

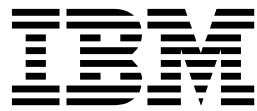
ITCAM for Service Oriented Architecture
7.2 Fix Pack 1 (updated November 2015)

User Guide



ITCAM for Service Oriented Architecture
7.2 Fix Pack 1 (updated November 2015)

User Guide



Note

Before you use this information and the product that it supports, read the information in “Notices” on page 375.

Edition Notice

This edition applies to version 7.2 Fix Pack 1 of ITCAM for SOA and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 2005, 2015.**

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

Contents

About this publication	vii
Intended audience	vii
Publications	vii
ITCAM for Applications library for ITCAM for SOA.	vii
Related publications	viii
Accessing terminology online	ix
Accessing publications online	ix
Ordering publications	ix
Accessibility	x
Application Performance Management community on Service Management Connect.	x
Tivoli technical training.	x
Tivoli user groups	x
Support information.	x
Conventions used in this publication	xi
Typeface conventions	xi
Operating system-dependent variables and paths	xi

Chapter 1. Overview	1
Service-to-service topology	2
Business process monitoring	2
Health and detailed monitoring information.	4
Message traffic management	4
Components of ITCAM for SOA.	5
Integration with WebSphere Service Registry and Repository	7
Web Services Navigator.	8

Chapter 2. Typical usage scenarios	9
Receiving a situation event related to a deployed service operation	13
Monitoring application servers and verifying the service flow through the server.	14
Making a priority assessment of problem tickets and immediately addressing the critical applications	16
Examining service interactions in the SOA	17
Diagnosing a process group problem	20
BPM impact analysis	21
Additional usage scenarios and task examples.	21

Chapter 3. Workspaces.	23
Predefined workspaces	24
The Navigator Physical view	24
The Navigator ITCAM for SOA view	26
Accessing the Navigator ITCAM for SOA view	26
Accessing workspaces by using links	27
Analyzing a triggered situation.	27
Examining a single operational flow from the Operational Flows workspace	28
Examining a single operational flow from the Operational Flow for Application Server workspace.	28
Viewing requester IDs for an operation instance	28

Linking to performance summary data from an Operational Flow workspace	29
Link from a Performance Summary workspace to operational flows and requester identity workspaces	29
Linking to other monitoring agent workspaces	29
Linking to the DataPower WebGUI console.	31
Linking from a Services Management topology view to the Operational Flow for Application Server workspace or Performance Summary workspace.	32
Viewing historical data	32
Workspace and link version dependencies	33

Chapter 4. Workspaces for configuration and metric summaries ..	35
Enterprise Status workspace.	35
Services Management Agent workspace	36
Message Arrival workspace	37
About message arrival situations	37
Message arrival situations in the workspace	38
Message Arrival Details table view	39
Message Arrival by Service view	39
Message Arrival by Operation view	39
Application Server Services Management workspace	39
Average Response Time by Operation view.	40
Number of Messages by Operation view	40
Average Message Size by Operation view	40
Services Management Agent Environment workspace.	40
Average Response Time by Operation view.	40
Fault Summary by Operation view	41
Average Message Size by Operation view	41
Performance Summary workspace.	41
Average Response Time by Operation view.	41
Services Inventory view	41
Message Summary workspace	42
Number of Messages by Service:Operation:Type view.	42
Average Message Size by Service:Operation:Type view.	43
Faults Summary workspace	44
Number of Faults by Operation view.	44
Fault Details view	44
DataPower Console workspace	46

Chapter 5. Workspaces for monitoring by requester identity.	47
Environments that support monitoring by requester identity.	47
Requester Identity Monitoring Configuration workspace.	49
Requester Identity Monitoring Status view	49
Monitored Requester Identities view	50
Requester Identities for Operation workspace	50

Performance Summary for Requester Identity workspace	51
Message and Fault Count view	51
Response Time by Operation view	52
Message Size by Operation view	52

Chapter 6. Workspace for service registry integration 53

Limitations of service registry integration monitoring	54
Services Management workspace	55
Services Overview table view	56
Static topology views	57
Resource type icons	57
Relationships between resource types.	58
Service Details view	58
Service Port Details view	61
Business Processes for Service view	63
Business Processes for Service Port view.	65
Business Process Details view	66
Viewing metadata	68
Setting the threshold for static topology	69

Chapter 7. Workspaces for service-to-service topology 71

Service-to-service topology concepts	71
How service-to-service topology works	73
Operational flows and service transaction flows	74
Topology data in the SOA Domain Management Server	74
Calculating the status of operations	74
Summarized relationship metrics	78
Topology display elements	79
Resource type icons for operation aggregates in the view	79
Resource type icons for operation instances in the view.	81
Status indicators for resource type icons	83
Relationships between resource types.	84
SCA component display	85
Business Process Definition display	90
Operational Flow workspaces	91
Operational Flows workspace	92
Operational Flow for Operation workspace.	93
Operational Flow for Application Server workspace.	95
Navigating the service-to-service topology	95
From the Situation Event Results workspace to an Operational Flows workspace	96
From a row in the Services Inventory table view to an Operational Flow workspace	97
From an operation instance in the Operational Flows workspace to the Operational Flow for Operation workspace	97
From an operation instance in the Operational Flow for Application Server workspace to the Operational Flow for Operation workspace.	97
From an operation instance in any Operational Flows workspace to the Requester Identities for Operation workspace	98

