select All** button.
2. To clear the selection check box for a column, click check box for the selected column. To clear the selection check box for all columns, click the **Deselect All** button.
3. To delete the selected column or columns, click the **Delete Selected** button.
4. Click **OK** to save your changes.

Selecting service templates for a tree template

This task describes how to select service templates for a tree template in TBSM.

About this task

The TBSM *Tree Template Editor* lets you specify the service templates you want to use for your service-tree column data. The `Default Tag` template contains the rules for the default **Services** tree. After you select a service template, you map template rule output values to display in a service tree column.

Before you select the service templates you want to use for the custom-service tree, you need to select the tree template you want to modify.

Service templates and tree templates

This topic describes the relationship between service templates and tree templates. This topic also describes template selection elements.

The TBSM *Tree Template Editor* lets you specify the service templates you want to use for your service-tree column data. The `Default Tag` template contains the rules for the default **Services** tree. After you select a service template, you map template rule output values to display in a service tree column.

Before you select the service templates you want to use for the custom-service tree, you need to select the tree template you want to modify.

Template selection elements

Table 71 on page 207 describes the elements of the Template Selection part of the *Tree Template Editor*.

Table 71. Service Template selection elements

Element	Description
Available Templates list	List of service templates configured on your system.
Select Template button **>>**	Click this button to move service templates to the Selected Templates list.
Deselect Template button **<<**	Click this button to remove service templates from the Selected Templates list.
Selected Templates list	List of service templates you have selected for the service tree column display.

Template rules and tree columns

This topic describes the relationship between template rules and tree columns.

The TBSM *Tree Template Editor* lets you specify the rules you want to use to display data in a given service-tree column. First you select the service template and then you select the rules you want to use to display data in a given service-tree column. You can assign multiple rules to a column and select which one to display. The column displays data from the rule for any service instance that is assigned to the selected service template.

You can display the output for both incoming status and aggregation rules in your custom-service-tree column. However, it is typically best to display the output of a numeric-status rule or a numerical formula rule in your column. When you use these rule types, your column displays a number value that measures the status of the service.

For each level of your service model, you can assign a different rule to your custom-service-tree column. For example, if you want to display a count of trouble tickets by geographical region and by service name, you configure the column to display the output of rules that count the trouble tickets for each service and region.

Adding a template rule to a column

This topic describes how to add a template rule to a column.

About this task

The Active Template list contains the service templates you selected for your custom service tree. To add a rule to a service-tree column:

Procedure

1. In the **Active Template** list, click the service template you want to use for the service-tree column.
2. For each rule you want to use, select the service-tree column for the rule using Table 72 on page 208 as your guide.

Table 72. Available Rules table rows

Column	Description
Display check box	Select this check box to display a column in your custom-service tree. You can assign multiple rules to a column and select which one to display.
Attribute Name	The name of the service-template rule that provides the data for a given column. The output from the selected rule displays in the tree column you select for the rule.
Type	The type of rule. The incoming status and aggregation rules for a template can be used for a service-tree column.
Column Display Name drop-down list	The name of the column that displays the rule output. To have custom column added to this drop-down list, you must add the column. Select the column that will be used to display the data for the rule.

3. Click **Ok** to save the updated tree template configuration.

   TBSM saves the tree-template configuration and the *Tree Template Editor* closes.

   To view the updated Services tree, refresh the Service Navigation portlet.
4. To change the tree-template configuration, click the **Tree Template Editor** button.

Working with reserved column names

This topic describes the default reserved column names, names and color value thresholds for tickets columns, revenue column format, percentage column format, customer column format, baseline column format, and state column format. This topic also has information about changing the default reserved column names.

TBSM includes a set of default reserved column names that either color code or convert the numerical values in the columns. You can find the reserved column name mappings in the policy named `RAD_GetTreeColumnValue.ipl` in the `$TBSM_DASHBOARD_SERVER_HOME/policy` directory.

The behavior of these default reserved column name types is described in Table 73 on page 208.

Table 73. Default reserved column name behavior

Default reserved column name types	Description
Color coding for `Tickets` columns	For Tickets columns, the color of the values displayed in the column changes is based on the numbers in the column. For example, if you have a column named Total Tickets, the numerical values shown in the column are green, orange, or red, depending on the value.
Dollar sign added($)	A dollar sign is added to values representing dollar amounts (Revenue).
Percentage sign added (%)	A percentage sign is added to the value when the column name includes a percentage sign.

Table 73. Default reserved column name behavior (continued)	
Default reserved column name types	**Description**
Strings converted to integers	When values represent totals, strings are converted to integers.
Decimals places restricted	When the value represents certain totals, the decimal point places are restricted.

Each default reserved column name type is described in the sections that follow.

Default names and color value thresholds for tickets columns

The reserved column names in the following list display color-coded values:

- Open tickets
- Closed tickets
- Total tickets - The values are also converted to integers.
- Waiting tickets

The default colors for numerical values in these columns are:

- Green when the value is less than or equal to 10
- Orange when the value is greater than 10 and less than 20
- Red when the value is greater than 20

In a column named `Total Tickets`, for example, by default the numbers in the column would be green when the value is less than 10, orange if the value is greater than 10, and red when the value is greater than 20.

Default revenue column format

The default revenue column names are:

- `New Customer Revenue`
- `Existing Customer Revenue`
- `Total Customer Revenue`

A dollar sign ($) is added to the values that display in these columns and they are converted to integers. The integer is limited to two decimal places. For example, if the rule-output value is `46.424321`, the value displays as $`46.42`.

Default percentage column format

If the column name has a percentage sign (%) in it, then only whole number values display in the column and a percentage sign is appended to the value. For example, if the value is `46.345678` and the column name is `Customer %`, the column displays the value 46%.

Default customer column format

The Customer reserved column names can be like any of the following:

- New Customer
- Tickets per Customer
- Total Tickets

The values in these columns are converted to integers with no decimals.

Default baseline column format

If the column name has the string `Baseline` in it, the color automatically changes based on the value. The value is bolded and the default color is green. For example, if the value is:

- Less than 90, the color is green
- Less than 80, the color is red
- More than or equal to 120, the color is red italic

Default state column format

If the column name has the string `State` in it, the status icons for the `Good`, `Marginal`, and `Bad` status display in the column. If the value is:

- Less than 3, the icon is green (clear)
- Less than 5, the icon is yellow (marginal)
- Greater than or equal to 5, the value is red (bad)

Changing the default reserved column names

For information about how to change these defaults, see the *TBSM Customization Guide*.

Viewing a custom tree

This topic describes how to view a custom tree. This topic also describes how to view a tree for a user-defined tree template.

About this task

To view a custom tree, reopen the page that contains the custom Service Tree portlet. You can also view a custom tree by selecting the **Refresh** button on the portlet toolbar.

Viewing a tree for a user-defined tree template

About this task

To view a user-defined template, you need to create a custom page for TBSM. For information about creating custom pages, see the *TBSM Scenarios Guide*.

Examples for creating custom trees

For examples of how to create custom trees, see the *TBSM Scenarios Guide*.

Chapter 29. Time Window Analyzer view

These topics describe how you can use the Time Window Analyzer view to display service key performance indicators (KPIs) in a time series line graph.

Time Window Analyzer overview

These topics describes the main features of the Time Window Analyzer view.

You can use the Time Window Analyzer view to show service trends over time and in relation to other services, service effecting events and compare it to historical data. When you configure the view you select the following options:

- one or more services where you want to show KPI values for a given time period
- a key performance indicator (KPI) value you want to plot over time. This is a numerical rule that is part of the service's template.
- an Overlay that shows, when the service status changed over the time period.
- other recent time periods showing the same data. The chart plots the data for the recent time period on top of the chart for the most recent time period.

The Time Window Analyzer contains two elements:

- The **Compare** frame lets you select the services you want to view in the chart, and history for those services.
- The chart frame lets you select the KPI you want to plot and an overlay you want to display in the chart.

Compare Services

The **Compare Services** frame lets you specify the services you want to show in your chart. This tab displays by default. You can hide this tab by clicking the **Hide Legend**(◂) button between the chart frame and the **Compare Services** frame. To restore the **Compare Services** frame, click the **Show Legend**(►) button. You can also adjust the **Compare Services** frame by dragging the divider to any size you want.

You can select a service three ways:

1. You can specify services from the **Services** tab using **Add Service** .
2. You can open the Time Window Analyzer in context from the Service Tree, Urgent Service, or Service Viewer portlets using the **Show -> Compare Service Metrics** option from the right-click menu.
3. You add the services you want to plot as part of your user preferences or as part of the default view for the Time Window analyzer:

You can remove a service from the **Compare Services** frame two ways:

1. To remove a service from the graph, deselect the service you want to remove in the list.
2. To remove a service from the list, close and reopen the Time Window Analyzer page. Add back any services you want.

Compare History

The Compare **History** tab lets you show historical data for past time periods for a given service and KPI (rule metric). When you show historical data, the data refresh is does not automatically refresh.

Chart frame

The chart frame lets you select the KPI you want to plot and an overlay you want to display in the chart. The chart plots the KPI and overlay values for the services you select.

Default view

By default the chart shows a static view of the data. If you select a non-static chart in your portlet preferences, the data is refreshed periodically according to your portlet preferences. The time to each refresh is shown in the lower-right, below the chart.

Time window period

The time window set in the portlet preferences displays above the chart. For example: `8/10/09 5:28 AM to 1:28 PM`

Zooming

To zoom-in to a time period within the time window, click and drag over the time interval you want to magnify. The chart shows only the time period you select, and Overview window opens to show the entire time window, with the magnified area highlighted. To cancel the magnification, close the Overview window.

When you zoom in, the data refresh is suspended.

Key performance indicator (KPI)

You can plot KPI values for each service in the **Compare Services** frame. Select the KPI you want from the drop-down list on the left side of the chart frame. The KPI can be any service template rule that outputs a numerical value. You can customize TBSM to show any numerical rule as a KPI. For more information, see the TBSM customization guide.

By default, you can only view status change values for a given service. The status change values show the event severity value from the Netcool/OMNIbus alerts.status table. The values are:

Table 74. Event severity values

Severity value	Description
0	Clear (green)
1	Indeterminate (purple)
2	Warning (blue)
3	Minor (yellow)
4	Major (orange)
5	Critical (red)

Overlay

The Overlay option lets you display a bar over the graph that contains status-change indicators for a given service. Select the **Overlay** check box to display the overlay data for a given service. To see an overlay for another service, click on the service you want in the **Services** tab. The service you select is highlighted with a bar similar to the overlay bar. You can only show an overlay for one service at a time.

Related tasks

Selecting services
This task describes how you select services for the Time Window Analyzer.

Showing historical comparison for a service
This task shows how you can compare performance indicators from past time periods to the current time period.

Graphing a performance indicator for services
This task describes how to plot a key performance indicator (KPI rule) value over time for a service.

Adding a Time Window Analyzer Portlet to a Dashboard Application Service Hub page

This topic describes how to add a Time Window Analyzer to a page.

Selecting services

This task describes how you select services for the Time Window Analyzer.

Before you begin
You need to do be logged in as a user with authority to view and modify the **Time Window Analyzer** page.

About this task
You can select services to compare in the Time Window Analyzer graph three ways:

Procedure

1. To select a service from the Time Window Analyzer, select **Time Window Analyzer** from the Availability tasks in the left navigation frame and add services to compare by hand.
2. Open the Time Window Analyzer in context of a service from the Service Tree, Urgent Service, or Service Viewer portlets.

   Note: If you select a service with no available data, the system displays the message :

   ```
   CTGBH0100W

   No historical metric data is available.

   This may simply mean that no data has ever been collected for DisplayName
   [ServiceName].
   ```

   More time may be needed to collect the data, or you may need to configure the Time Window Analyzer to collect data for the service as described in the TBSM *Administrator's Guide*.

   The Time Window Analyzer window opens and shows the service you specified in the **Services** tab.
3. You can specify the default services you want to compare in Time Window Analyzer. You cannot set the service context preferences specific to a user in DASH. DASH will allow global preferences, provided the users belong to same group. When you create a custom page that includes a Time Window Analyzer portlet, you may want to set the default services for the portlet.

Related concepts
Time Window Analyzer preferences
These topics describe the user and default preferences you can set for Time Window Analyzer.

Time Window Analyzer overview
These topics describes the main features of the Time Window Analyzer view.

Opening Time Window Analyzer for a service

This task describes how you open the Time Window Analyzer from a given TBSM service.

Before you begin
You need to do be logged in as a user with authority to view and modify the **Time Window Analyzer** page.