From a DataPower operation instance in any Operational Flows workspace to the DataPower Console workspace.	98
From a data collector node in the Navigator Physical view to the Operational Flow for Application Server workspace	98
From an operation instance in any Operational Flows workspace to the Performance Summary workspace.	98
From a view in the Services Management workspace to the Operational Flow for Application Server workspace	99
From a node in the Interaction Detail view to the Business Process Manager Summary workspace	99
Additional features of Operational Flow workspaces	99
Using resource actions in the Operation Flow portion of the view	99
Using resource actions in the Interaction Detail portion of the view	100
Using toolbar functions	101
Viewing details in a flyover window	102
Searching views	102
Viewing topology in table mode	104
Using the status area	106
Viewing detailed information from Operational Flow workspaces	107
Viewing metrics	107
Viewing details.	110
Viewing business processes.	120
Settings in Operational Flow workspaces	120
Selecting the time span for topology metrics	120
Viewing and setting properties	121
Maintenance actions for Operational Flow workspaces	122
Deleting an operation instance from the service-to-service topology views.	122
Deleting a BPD node from the service-to-service topology views.	123
Deleting unmanaged objects	123
Updating IP addresses for operation instances	126
Service-to-service topology for DataPower.	128
DataPower firmware levels and configuration	128
Topology views for multiple domains	128
Topology views for a domain assigned to multiple display groups	129
Service-to-service topology for WebSphere Message Broker.	130
Mapping WebSphere Message Broker concepts to the service model	130
Supported transport protocols.	132
SOAP node support	135
Collector node support	136
Message flow to topology mapping rules	136
Sample applications and resulting topology	138
Limitations	145

Chapter 8. Workspace for monitoring service health 149

Service groups and process groups	149
Determining front-end services	150
Health indicators of a group	151

The Group Summary workspace	152
The Group Summary view	153
Managing groups	163
The Groups window	164
Configuring for unavailability	180
Configuring the list of unavailability situations	180
Displaying unavailability	181

Chapter 9. Creating custom workspaces and links 185

Link symbols for the Operation Flow view . . .	185
Building your own views	186
Linking to a new workspace containing an	
Operational Flow for Operation topology view ..	187
Creating the workspace	187
Defining the dynamic workspace link . . .	189
Linking from the Situation Event Results	
workspace	195

Chapter 10. Situations 199

How the situations work	199
Avoid using negative values	200
Expert advice	200
Predefined situations	200
The Fault_610 situation	201
The MaxMessageSize_610 situation	202
The MaxResponseTimeCritical_610 situation ..	203
The MaxResponseTimeWarning_610 situation	204
The MessageArrivalClearing_610 situation. .	205
The MessageArrivalCritical_610 situation . .	207
The MessageSize_610 situation	209
The ResponseTimeCritical_610 situation . .	210
The ResponseTimeWarning_610 situation . .	211
The BusinessProcessFault situation	212
The BusinessProcessTerminated situation . .	212
Creating your own situations	213
Creating situations for message arrival traffic	213
Creating situations using the requester identity	
attribute	214
Measuring service unavailability	214
Using workspace links	221

Chapter 11. Take Action commands 223

About Take Action commands.	223
More information about Take Action commands	223
Predefined Take Action commands	223
AddFltrCntrl_610 Take Action command . . .	226
AddMntrCntrl_610 Take Action command. . .	229
AddRequesterIdentity_610 Take Action	
command	233
DeleteRequesterIdentity_610 Take Action	
command	235
DeleteSubnode Take Action command	236
DelFltrCntrl_610 Take Action command . . .	238
DelMntrCntrl_610 Take Action command . . .	241
DisableDC_610 Take Action command	243
DisableReqIDMntr_610 Take Action command	244
EnableDC_610 Take Action command	245
EnableReqIDMntr_610 Take Action command	247
SetReqIDTypeHostIP Take Action command ..	247

SetReqIDTypeUserInfo Take Action command	248
updateLogging_610 Take Action command . . .	249
updateTracing_610 Take Action command . . .	251
UpdMntrCntrl_610 Take Action command. . . .	252

Chapter 12. Attribute groups 257

Attribute groups by product version.	257
Attribute groups used by predefined workspaces	258
Attribute groups and situations	259
Estimating table sizes in the Tivoli Data Warehouse	
database for historical data collection	259
Agent Global Configuration_610 attributes . .	260
Data Collector Filter Control_610 attributes . .	261
Data Collector Global Configuration_610 attributes	263
Data Collector Monitor Control_610 attributes .	264
Fault Log_610 attributes	266
Message Arrival Threshold_610 attributes . . .	269
Requester Identity Monitor Control_610 attributes	270
Services Inventory_610 attributes	271
Services Inventory Requester Identity_610	
attributes	276
Services Message Metric_610 attributes	282
Endpoint Inventory attributes	285
Business Process Events attributes	290
BPM Associated Errors attributes.	294
Mediation Configuration_610 attributes. . . .	296
Internal tables	297
Relationship Request Metrics attributes. . .	297
Relationship Response Metrics attributes . .	299
Relationships attributes	302
Service Flow Metrics attributes	304
Service Port Operation Mapping attributes ..	305
Environment Mapping attributes	307
Subnode Environment Mapping attributes. . .	309
Adapter_Binding_Metrics attributes	310
BPM_Dynamic_Data attributes	311
BPM_Static_Data attributes.	312
BPD_Activity_Events attributes	312
Business_Process_Description attributes . .	314
BPD_Application_Environment attributes . .	316
BPD_Gateway_Events attributes	316
BPD_Notification_Events attributes	318
BPD_Process_Execution_Events attributes . .	320
BPD_Faults attributes.	322
Business_Process_Situation_Data attributes ..	322

Chapter 13. Workflow policies 325

Creating policies	325
Predefined policies	325
The MessageArrival_610 policy	325

Appendix A. Integrating with other products 327

Integration with IBM Tivoli License Compliance	
Manager	327
Integration with IBM Tivoli Business Service	
Manager	327
Integration using Workspace Links	327
Integration with IBM WebSphere Service Registry	
and Repository	327

Integration with IBM WebSphere Business Modeler	328
Integration with Tivoli Change and Configuration Management Database and Tivoli Application Dependency Discovery Manager	328