About this task
To open the Time Window Analyzer from a service from the Service Tree, Urgent Services, or Service Viewer portlets:

Procedure

1. Right-click on the service you want to plot in the chart.
2. Select **Show ->Compare Service Metrics** from the menu that appears.

The Time Window Analyzer window opens and shows the service you specified in the **Services** tab.

3. To add more services to the graph, click the **Add Service** button.

Adding services to compare in the graph

This task describes how you add services to compare in the Time Window Analyzer.

Before you begin

You need to do be logged in as a user with authority to view and modify the **Time Window Analyzer** page.

About this task

To add a service to compare in a graph:

Procedure

1. From the Dashboard Application Service Hub navigation frame, select: **Availability -> Time Window Analyzer**.
2. Click on the **Add Service** button.
3. From the **Search for Service** window that opens, type all are part of the service name you want to view and click **Search** to display a list of matching service names.

   Note: This value is not the service display name, but the actual service name in the TBSM database.

   The service displays on the **Compare Services** list.

Results

The results are displayed in a list (up to 10). Scroll to the item you want, and then click it. You might need to use the **More** button to see additional services.

What to do next

You can remove a service from the **Compare Services** list two ways:

1. To remove a service from the graph, deselect the service you want to remove in the list.
2. To remove a service from the list, close and reopen the **Time Window Analyzer** page. Add back any services you want.

Graphing a performance indicator for services

This task describes how to plot a key performance indicator (KPI rule) value over time for a service.

Before you begin

You need to do be logged in as a user with authority to view and modify the **Time Window Analyzer** page.

About this task

In this task, you select the service, rule, and overlay values you want to plot in your graph.

Procedure

1. Add the services you want to analyze to the Time Window Analyzer **Services** tab.
2. Select a KPI rule from the left drop-down list. This KPI rule can be any numeric rule from the service template assigned to the services you selected.
3. Click the check box for the service or services you where you want to plot data.

   The data values for the selected service are plotted on the graph.
4. To hide the data for a service, click the check box for the service or services where you plotted the data previously.

5. To display an overlay value on the graph, select the **Overlay** check box. By default, you can display the **Status Changes** overlay for the service you select.

   For each status change, an icon displays in the overlay bar.

Related concepts

Time Window Analyzer preferences
These topics describe the user and default preferences you can set for Time Window Analyzer.

Time Window Analyzer overview
These topics describes the main features of the Time Window Analyzer view.

Showing historical comparison for a service

This task shows how you can compare performance indicators from past time periods to the current time period.

Before you begin

You need to do be logged in as a user with authority to view and modify the **Time Window Analyzer** page.

About this task

In this task, you select the service, rule, overlay time period values you want to plot in your graph.

Procedure

1. Add the services you want to analyze to the Time Window Analyzer **Services** tab.
2. Click the check box for the service or services you where you want to plot data.
3. Select a KPI rule from the left drop-down list. This KPI rule can be any numeric rule from the service template assigned to the services you selected.
4. To display an overlay value on the graph select the Overlay check box. You can plot the overlay values:

   Change Events (Occurrences). Value2, Value3.

   For each change event, you can show the event or events associated with the status change. If there are multiple events, the event indicator is circled.
5. Click the **History** tab.
6. Select the single service you want to compare from the services list.
7. Select check box for the time periods you want to compare to the current time period.

   The data points for each time period are plotted on the chart.

Related concepts

Time Window Analyzer preferences
These topics describe the user and default preferences you can set for Time Window Analyzer.

Time Window Analyzer overview
These topics describes the main features of the Time Window Analyzer view.

Adding a Time Window Analyzer Portlet to a Dashboard Application Service Hub page

This topic describes how to add a Time Window Analyzer to a page.

Before you begin

You need to have the administrator user roles the let you create pages in Dashboard Application Service Hub. For example, the **administrator** user has the roles you need to create pages.

For more information on pages and portlets, see the help for these pages:

To manage user roles, from the left navigation pane, click Users and Groups → Roles. For more information, see the console help.

To manage pages, from the left navigation pane, click Settings → Pages. For more information, see the console help.

To manage portlets, from the left navigation pane, click Settings → Portlets. For more information, see the console help.

About this task

To add a Time Window Analyzer portlet to a page, you need to create a new page and add a Time Window Analyzer portlet. You may need to add a new page to show the Time Window Analyzer with other portlets on the same page. After you create the page, you can set the preferences for the Time Window Analyzer such as the portlet title, the services you want to compare, the type of chart you want, and the refresh rate.

Procedure

1. Open **Settings -> Pages** in the left navigation frame. You can also click the **+ (Create New Work Page)** in the page tabs to open **Page Settings**.
2. Click **New Page**.
3. Select the **Time Window Analyzer** portlet from the Choose a Portlet page and click **OK**.
4. To save the page, click **Save**.
5. Click **Role with Access to this Page** and add the roles you want. For example, add all the roles starting with `tbsm`.
6. Click **Save**.
7. To set your the preferences, click on **Edit options** in the title bar and select either **Personalize** or **Edit Shared Settings** from the menu that appears. **Personalize** controls the settings for the current user. **Edit Shared Settings** controls the settings for all users. Otherwise, the options are the same.

   For example: You can use the **Portlet Preference** tab options to change the display mode size or set how the chart responds to new service context data using the **New Context Handling** settings.
8. When you have finished setting your preferences, click **Save**.

Related concepts

Time Window Analyzer preferences
These topics describe the user and default preferences you can set for Time Window Analyzer.

Time Window Analyzer overview
These topics describes the main features of the Time Window Analyzer view.

Time Window Analyzer preferences

These topics describe the user and default preferences you can set for Time Window Analyzer.

To set your the preferences, click on **Edit options** in the title bar and select either **Personalize** or **Edit Shared Settings** from the menu that appears. **Personalize** controls the settings for the current user. **Edit Shared Settings** controls the settings for all users. Otherwise, the options are the same.

Press **Save** to save your changes and return to the analyzer. Press **Cancel** to discard any changes and return to the analyzer. You can edit these types of preferences:

- **Window**: This tab allows you to customize aspects of the analysis time window
- **Chart**: This tab allows you to define various visual aspects of the time window chart.
- **Context**: This tab allows you to define the analysis context to use each time this view instance opens.
- **Portlet:** This tab allows you to define items for this particular portlet instance.

Related tasks

Selecting services
This task describes how you select services for the Time Window Analyzer.

Showing historical comparison for a service
This task shows how you can compare performance indicators from past time periods to the current time period.

Graphing a performance indicator for services
This task describes how to plot a key performance indicator (KPI rule) value over time for a service.

Adding a Time Window Analyzer Portlet to a Dashboard Application Service Hub page
This topic describes how to add a Time Window Analyzer to a page.

Window preferences

Window preference for the Time Window Analyzer.

Purpose

These preferences let you to customize aspects of the analysis time window.

Preferences

Set preferences for the window. In any drop-down lists, the default setting is **bold** and the current setting is underlined.

Time Window Size
Specifies the time interval shown in the chart. You specify hours or days.

Maximum Services
Specifies the maximum number of services shown in the **Services** tab.

Maximum Historical Time Slots
Specifies the maximum number of time slots shown in the **History** tab.

Data Display Mode
Select one of these options from the drop-down list:

- One-time Snapshot shows a static picture of the data for the service you select.
- Running History (past to now) shows data from the time window until the current time. The data is updated according to the time window you specified in the **Check For New Data** setting.
- Running Accumulation (now forward) shows data from the time you select the service and onward. The data is updated according the time you specify in the **Check For New Data** setting.

Check for New Data
Specifies how often the chart checks for the availability of new data.

Chart preference

Chart preferences for the Time Window Analyzer.

Purpose

These preferences set the line width and line type in the chart.

Preferences

Set these parameters for the lines on your chart. In any drop-down lists, the default setting is **bold** and the current setting is underlined.

Line Width
Use the slider bar to change the width of the line in the chart. Lower numbers equal thinner lines.

Default: 2

Line Type

You display a line with data point markers or simple line with no data point markers.

When you select **Visible data point markers and tooltips**, a point displays for each instance of data plotted in the chart. In addition, tooltips show information about how the data point relates to the other data points in the chart. When you point to a data point, you see this information:

The title of the tooltip is the service name.

Table 75. Tooltips for data points

Tooltip name	Value
Before.	The value and time stamp for the preceding data point.
This	The value and time stamp for the data point.
After	The value and time stamp for the subsequent data point.
Point	The numerical position of the data point in relation to all the data points in the chart. From example: `15 of 21`.

Default: No data point markers.

Context preferences

Context preferences for the Time Window Analyzer.

Purpose

If you open a page that includes the Time Window Analyzer Portlet from the task navigator, the chart shows data for the services specified in the Context preferences. Specify the services and rules you want to use for your service metrics.

Preferences

Setting service context

Use the **Browse** button to search for services. Repeat this step for each service you want to add.

The next time you open the Time Window Analyzer from the task navigator, the **Services** tab displays the service or services you specified.

To remove a service, click the **Clear** button.

Metric Initially Displayed

By default, the chart shows the `OverallAttribute` status metric for the service.

To initially display a different metric. enter the template and rule name in the **Metric Initially Displayed** field in the format:

```
template.rulename
```

For example, for the `WAS Method Invocation Rate` rule in the `BSMAccelerator_WASMetrics` service template, enter:

```
BSMAccelerator_WASMetrics.WAS Method Invocation Rate
```

To revert to the default, delete the metric.

Portlet preferences

Portlet preference for the Time Window Analyzer.

Purpose

These preferences specify a unique title, display mode, and context for your Time Window Analyzer portlet.

Preferences

Set these preferences for the portlet. In any drop-down lists, the default setting is **bold** and the current setting is underlined.

Title
Specifies the title for the portlet.

User Interface Display Mode
Select the display type you want from these drop-down list options:

Standard size
The full-size portlet. Default.

Reduced size
Reduces heading font size and width of the **Compare** area

New Context Handling

Controls how the chart responds to new service context data when the page is launched from another page (launchPage), or when a service is clicked in a Service Tree or other portlet on the same page (NodeClickedOn). By default, the chart replaces the service data when it receives new service context data (**New context replaces current** setting).

When you first open a page that includes a Time Window Analyzer portlet, the chart shows data for a service or the services as follows:

- When you right-click on a service and select the **Compare Service Metrics** option, the chart shows data for the service you clicked. The context of the chart is for a single service, which is the default setting: **New context replaces current**.
- If you open a page that includes the Time Window Analyzer Portlet from the task navigator, the chart shows data for the services specified in the **Context Preferences**.

Once the portlet is open, these options specify how the chart responds when the new context data is sent to the portlet. Select one of these options from the drop-down list:

Ignore new context
The chart does not respond to any new context data from other portlets on the same page. However, if you launch from another page, the context data changes.

New context replaces current.
Replace the current data with data from the new service context. Default setting.

New context added to current
Add the new service context data to the current service data in the chart. Services accumulate until the **Maximum Services** limit is reached. The **Maximum Services** setting is on the **Window** tab.

Chapter 30. Reference

Reference information is organized to help you locate particular facts quickly.

TBSM-specific projects

The Tivoli Business Service Manager version of Netcool/Impact has additional Tivoli Business Service Manager specific projects.

- **TBSM_BASE** project contains all the predefined Netcool/Impact data sources, data types, services, and policies, that are specific to Tivoli Business Service Manager. Modifications to any of the items in this project must be done with caution.
- **TBSM** project contains all Netcool/Impact data sources, data types, services, and policies that are created using the Tivoli Business Service Manager configuration interface.
- **TBSM_SAMPLES** project contains sample policies used to interact with the Services Component Registry.
- **ForImpactMigration** project contains the data sources and data types necessary for a remote Impact Server to send events using the **PassToTBSM** function to Tivoli Business Service Manager. To send events to Tivoli Business Service Manager from a remote Impact Server, you must export the **ForImpactMigration** project from the Tivoli Business Service Manager server and import it into the Impact Server. For more information about **PassToTBSM**, see the *Netcool/Impact Solutions Guide.*

Event enrichment solution

Event enrichment is the process by which Netcool/Impact monitors an event source for new events, looks up information related to them in an external data source and then adds the information to them.

An event enrichment solution consists of the following components:

- A data model that represents the data you want to add to events
- An OMNIbus event reader service that monitors the event source
- One or more event enrichment policies that look up information related to the events and add the information to them

For a sample event enrichment solution, see “Event enrichment tutorial” on page 225.

Creating the event source

When you have the connection information for the ObjectServer, you can create the event source using the Dashboard Application Service Hub GUI. You create the event source in the same way that you create any other data source.