Appendix B. Determining status for operation instances, operation aggregates, and groups 331

Status values	331
Determining status for operation instances	332
Determining status for operation aggregates	332
Rules for calculating operation aggregate status	333
Determining status for dependent pairs	336
Determining status for composite relationships	337
Cyclic flows	339
Determining status for groups	340

Appendix C. Platform tuning 341

Configuring Monitoring Intervals	341
SOA Domain Management Server considerations	342
Verifying the configured monitoring interval	345
Memory usage considerations	345
Preventing data for the Services Message Metric attributes group from being saved	347
Selectively disabling data collection	347
Configuring the Cleanup Service parameters	348
Setting data age for BPDs	349
Excluding faults from the calculation of service metric values	350

Reducing on-demand user resource usage	351
Calculating and reducing memory required for workspace queries	352
Using custom queries to reduce resource usage	352
Restricting the number of rows	352
Restricting the number of columns	352
Using the same query in a workspace	352
Collecting agent data less frequently	353

Appendix D. ITCAM for SOA SDMS agent Attributes reference 355

Attribute groups for the ITCAM for SOA SDMS monitoring agent	355
Attributes in each attribute group	355
Performance Object Status attribute group	356
Process Groups attribute group	359
SOA All Groups attribute group	360
SOA Group Status attribute group	362

Appendix E. Accessibility 367

Index 369

Trademarks 373

Notices 375

Privacy policy considerations	376
-------------------------------	-----

About this publication

This publication provides information about configuring and using monitoring agent functions to monitor web services.

Intended audience

This guide is for services architects and services application support personnel who use ITCAM for SOA to monitor and manage web services in a service-oriented architecture (SOA) environment on distributed Microsoft Windows, Linux, AIX®, HP-UX, and Solaris systems, and IBM® z/OS® enterprise systems.

Users of this publication must be familiar with these topics:

- Monitoring concepts
- The commonly shared components of IBM Tivoli® Management Services
- The Tivoli Enterprise Portal user interface
- The IBM Tivoli Monitoring
- Services that you want to monitor

Publications

This section lists publications in the product library and related documents. It also describes how to access Tivoli publications online and how to order Tivoli publications.

ITCAM for Applications library for ITCAM for SOA

The following publications are included in the ITCAM for Applications library, available in the ITCAM for Applications Information Center:

- *IBM Tivoli Composite Application Manager for SOA Installation Guide*

Provides an overview of the IBM Tivoli Management Services environment and the planning information and procedures you need to install and upgrade the application support files and the monitoring agent in a distributed operating system environment.

This guide also includes procedures for configuring support for the service-to-service topology function, including creating databases and configuring SOA Domain Management Server and Tivoli Common Object Repository in your Tivoli Enterprise Portal Server environment.

This guide includes procedures for enabling and disabling the various supported runtime environments for data collection by the ITCAM for SOA, version 7.2 or later monitoring agent, and optional administrative tasks to further configure your installation.

- *IBM Tivoli Composite Application Manager for SOA User's Guide*

Provides information on monitoring and managing resources in the Tivoli Enterprise Portal environment, including details about Take Action commands, situations, workspaces and views, including service-to-service topology workspaces and views. Some problem determination information about the various components of ITCAM for SOA is also provided, as well as information

about log files and informational, warning, and error messages. This publication complements the Tivoli Enterprise Portal online help information for this monitoring agent.

- *IBM Tivoli Composite Application Manager for SOA Tools*
Provides information about installing and using the IBM Web Services Navigator, an Eclipse based plugin for extracting services information that has been collected by monitoring agents and stored, either locally or in a historical database. This tool provides the capability to retrieve historical metric data from a connected database, or assemble several locally stored metric and content log files, and display the resulting data in several views to assist a services architect in visualizing relationships between services.
- *IBM Tivoli Composite Application Manager for Discovery Library Adapters Guide*
Provides information about installing and running the following discovery library adapters (DLAs) provided with ITCAM for SOA: WebSphere® Service Registry and Repository Discovery Library Adapter, Business Process Execution Language for Web Services Discovery Library Adapter, and IBM Tivoli Composite Application Manager for SOA Discovery Library Adapter.
- *IBM Tivoli Composite Application Manager for SOA Troubleshooting Guide*
Provides information about recovering from problems that you might encounter while installing, configuring, and using the product. Typical problem scenarios are described, and recovery procedures are provided. Error messages for the product are also documented in this guide.
- *IBM Tivoli Composite Application Manager for SOA WSRR Integration Guide*
Provides information about integrating ITCAM for SOA version 7.2 or later with WebSphere Services Registry and Repository version 7.5 or later. The procedure for subscribing to WSRR events related to service-level definitions and the procedure for creating and deploying an SDMS configuration file is documented. The configuration file defines the rules for processing WSRR events in SDMS. Based on these rules, situations are automatically created, updated, or deleted by IBM Tivoli Monitoring when a lifecycle changes notification is received from WSRR.
- *IBM Tivoli Composite Application Manager for SOA BPM Monitoring Deployment Guide*
Provides information about implementing an IBM BPM monitoring solution.
- *IBM Tivoli Composite Application Manager for SOA Reports Guide*
Provides information about installing and using ITCAM for SOA Reports.