Table 76 on page 221 shows the configuration properties for an event source.

Table 76. Event Source Configuration Properties

Property	**Description**
Data Source Name	Unique name for the event source.
User Name	Valid user name for the event source.
Password	Valid password for the event source.

Table 76. Event Source Configuration Properties (continued)	
Property	**Description**
Primary Source Host Name	Host name or IP address of the system where the event source is located.
Primary Port	Port used by the event source. Default port is 4100.
Backup Host Name	Host name or IP address of the system where the backup event source is located. Optional.
Backup Port	Port used by the backup event source. Optional. Default port is 4100.

To create the event source:

1. In the navigation tree, expand the **Impact Menu** click **Data Model** to open the **Data Model tab**.
2. Select a cluster and a project from the **Cluster** and **Projects** lists.
3. Click the **New Data Source** icon in the toolbar and select the type of data source you want to create from the menu.

   The data source window opens.
4. Enter the required configuration properties.
5. Click the **Save** icon on the tab, to create the event source.

After you have created the event source, you can then create and configure an associated event reader service.

OMNIbus event reader service

OMNIbus event readers are services that monitor a Netcool/OMNIbus ObjectServer event source for new, updated, and deleted alerts and then runs policies when the alert information matches filter conditions that you define.

The event reader service uses the host and port information of a specified ObjectServer data source so that it can connect to an Objectserver to poll for new and updated events and store them in a queue. The event processor service requests events from the event reader. When an event reader discovers new, updated, or deleted alerts in the ObjectServer, it retrieves the alert and sends it to an event queue. Here, the event waits to be handled by the event processor.

You configure this service by defining a number of restriction filters that match the incoming events, and passing the matching events to the appropriate policies. The service can contain multiple restriction filters, each one triggering a different policy from the same event stream, or it can trigger a single policy.

You can configure an event reader service to chain multiple policies together to be run sequentially when triggered by an event from the event reader.

Important: Before you create an OMNIbus event reader service, you must have a valid ObjectServer data source to which the event reader will connect to poll for new and updated events.

OMNIbus event reader configuration

You can configure the following properties of an OMNIbus event reader.

- Event reader name
- ObjectServer event source you want the event reader to monitor
- Interval at which you want the event reader to poll the ObjectServer
- Event fields you want to retrieve from the ObjectServer

- Event mapping
- Event locking
- Order in which the event reader retrieves events from the ObjectServer
- Start up, service log, and reporting options

OMNIbus event reader service General Settings tab

Use this information to configure the general settings of the OMNIbus event reader service.

Table 77. EventReader service - general settings tab

Table Element	Description
Service name	Enter a unique name to identify the service.
Data Source	Select an OMNIbusObjectServer data source. The ObjectServer data source represents the instance of the Netcool/OMNIbus ObjectServer that you want to monitor using this service. You can use the default ObjectServer data source that is created during the installation, defaultobjectserver.
Polling Interval	The polling interval is the interval in milliseconds at which the event reader polls the ObjectServer for new or updated events. Select or type how often you want the service to poll the events in the event source. If you leave this field empty, the event reader polls the ObjectServer every 3 seconds (3000 milliseconds).
Restrict Fields: Fields	You can complete this step when you have saved the **OMNIbusEventReader** service. You can specify which event fields you want to retrieve from the ObjectServer. By default, all fields are retrieved in the alerts. To improve OMNIbus event reader performance and reduce the performance impact on the ObjectServer, configure the event reader to retrieve only those fields that are used in the corresponding policies. Click the **Fields** button to access a list of all the fields available from the selected ObjectServer data source. You can reduce the size of the query by selecting only the fields that you need to access in your policy. Click the **Optimize List** button to implement the changes. The **Optimize List** button becomes enabled only when the **OMNIbusEventReader** service has been saved.
Startup: Automatically when server starts	Select to automatically start the service when the server starts. You can also start and stop the service from the GUI.
Service log: Write to file	Select to write log information to a file.
Collect Reports: Enable	Select to enable data collection for the Policy Reports.

Table 77. EventReader service - general settings tab (continued)	
Table Element	**Description**
Clear State: Clear	When you click the **Clear State** button, the `Serial` and `StateChange` information stored for the event reader is reset to 0. The event reader retrieves all events in the ObjectServer at startup and places them in the event queue for processing. If the event reader is configured to get updated events, it queries the ObjectServer for all events where `StateChange >= 0`. Otherwise, it queries the ObjectServer for events where `Serial > 0`. You can use the **Clear State** button only to clear the event reader state when the service is stopped. Clicking the button while the service is running does not change the state of the event reader.
Clear Queue: Clear	Click to clear unprocessed events.

Database event reader configuration window - general settings

Use this information to configure the general settings of the database event reader.

Table 78. Database event reader configuration window - General Settings tab	
Window element	**Description**
Service name	Enter a unique name to identify the service.
Data Source	Select an external data source from the list. The data source must have a field that is guaranteed to be incremented every time a new record is added to avoid rereading the entire table every time the data source is accessed. If you want to use the `GetUpdates` function in a policy for this data source, the table also must have a time stamp field that is automatically populated when an insert or update occurs.
Data Type	After you select a data source, the system populates the data type field with a list of data types created in Netcool/Impact corresponding to that particular data source. Select a data type from the list.
Polling Interval	Select or enter a polling time interval to establish how often you want the service to poll the events in the event source. The polling time selections are in milliseconds and the default value is `3000` milliseconds
Restrict fields	Click **Fields** to access a selection list with all the fields that are available from the selected data source. You can reduce the size of the query by selecting only the fields that you need to access in your policy.
Startup: Automatically when server starts	Select to automatically start the service when the server starts. You can also start and stop the service from the GUI.
Service log: Write to file	Select to write log information to a file.

Table 78. Database event reader configuration window - General Settings tab (continued)	
Window element	**Description**
Clear State	When you click **Clear**, the internally stored value for the **Key** field and **Timestamp** field are reset to 0. This causes the event reader to retrieve all events in the data source at startup and place them in the event queue for processing. If the event reader is configured to get updated events, it requires the **Timestamp** field along with the **Key** field. The **Timestamp** field must point to a column in the table which is automatically populated with a timestamp when an insert or update occurs. The **Key** field must point to a column which uniquely identifies a row (it does not have to be an automatically incremented field). However, when the **Actions Get updated events** check box in the Event Mapping tab is not selected, you do not have to configure the **Timestamp** field. The **Key** field *MUST* in this case be an automatically incremented numeric field. You can only use **Clear State** to clear the event reader state when the service is stopped. Clicking **Clear** while the service is running does not change the state of the event reader.
Clear Queue	Click **Clear** to enable the database event reader to delete unprocessed events that it has fetched from an SQL data source.