Related publications

The following documentation also provides useful information:

- IBM Tivoli Documentation Central:
Information about IBM Tivoli Documentation is provided on the following website:
https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli_Documentation_Central
- IBM WebSphere Application Server:
Information about IBM WebSphere Application Server is provided on the following website:
<http://www.ibm.com/software/webservers/appserv/was/library/>
- ITCAM for Application Diagnostics library:

Information about ITCAM for Application Diagnostics Managing Server is provided on the following website:

http://publib.boulder.ibm.com/infocenter/tivihelp/v24r1/index.jsp?topic=%2Fcom.ibm.itcamfad.doc_7101%2Fic-homepage.html

- IBM DB2®:

Information about IBM DB2 is provided on the following website:

<http://www.ibm.com/software/data/sw-library/>

- IBM SmartCloud® Application Performance Management UI:

The *IBM SmartCloud Application Performance Management User Interface User's Guide* is available from the SmartCloud Application Performance Management information center at the following URL:

http://pic.dhe.ibm.com/infocenter/tivihelp/v63r1/index.jsp?topic=%2Fcom.ibm.apm.doc_7.6%2Fapm_ui_docs%2Fapmui_76%2Ffac_landing_user.html

Accessing terminology online

The IBM Terminology website consolidates the terminology from IBM product libraries in one convenient location. You can access the Terminology website at <http://www.ibm.com/software/globalization/terminology>.

Accessing publications online

The documentation CD contains the publications that are in the product library. 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 Documentation Central website at [https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli Documentation Central](https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Documentation%20Central)

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

Ordering publications

You can order many Tivoli publications online at: <http://www.ibm.com/e-business/weblink/publications/servlet/pbi.wss>.

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

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

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

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

Accessibility

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

For additional information, see Appendix E, “Accessibility,” on page 367.

Application Performance Management community on Service Management Connect

Connect, learn, and share with Service Management professionals: product support technical experts who provide their perspectives and expertise.

Access Service Management Connect at <https://www.ibm.com/developerworks/servicemanagement/apm/index.html>. Use Service Management Connect in the following ways:

- Become involved with transparent development, an ongoing, open engagement between other users and IBM developers of Tivoli products. You can access early designs, sprint demonstrations, product roadmaps, and prerelease code.
- Connect one-on-one with the experts to collaborate and network about Tivoli and the (enter your community name here) community.
- Read blogs to benefit from the expertise and experience of others.
- Use wikis and forums to collaborate with the broader user community.

Tivoli technical training

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

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

Tivoli user groups

Tivoli user groups are independent, user-run membership organizations that provide Tivoli users with information to assist them in the implementation of Tivoli Software solutions. Through these groups, members can share information and learn from the knowledge and experience of other Tivoli users. For more information about Tivoli Users Group, see www.tivoli-ug.org.

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 <http://www.ibm.com/software/support/probsub.html>.

Troubleshooting Guide

For more information about resolving problems, see the *IBM Tivoli Composite Application Manager for SOA Troubleshooting Guide*.

Conventions used in this publication

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

Typeface conventions

This publication uses the following typeface conventions:

Bold

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

Italic

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

Monospace

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

Operating system-dependent variables and paths

This guide refers to the following variables:

- *ITM_home*: the top-level directory for installation of IBM Tivoli Monitoring components. The default location is C:\IBM\ITM on Windows systems and /opt/IBM/ITM on Linux and UNIX systems?
- *ITCAM4SOA_Home*: the directory location where IBM Tivoli Composite Application Manager for SOA monitoring agent is installed in the IBM Tivoli Monitoring environment:
 - For Windows systems: *ITM_Home\TMAITM6*
 - For Linux, HP-UX, AIX, and Solaris systems: *ITM_Home/platform/d4*

Determining the *platform* value in directory paths

Throughout this product library, reference is made to the *<platform>* variable, which is part of the Linux or UNIX directory path specification for certain files that you need to access, for example:

<ITM_Home>/<platform>/<product>

In this example, the two-character *<product>* variable is also part of the directory path, and is typically specified as *cq*, *d4*, or *iw* in this guide.

On supported Linux and UNIX operating systems, you can find the value for *<platform>* with this short procedure:

1. From a command prompt, navigate to the *<ITM_Home>/bin* directory.
2. Run the following command:

```
./cinfo -d
```

3. Locate the line for product code *<product>*, for example:

<i>cq</i>	Locate this product code when you are looking up the <i><platform></i> value for Tivoli Enterprise Portal Server.
<i>iw</i>	Locate this product code when you are looking up the <i><platform></i> value for Tivoli Enterprise Portal Server Extension.
<i>d4</i>	Locate this product code when you are looking up the <i><platform></i> value for the IBM Tivoli Composite Application Manager for SOA monitoring agent.

The platform designation is found under the *Platform* column.

The platform designation depends on the operating system, the computer type, and the version of IBM Tivoli Monitoring that is installed. The platform for the *d4* product code is typically not the same as for the *cq* and *iw* product codes.

The following example shows the output of the **cinfo** command when ITCAM for SOA and IBM Tivoli Monitoring are installed on a supported Red Hat Linux operating system on a 64-bit Intel computer:

```
"ProdCode","Description","Platform","Version","Release"

"ax","IBM Tivoli Monitoring Shared Libraries","li6263","06230100","100"
"ax","IBM Tivoli Monitoring Shared Libraries","lx8263","06230100","100"
"ax","IBM Tivoli Monitoring Shared Libraries","lx8266","06230100","100"
"cq","Tivoli Enterprise Portal Server","lx8263","06230100","100"
"cw","Tivoli Enterprise Portal Browser Client","lx8263","06230100","100"
"d4","IBM Tivoli Composite Application Manager for SOA","lx8266","07200100","100"
"gs","IBM GSKit Security Interface","li6243","07402700","100"
"gs","IBM GSKit Security Interface","lx8266","07402700","100"
"hd","Warehouse Proxy","lx8266","06230100","100"
"iu","IBM HTTP Server","li6263","07000000","100"
"iw","IBM Tivoli Enterprise Portal Server Extensions","li6263","07001900","100"
"jr","Tivoli Enterprise-supplied JRE","li6263","06090200","100"
"jr","Tivoli Enterprise-supplied JRE","lx8266","06090200","100"
"kf","IBM Eclipse Help Server","li6263","06230100","100"
"ms","Tivoli Enterprise Monitoring Server","lx8266","06230100","100"
"pa","Tivoli Performance Analyzer","lx8266","06230100","100"
```

"sh","Tivoli Enterprise Monitoring SOAP Server","lx8266","06230100","100"
"sy","Summarization and Pruning Agent","lx8266","06230100","100"
"t1","File Transfer Enablement","lx8266","07300000","000"
"ue","Tivoli Enterprise Services User Interface Extensions","lx8266","06230100","100"
"ui","Tivoli Enterprise Services User Interface","lx8263","06230100","100"

This example shows the following information:

- *lx8263* is the platform for Tivoli Enterprise Portal Server (product code *cq*)
- *lx8266* is the platform for the ITCAM for SOA monitoring agent (product code *d4*)
- *li6263* is the platform for Tivoli Enterprise Portal Server Extensions (product code *iw*)

Chapter 1. Overview

IBM Tivoli Composite Application Manager for SOA (ITCAM for SOA) provides monitoring and management of services and mediations in a service-oriented architecture (SOA) environment. ITCAM for SOA monitors a variety of metrics on many application server runtime environments and enterprise services buses.

ITCAM for SOA supports the following IBM environments:

- IBM Business Process Manager Server
- IBM WebSphere Enterprise Service Bus
- IBM WebSphere Process Server
- IBM WebSphere DataPower® SOA Appliance
- IBM WebSphere Message Broker
- IBM WebSphere Application Server
- IBM CICS® Transaction Server

Other SOA environments are also supported, including:

- Microsoft .Net
- BEA WebLogic Server
- JBoss
- SAP NetWeaver

The term *services* applies to both web services and other services with fixed interfaces. In a distributed SOA environment, a single business operation might involve several interactions between services on different hosts in diverse geographic locations. ITCAM for SOA tracks these interactions.

You can use ITCAM for SOA to view service-to-service topology, with the response times for all the interactions integrated across servers and locations. You can also view detailed information for individual instances of a service.

For services that are based on IBM Business Process Manager (BPM), you can monitor the interaction of various components that implement the service.

Important: Compliance with JSR 109: The client applications for your web services that are monitored by ITCAM for SOA must comply with the conventions described in chapter 4 of Java™ Specification Request 109 (JSR 109), *Java Web Services for J2EE*. For more information, refer to the specification found at <http://www.jcp.org/aboutJava/communityprocess/final/jsr109/>.

You can use two user interfaces to access ITCAM for SOA information. Tivoli Enterprise Portal provides current and historical monitoring information spanning your entire environment. A separate application, Web Services Navigator, provides detailed analysis of offline historical information about service interaction.

Important: For the prerequisites for ITCAM for SOA version 7.2 Fix Pack 1, see the Software product compatibility reports.

Service-to-service topology

ITCAM for SOA provides a set of *service-to-service* topology workspaces and views. These workspaces and views display the structure of interactions between service operations in the entire SOA environment.

A *service operation*, in the context of service-to-service topology, defines a specific function that is provided by a service. One or more service operations and their collective functions represent the service in the topology.

By default, the topology workspaces include all service operations in the environment. You can also organize service operations in *groups*, and display the topology for operations within a group.

A service operation can have several *instances* in various geographic locations. For the top level *operation flow* view, ITCAM for SOA aggregates them into a single entry, called an *operation aggregate*. In this view, you can see a concise summary of the interaction logic.

You can also open an *interaction detail* view for an operation. This view shows the individual operation instances and their interaction with other operations.

Topology workspaces include summary response time metrics. Using workspace linking, you can access detailed statistics for any operation.

Service-to-service topology views help you to understand how services interact. They also display performance metrics for the interactions. Use this information to determine whether a problem exists and to identify the services that are likely to be involved in the problem.

As a services architect, you can use service-to-service topology views to validate the behavior of your SOA design. You can determine whether discrepancies exist between your SOA design and observed service interactions. This information can help you work out whether services are being called and where services are interacting unexpectedly. To identify possible architecture changes to bring about improved performance, you can also analyze relationships and associated metric data.

Business process monitoring

ITCAM for SOA can monitor interactions between components implementing a service in the IBM Business Process Manager (BPM) environment.

ITCAM for SOA displays component interaction topology for every process application that is deployed and started on the BPM server. You can drill down into every displayed component; ITCAM for SOA displays detailed information specific for this component.

This topology includes the following items and interactions:

- Business Process Definitions (BPDs). Each BPD is displayed as a single node, showing interactions with SCA components (including BPEL processes and mediation flows). You can also view a detailed list of the elements within the BPD and the associated performance metrics.
- Business Process Execution Language (BPEL) components. Each BPEL component is displayed as a single node, showing inbound connections (other

services and components calling the component) and outbound connections (this component calling other components or services). You can also view a detailed list of the activities within the BPEL process and the associated performance metrics.

- Mediation flow components. A mediation flow component is displayed as a node; interactions with other mediation flows, other service components, and external systems (databases and registries) are displayed. You can also view a detailed list of the mediation primitives within the flow and the associated performance metrics. Mediation subflows are displayed as separate nodes.

Important: Mediation flow components on IBM WebSphere Enterprise Service Bus are also supported.

- Other SCA components, including adapters and human tasks that are displayed as separate nodes.

ITCAM for SOA automatically gathers the data from all running BPM applications. When a service application is deployed and started, ITCAM for SOA includes BPM nodes for the application in the service-to-service topology display. If an application is uninstalled, ITCAM for SOA tracks the uninstallation to ensure that the topology information remains up-to-date.

The topology workspace also includes summary response time metrics for the interactions. These metrics are only available when the service is called.

By default, the service topology workspace includes all BPM component nodes. You can organize these nodes into *process groups*, and view the topology that is specific to a process group.