Event enrichment tutorial

The goal of this tutorial is to develop an event enrichment solution to enhance the value of an existing Netcool/Impact installation.

This solution automates common tasks performed manually by the network operators and helps to integrate related business data with alerts in the ObjectServer.

Tutorial overview

This tutorial uses a sample environment that provides the background for understanding various event enrichment concepts and tasks.

The environment is a network operations center for a large enterprise where the company has installed and configured Netcool/OMNIbus and is currently using it to manage devices on its network. The sample environment is a scaled down representation of what you might actually find in a real world operations center. It contains only the network elements and business data needed for this tutorial.

This tutorial leads you through the following steps:

- Understanding the Netcool/Impact installation
- Understanding the business data
- Analyzing the workflow in the environment
- Creating a project
- Setting up a data model
- Setting up services
- Writing an event enrichment policy
- Configuring the OMNIbus event reader to run the policy
- Running the complete solution

Understanding the Netcool/Impact installation

The first step in this tutorial is to understand the current Netcool installation.

Generally, before you start developing any Netcool solution, you must find out which products in the Netcool suite you installed and which devices, systems, or applications are being monitored in the environment.

The Netcool installation in the sample environment consists of Netcool/OMNIbus and a collection of probes that monitor devices on the network. This installation uses two instances of an ObjectServer database named NCOMS that is set up in a backup/failover configuration. These ObjectServers are located on host systems named NCO_HOST_01 and NCO_HOST_02, and run on the default port of 4100.

The probes in this installation monitor various network devices. The details of the devices are not important in this tutorial, but each probe sends the basic set of alert fields to the ObjectServer database, including the Node, Summary, Severity, AlertKey, and Identifier fields.

Understanding the business data

The next step in this tutorial is to understand the location and structure of the business data in your environment.

In the sample environment, the company uses instances of the Oracle database to store network inventory information, customer service information, and general organizational information about the business.

The information that you want to use is stored in two databases named ORA_01 and ORA_02. ORA_01 is a network inventory database that stores information about the devices in the network, including their technical specification, facility locations, and rack numbers. ORA_01 is located on a system named ORA_HOST_01. ORA_02 is a database that contains information about the various departments in the business. ORA_02 is located on a system named ORA_HOST_02. They both run on port 1521

Analyzing the workflow

After you find the location and structure of the business data, the next step is to analyze the current event management workflow in your environment.

The tutorial work environment is a network operations center. In this center, a number of operators are on duty at all times. They sit in an open work area and each one has access to a console that displays a Netcool/OMNIbus event list. On large projector screens on one wall of the operation center are large map visualizations that provide geographical views into the current network status.

As alerts flow to the ObjectServer from the various Netcool probes and monitors that are installed in the environment, they are displayed in the event lists available to the operators. Depending on the severity of the alerts, the operators manually perform a set of tasks using the event list tools, third-party applications, and typical office tools like cell phones and email.

For the sake of this tutorial, we assume that, among other tasks, the operators perform the following actions for each high severity alert. The operators:

- Manually acknowledge the alert using the event list.
- Use an in-house database tool to find information about the device causing the alert. This tool runs a query against the network inventory database and returns technical specifications, the location, and other information.
- Use another in-house tool to look up the business department being served by the device that caused the alert.
- If the business department is part of a mission critical business function, they increase the severity of the alert and update it in the ObjectServer database.

The operators might perform other actions, like looking up the administrators on call at the facility where the device is located and contacting them by phone or pager. After the problem that caused the alert is addressed, the operators might also record the resolution in a problem log and delete the alert from the ObjectServer. For this tutorial, however, only use the workflow tasks listed.

Creating the project

After you finish analyzing the workflow, the next step is to create a project in the Dashboard Application Service Hub GUI.

About this task

You can use this project to store the data model, services, and policies that are used in this solution. The name of this project is `NCI_TUT_01`.

Procedure

1. Open the Dashboard Application Service Hub in a web browser and log in.
2. In the navigation tree, expand the **Impact Menu** click one of the links, for example **Data Model**, to view the project and cluster selection lists on the **Data Model** tab.
3. Select a cluster from the **Cluster** list. From the **Project** list, select **Global**.
4. Click the **New Project** icon on the toolbar to open the **New Project** window.
5. Use the **New Project** window to configure your new project.
6. In the **Project Name** field, type `NCI_TUT_01`.
7. Click **OK** then click **Close**.

Setting up the data model

After you create a project for this tutorial, the next step is to set up a Netcool/Impact data model.

This data model consists of the event sources, data sources, and data types that are required by the event enrichment solution. It also consists of a dynamic link that is used to define the relationship between the data types.

You use the Dashboard Application Service Hub GUI to perform all the tasks in this step.

To set up the data model, you perform the following tasks:

- Create the event source
- Create the data sources
- Create the data types
- Create the dynamic link

Creating the event source

The first task in setting up the data model is to create the event source. As you learned when you investigated the details of the Netcool installation, the example environment has one event source, an ObjectServer named `NCOMS`.

About this task

Because you want to tap into the alerts that are stored in this ObjectServer, you must create an event source that represents it in Netcool/Impact.

An event source is a special type of data source that Netcool/Impact can use to represent a physical source of event data in the environment. Since your source of event data is an ObjectServer database, you must create an ObjectServer data source and configure it with the connection information you discovered when you investigated the details of the Netcool installation.

To create the event source:

Procedure

1. In the navigation tree, expand the **Impact UI** click **Data Model** to open the **Data Model** tab.
2. Select a cluster from the **Cluster** list. From the **Project** list, select **NCI_TUT_01**.
3. Click the **New Data Source** icon and select **ObjectServer** from the list.

The **New Data Source** opens.

4. Type NCOMS in the **Data Source Name** field.
5. Type the name and password of an ObjectServer user in the **Username** and **Password** fields.
6. Type NCO_HOST_01 in the **Primary Host Name** field.
7. Type 4100 in the **Primary Port** field.
8. Click **Test Connection** to test the ObjectServer connection.
9. Type NCO_HOST_02 in the **Backup Host Name** field.
10. Type 4100 in the **Backup Port** field.
11. Click **Test Connection** to test the ObjectServer connection.
12. Click **OK**.