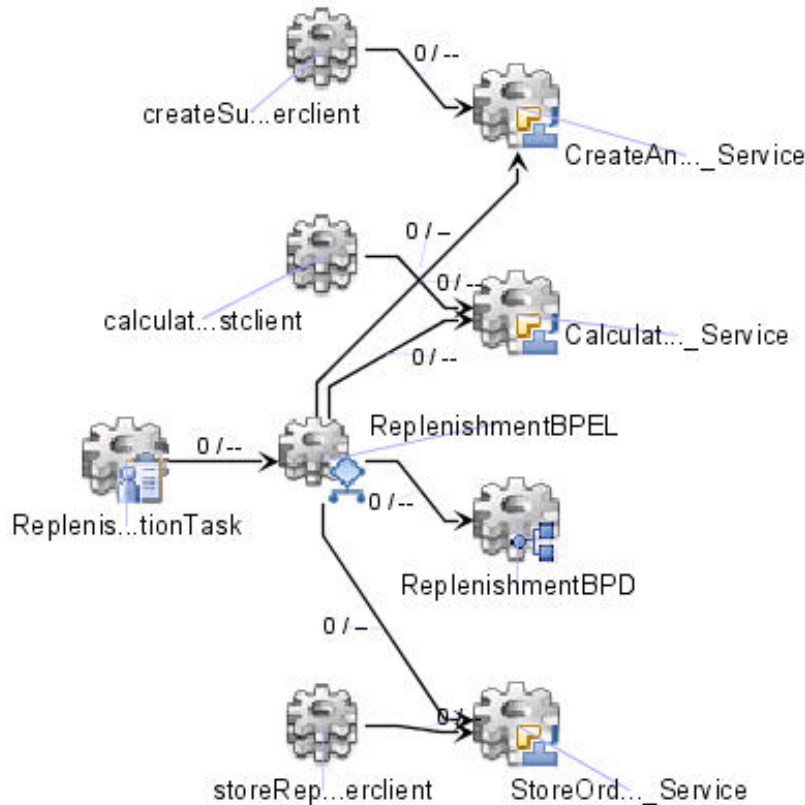


Figure 1. Sample BPM process topology

Health and detailed monitoring information

For each runtime environment, ITCAM for SOA provides health overview and detailed monitoring data. A health overview for the entire enterprise SOA environment is also available.

The health information is determined by overall service response times and is displayed in a summary workspace.

You can also display health information that is specific to a service group or process group. ITCAM for SOA displays alerts when any services in a group are unavailable, enabling prompt resolution.

ITCAM for SOA also displays detailed response time information for services in the environment. Using workspace linking, you can drill down into this information from summary workspaces and service-to-service topology views.

Message traffic management

Using ITCAM for SOA, you can perform simple tasks that control message traffic between services.

You can use Take Action commands to filter messages. The filtering is based on user-configurable criteria such as computer name, service port name, operation name, and the IP address of the requester.

Components of ITCAM for SOA

ITCAM for SOA is installed and operates within the management infrastructure of the IBM Tivoli Monitoring environment. ITCAM for SOA has several components.

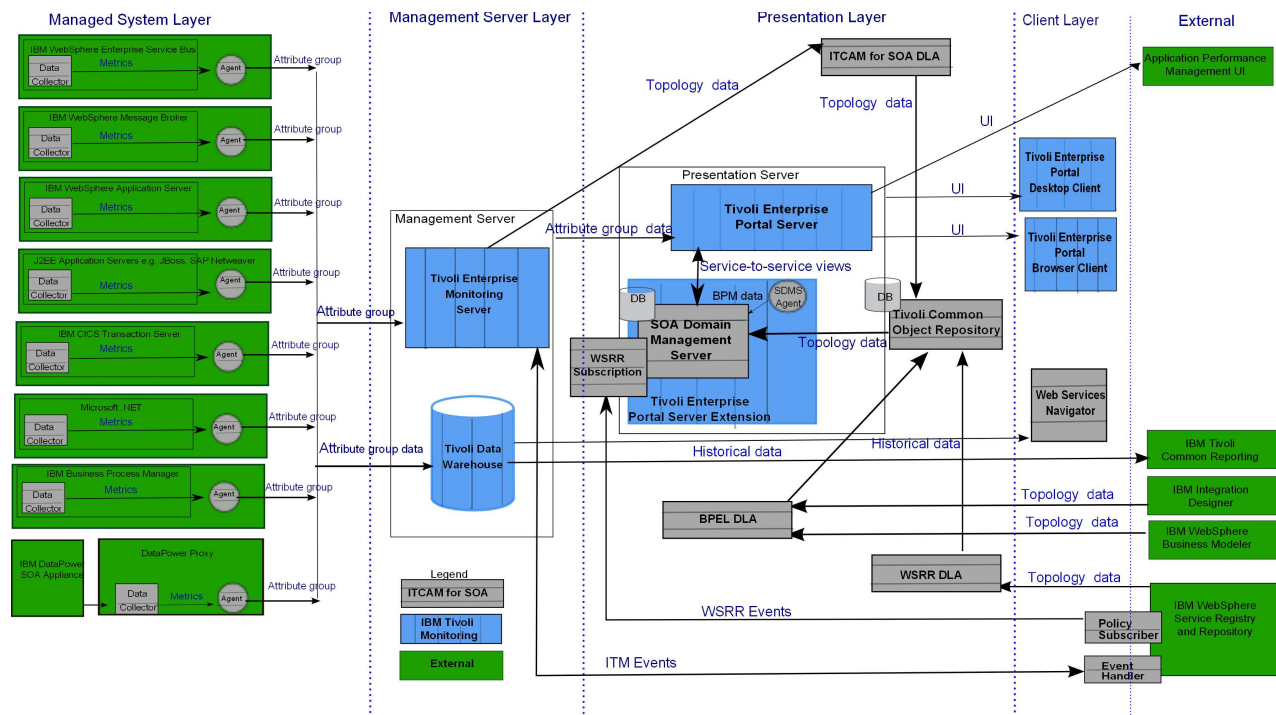


Figure 2. The ITCAM for SOA architecture diagram

Data collector

An ITCAM for SOA data collector must be deployed and configured for every monitored runtime environment. The data collector is installed locally on the runtime environment host. The data collector saves data in metric files. The monitoring agent uses these files to prepare information for display in the Tivoli Enterprise Portal. You can also use these files with Web Services Navigator for detailed analysis.