Creating the data sources

The next task in setting up the data model is to create the data sources.

About this task

As you learned when you discovered the location and structure of the business data in your environment, the data you want to use in this solution is in two Oracle databases named ORA_01 and ORA_02. Since you want to access these databases, you must create a data source that corresponds to each one.

To create the data sources:

Procedure

1. In the navigation tree, expand the **Impact Menu**, click **Data Model** to open the **Data Model** tab.
2. Click the **New Data Source** icon and select **Oracle** from the list.
   The **New Data Source** window opens.
3. Type ORACLE_01 in the **Data Source Name** field.
4. Type an Oracle user name and password in the **Username** and **Password** fields.
5. Type ORA_HOST_01 in the **Primary Host Name** field.
6. Type 1521 in the **Primary Port** field.
7. Type ORA_01 in the **SID** field.
8. Click **Test Connection** to test the ObjectServer connection.
9. Click **OK**.

Results

Repeat these steps to create another data source that corresponds to the ORA_02 database. Name this data source ORACLE_02.

Creating the data types

The next task in setting up the data model is to create the data types.

About this task

As you learned when you discovered the location and structure of the business data in your environment, the data that you want to use is contained in two tables.

The first table is called Device and is in the ORA_01 database. This table contains information about each device on the network. Columns in this table include Hostname, DeviceID, HardwareID, Facility, and RackNumber.

The second table is called Department and is in the ORA_02 database. This table contains information about each functional department in the business. Columns in this table include DeptName, DeptID, and Location.

Since you want to access the data in both of these tables, you must create a data type for each. Name these data types `Device` and `Department`.

To create the data types:

Procedure

1. In the navigation tree, expand the **Impact Menu** click **Data Model** to open the **Data Model** tab.
2. Select `ORACLE_01` from the data sources list.
3. Click the **New Data Type** icon.

   A new Data Type Editor tab opens.
4. Type `Device` in the **Data Type Name** field.
5. Select `ORACLE_01` from the **Data Source Name** drop down menu.
6. Ensure that the **Enabled** check box is selected. It is selected by default.
7. Scroll down the Data Type Editor tab so that the **Table Description** area is visible.
8. Select `Device` from the **Base Table** list.
9. Click **Refresh**.

   Netcool/Impact queries the Oracle database and populates the **Table Description** browser with the names of each column in the `Device` table.
10. Specify that the `DeviceID` field is the key field for the data type by selecting the **Key** option in the **DeviceID** row.
11. Select `Hostname` from the **Display Name Field** list.
12. Click **Save** in the Data Type Editor tab.
13. Click **Close** in the Data Type Editor tab.

Results

Repeat these steps to create another data type that corresponds to the `Department` table in the `ORA_02` database. Name this data type `Department`.

Creating a dynamic link
The next step is to create a dynamic link between the `Device` and `Department` data types.

About this task

One property of the business data that you are using in this solution is that there is a relationship between devices in the environment and departments in the business. All the devices that are located in a certain facility serve the business departments in the same location. You can make this relationship part of the data model by creating a dynamic link between the `Device` and `Department` data types. After you create the dynamic link, you can use the the `GetByLinks` function to traverse it within a policy.

In this relationship, `Device` is the source data type and `Department` is the target data type. When you create the link between the two data types, you can define it using the following syntax:

```
Location = '%Facility%'
```

This filter tells Netcool/Impact that `Device` data items are linked to `Department` data items if the value of the `Location` field in the `Department` is equal to the value of the `Facility` field in the `Device`.

To create the dynamic link:

Procedure

1. In the navigation tree, expand the **Impact Menu**, click **Data Model** to open the **Data Model** tab.
2. Click the name of the `Device` data type.

A new Data Type Editor tab opens in the Main Work panel of the GUI. This editor displays configuration information for the Device data type.

3. Select the **Dynamic Links** tab in the editor.

   The **Links From This Data Type** area opens in the editor.

4. Click the **New Link By Filter** button to open the Link By Filter window.
5. Select Department from the **Target Data Type** list.
6. In the **Filter ...** Field, type the filter string that defines the relationship between the Device and Department list. As noted in the description of this task above, the filter string is Location = '%Facility%'. This means that you want Device data items to be linked to Department data items if the Location field in the Department is the same as the Facility field in the Device.
7. Click **OK**.
8. Click the **Save** button in the Data Type Editor tab.
9. Click the **Close** button in the Data Type Editor tab.

Reviewing the data model
Verify the data model data in the **Impact UI**.

Setting up services

The next step in this tutorial is to set up the OMNIbus event reader required by the solution.

Creating the event reader
The OMNIbus event reader for this solution must check the NCOMS ObjectServer every 3 seconds and retrieve any new events.

Procedure

1. In the navigation tree, expand the **Impact Menu**, click **Services** to open the **Services** tab.
2. Click the **Create New Service** icon and select **OMNIbusEvent Reader** from the list.
3. Type **TUT_READER_01** in the **Service Name** field.
4. Select **NCOMS** from the **Data Source**list.
5. Type 3000 in the **Polling Interval** field.
6. Select the **Startup** option. This option specifies whether the service starts automatically when you run Netcool/Impact.
7. Click **OK**.

Reviewing the services
To view the created service, check the services tab in the **Impact UI**.

Writing the policy

After you set up the OMNIbus event reader service, the next step is to write the policy for the solution.

This policy is named EnrichEvent and it automatically performs the tasks that you discovered when you analyzed the workflow in the environment.

You can use the EnrichEvent policy to complete the following tasks:

- Look up information about the device that is causing the alert.
- Look up the business departments that are served by the device.
- If one of the business departments is part of a mission critical business function, the policy increases the severity of the alert to critical.

This section assumes that you already know how to create, edit, and save a policy using the policy editor tools in the Impact GUI. For more information about these tools, see the User Interface Guide.

Looking up device information

The first task that you want the policy to perform is to look up device information that is related to the alert in the network inventory database.

About this task

Specifically, you want the policy to retrieve technical specifications for the device that is causing the alert, and information about the facility and the rack number where the device is located.