The data collector detects the information flow topology on the runtime environment and tracks the message traffic within the environment. The data collector collects the following information for every message:

- Source and destination (system name, application server name, service port name, and operation name)
- If the message is a request or a response
- Interaction type (synchronous or asynchronous)
- The association of a request to its response during an asynchronous interaction
- The response time for the message
- Whether the message generated a fault
- Whether the message is a one-way or a two-way message
- Optionally, the actual header and body content of the message itself

A separate data collector is used for monitoring DataPower appliances. It collects the same information by acting as a proxy between the appliance and other services.

ITCAM for SOA monitoring agent

An ITCAM for SOA Tivoli Enterprise Monitoring Agent (*monitoring agent*) interacts with the data collectors.

The monitoring agent is installed on every host that runs one or more runtime environments. It receives data from the data collectors that are configured for the environments, and transmits it to the Tivoli Enterprise Monitoring Server.

ITCAM for SOA SDMS monitoring agent

The ITCAM for SOA SDMS agent is provided with ITCAM for SOA version 7.2 Fix Pack 1 and later. The agent is required only if you want to view Business Process Management monitoring data in the IBM SmartCloud Application Performance Management UI version 7.6 or later. The agent provides minimal user-visible content in the Tivoli Enterprise Portal. No agent-specific workspaces, situations, or take action commands are provided. The attribute groups are designed to provide a data interface to the Application Performance Management UI rather than for use in the Tivoli Enterprise Portal.

Tivoli Enterprise Monitoring Server infrastructure

The Tivoli Enterprise Monitoring Server infrastructure consists of one or more Tivoli Enterprise Monitoring Servers and a Tivoli Enterprise Portal Server. It works with various agents, including the ITCAM for SOA monitoring agent. For more information about this infrastructure, see the Tivoli Enterprise Monitoring information center at http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/index.jsp?topic=%2Fcom.ibm.itm.doc_6.2.2fp2%2Fwelcome.htm.

Use the Tivoli Enterprise Portal to view service-to-service topology spanning multiple servers, along with detailed information for individual servers.

The Tivoli Monitoring Infrastructure also saves the monitoring information to the Tivoli Data Warehouse. Use ITCAM for SOA Agent Reports to create predefined and on-demand reports based on the monitoring information in Tivoli Data Warehouse.

Discovery Library Adapters

Discovery Library Adapters (DLAs) run on the Tivoli Enterprise Portal Server.

A DLA is a program that extracts data from a source application and creates XML files to pass to Tivoli Common Object Repository.

Three DLAs are provided with ITCAM for SOA:

IBM Tivoli Composite Application Manager for SOA DLA

Discovers application server data from the ITCAM for SOA monitoring agents.

WebSphere Service Registry and Repository DLA

Discovers the relationships between services, service ports, operations, and port types of web services that are registered in WebSphere Service Registry and Repository.

Business Process Execution Language for Web Services DLA

Discovers the relationships between port types, operations, and business processes based on business processes that are defined in IBM WebSphere Integration Developer and WebSphere Business Modeler. The BPEL DLA is required for detailed graphical visualization of BPEL process structure.

Tivoli Common Object Repository

Tivoli Common Object Repository (TCORE) runs on the Tivoli Enterprise Portal Server. TCORE stores information that is received from DLAs in a database and passes the information to the SOA Domain Management Server. This information enables SOA Domain Management Server to integrate service registry and business process information with the information about service resources.

SOA Domain Management Server

The SOA Domain Management Server runs on the Tivoli Enterprise Portal Server. The management server aggregates information received from monitored servers and prepares a single view of service-to-service topology.

The management server also manages interaction between ITCAM for SOA and several other products.

Integration with WebSphere Service Registry and Repository

ITCAM for SOA integrates with IBM WebSphere Service Registry and Repository and with IBM DataPower to display registered services in topology views and to monitor compliance with predefined service levels.

WebSphere Service Registry and Repository is a metadata repository for service descriptions.

ITCAM for SOA can obtain the list of registered services from WebSphere Service Registry and Repository and display them in the topology views. To use this function, see the *IBM Tivoli Composite Application Manager for SOA Discovery Library Adapters* guide.

Within WebSphere Service Registry and Repository, you can set up *Service Level Definitions* (SLD) for individual services within your SOA environment. The SLD in the WebSphere Service Registry and Repository governance enablement profile provides a formal specification of the required interaction with a provided service. The SLD also includes non-functional characteristics related to the interaction, such as the service level agreement policies.

ITCAM for SOA monitors the conditions that the service must meet, such as response time, based on these policies.

To monitor the conditions, ITCAM for SOA maintains one or several Tivoli Monitoring situations for every SLD to be monitored. If any policy that is defined in the SLD is breached, a situation is triggered. Then Tivoli Monitoring can notify users and run actions, such as scripts.

ITCAM for SOA sends service status information and situation notifications to WebSphere Service Registry and Repository. You can use WebSphere Service Registry and Repository to select services based on performance and other metrics, and also to track resolution when service levels are not met.

Web Services Navigator

Web Services Navigator comes with ITCAM for SOA. Use Web Services Navigator for a detailed analysis of historical data in Tivoli Data Warehouse and in data collector log files. This analysis is useful for development and detailed support investigation.