To do this, the policy must perform a SELECT at the database level on the table that contains the device data and return those rows that are related to the incoming alert. Viewed from the data model perspective, the policy must get data items from the `Device` data type where the value of the `Hostname` field is the same as the value of the `Node` field in the alert.

To retrieve the data items, you type the following code into the Netcool/Impact policy editor tab:

```
DataType = "Device";
Filter = "Hostname = '" + @Node + "'";
CountOnly = False;

MyDevices = GetByFilter(DataType, Filter, CountOnly);
MyDevice = MyDevices[0];

If (Length(MyDevices) < 1) { Log("No matching device found."); }
If (Length(MyDevices) > 1) { Log("More than one matching device found."); }
```

Here, `GetByFilter` is retrieving data items from the `Device` data type where the value of the `Hostname` field is equal to the value of the Node field in the incoming alert. The data items are stored in an array named `MyDevices`.

Although `GetByFilter` is able to return more than one data item in the array, you only expect the array to contain one data item in this situation, as each device in the database has a unique `Hostname`. The first element of the `MyDevices` array is assigned to the `MyDevice` variable so that `MyDevice` can be used as shorthand later in the policy.

Because you want to retrieve only one data item from the data type, the policy also prints error messages to the policy log if `GetByFilter` retrieves less than or more than one.

Looking up business departments

The next task that you want the policy to perform is to look up the business departments that are served by the device that caused the alert.

About this task

When you set up the data model for this solution, you created a dynamic link. This link defined the relationship between the devices in the environment and departments in the business. To look up the business departments that are served by the device, the policy must take the data item that it previous retrieved from the `Device` data type and traverse the links between it and the `Department` data type.

To retrieve the `Department` data items that are linked to the `Device`, type the following text into the policy editor below the code you entered previously:

```
DataTypes = {"Department"};
Filter = NULL;
MaxNum = 10000;

MyDepts = GetByLinks(DataTypes, Filter, MaxNum, MyDevices);

If (Length(MyDepts) < 1) { Log("No linked departments found."); }
```

Here, `GetByLinks` retrieves up to 10,000 `Department` data items that are linked to data items in the `MyDevices` array. Since you are certain that the business has less than 10,000 departments, you can use a large value such as this one to make sure that all `Department` data items are returned.

The returned data items are stored in the `MyDepts` array. Because you want at least one data item from the data type, the policy also prints an error message to the policy log if `GetByLinks` does not return any.

Increasing the alert severity

The final task that you want the policy to perform is to increase the severity of the alert.

About this task

For example, if the department that it affects has a mission critical function in the business. For the purposes of this tutorial, the departments in the business whose function is mission critical are the data center and transaction processing units.

To perform this task, the policy must iterate through each of the `Department` data items that are retrieved in the previous step. For each `Department`, it must test the value of the `Name` field against the names of the two departments in the business that have mission critical functions. If the `Department` name is that of one of the two departments, the policy must increase the severity of the alert to `Critical`.

```
Count = Length(MyDepts);

While (Count > 0) {

    Index = Count - 1;
    MyDept = MyDepts[Index];

    If (MyDept.Name == "Data Center" || MyDept.Name == "Transaction Processing") {
        @Severity = 5;
    }

    Count = Count - 1;

}
```

Here, you use a `While` loop to iterate through the elements in the `MyDepts` array. `MyDepts` is the array of `Department` data items that were returned previously in the policy by a call the `GetByLinks`.

Before the `While` loop begins, you set the value of the `Count` variable to the number of elements in the `MyDepts` array. Each time the loop runs, it tests the value of `Count`. If `Count` is greater than zero, the statements inside the loop are executed. If `Count` is less than or equal to zero, the statements are not executed. Because `Count` is decremented by one each time the loop is performed, the `While` loop runs once for each data item in `MyDepts`.

A variable named `Index` is used to refer the current element in the array. The value of `Index` is the value of `Count` minus one, as Netcool/Impact arrays are zero-based structures whose first element is counted as zero instead of one.

Inside the loop, the policy uses an `If` statement to test the name of the current `Department` in the array against the name of the two mission-critical business departments. If the name of the current `Department` matches the mission-critical departments, the policy sets the value of the `Severity` field in the alert to 5, which signifies a critical severity.

Reviewing the policy

After you finish writing the policy, you can review it for accuracy and completeness.

About this task

The following example shows the entire text of this policy.

```
// Look up device information

DataType = "Device";
Filter = "Hostname = '" + @Node + "'";
CountOnly = False;

MyDevices = GetByFilter(DataType, Filter, CountOnly);
MyDevice = MyDevices[0];

If (Length(MyDevices) < 1) { Log("No matching device found."); }

// Look up business departments
```

```
DataTypes = {"Department"};
Filter = NULL;
MaxNum = 10000;

MyDepts = GetByLinks(DataTypes, Filter, MaxNum, MyDevices);

If (Length(MyDepts) < 1) { Log("No linked departments found."); }

// If department is mission-critical, update severity of alert

Count = Length(MyDepts);

While (Count > 0) {

    Index = Count - 1;
    MyDept = MyDepts[Index];

    If (MyDept.Name == "Data Center" || MyDept.Name == "Transaction Processing") {
        @Severity = 5;
    }

    Count = Count - 1;

}
```

Running the solution

The final step in this tutorial is to run the event enrichment solution.

Before you begin

Before you run the solution, you must configure the TUT_READER_01 OMNIbus event reader service so that it triggers the EnrichEvent policy. To configure TUT_READER_01:

1. Open Netcool/Impact and select **Impact UI** > **Services**
2. Select the the TUT_READER_01 service and click **Edit**.
3. Click the **Event Mapping** tab.
4. To create a mapping, click the **New Mapping** button.
5. If you want to trigger the **EnrichEvent** policy for all events, leave the **Filter Expression** field empty. If you want to trigger the **EnrichEvent** policy for specific events, enter the values for these events.
6. Select **EnrichEvent** in the **Policy to Run** field.
7. Click the **Active** check box.
8. To save the configuration, click **Ok**.
9. To save the changes to the TUT_READER_01 service, click the save icon.

Procedure

To start the solution, you simply start the OMNIbus event reader service.

The event reader then begins to monitor the ObjectServer and retrieves any new events that appear. When a new event appears, the event reader brings it back into Netcool/Impact, where it is processed by running the `EnrichEvent` policy.

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
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

Z

Part Number:
Product Number:

(1P) P/N:

SC27-8783-03