ITCAM for SOA data collectors log all message information. The Web Services Navigator tool uses the log files and the data that is saved in the Tivoli Data Warehouse data warehouse. The tool displays information about the monitored web services and the interactions between them. You can import multiple log files from different runtime environments into Web Services Navigator at the same time.

To use Web Services Navigator, you must install it separately on your client system. Web Service Navigator is available in the ITCAM for SOA Tools package.

Chapter 2. Typical usage scenarios

The following scenarios describe some usage scenarios using the Tivoli Enterprise Portal and ITCAM for SOA to monitor and diagnose problems in your application environment.

These usage scenarios refer to the following fictitious personas that might reflect typical positions in your organization:

Table 1. Primary Personas

Persona	Description
Annette/Olivia – Level 2 Operator	<p>Annette's (or Olivia's) primary focus is to ensure service availability by fixing the important incidents according to the following criteria:</p> <ul style="list-style-type: none">• Fix the incidents as quickly as possible.• Fix as many incidents within the allotted timeframe as possible.• Understand the impact of any incident to a particular service.• If the incident cannot be resolved, then follow the escalation procedures.• Do not miss any critical incidents.• Know the incident to be worked next, prioritize according to urgency, impact, and service level agreement (SLA).• Know about the incidents before any customer calls.• Communicate with the necessary people (within Network Operations Center (NOC), to SMEs and outside to the business units).• Have a smooth transition between shifts (outstanding problems resolved or at a state where they are being worked on). <p>Annette (or Olivia) proactively finds negative trends affecting the business and follows proper change management procedures to prevent incidents from occurring or recurring. She does not like to be bombarded by false alarms.</p>

Table 1. Primary Personas (continued)

Persona	Description
Jim/Miles – Middleware/ Application Support Subject Matter Expert (SME)	<p>Jim's (or Miles') primary focus is to ensure that the middleware applications he is responsible for are up and running at all times and performing within expected response thresholds. If an application goes down, then a line of business is affected and has a direct impact on how his team is rated against the SLA. He also works with the systems monitoring and automation group to define the appropriate monitors and thresholds for his domain area of responsibility.</p> <p>Additional goals:</p> <ul style="list-style-type: none"> • To be notified of a problem as specified by the agreed-upon escalation procedures and not be distracted in things he is not interested in. • To make sure that everything is working as expected, so the company is running smoothly, and that appropriate process are in place, so that L1 and L2 can fix simple issues. If he has to be contacted at home for an issue, he knows that it requires his expertise. • Quickly and accurately determine the subject matter expert to whom he is to route the problem if he, himself, cannot fix it. • Provide other subject matter experts (application developer, web administrator, network administrator, and so on) with enough information to fix availability and performance problems. • Have the correct tools to control the content he can see, while not having control over how monitoring is implemented. • To focus more on transaction monitoring in his middleware. • To gauge capacity that his systems will need in the future. • To have adequate levels of monitoring in place (that can detect problems ahead of time and alert only on real problems) so that being paged in the middle of the night is more the exception than the rule. • Refine and update SLA thresholds to better reflect actual needs of line of business (LOB), meeting with most LOB managers to understand their experiences with WAS-hosted applications and gather information to help with his SLA projects. • Drive an increase in the percentage of upstream resolutions, and document and quantify the amount of time and resources saved. • Deliver SLA threshold compliance reports to all stakeholder groups on a quarterly basis: <ul style="list-style-type: none"> – For outages that exceed thresholds provide problem details and business user impact. – Derive some of this data from trouble ticket system reports. • Sync up with other SMEs in Level 2 to improve efficiencies in business processes and look at ways to grow the team next year. • Correlate problem occurrences with changes in the IT environment. • Definitively identify the root cause of problems, as well as the potential impact those problems have or potentially could have. • Fix problems (in the agreed upon timeframe) without having to escalate them to the services, development, or architecture teams. • Be aware of changes to the services architecture before they are rolled into the production environment.

Table 1. Primary Personas (continued)

Persona	Description
Maria – BPM Maintainer	Maria's primary focus is to reproduce BPM reported problems using the administration account or using the user account. She must be able to identify where the BPM problem is located, and resolve simple problems herself or ask for help from the IT Administrator. She might sometimes deploy the problem fix upon the user requirement, or refer to the service level agreement to deploy the fix. She also educates users on how to use the system correctly.
Adam – BPM IT Administrator	Adam's primary focus is to support the line of business (LOB) when they want to author a process, need a new role, need a new service, or get confused. Adam works within a test and production environment that empowers the business user and is available at all times.
Sophie – Install and Configuration Administrator	<p>Sophie's primary focus is on installing and configuring the tools (input feeds) needed to produce relevant and accurate information, in terms of classification, severity, and priority. Sophie is responsible for keeping the tools available at all times and updated with latest versions, patches, and e-fixes. Her duties include installation, verification in a production environment, and any data migration needed.</p> <p>Sophie assists the lead product administrator in supporting the users of the tools by:</p> <ul style="list-style-type: none"> • Training operators on how to use the tools. • Answer “why did this issue happen” questions. • Perform after-the-fact problem determination or error analysis. • Integrate the data sharing between tools and the smooth launching between them <p>Sophie must be able to quickly and accurately pinpoint software problems with a tool, and work with the tool vendors regarding problems and enhancements.</p>

Table 2. Secondary Personas

Persona	Description
Dave (or Deepa) – Application Developer	Dave's (or Deepa's) primary focus is to improve the company profits by developing in-house applications that work correctly the first time. He incorporates best practices in his coding to leverage the knowledge of others in his programs, and is diligent when conducting code inspections. When a problem comes up in a production application he is sent trace files so he can analyze the problem, which he then tries to simulate in his environment. He accurately determines the cause of performance problems when requested, and has a good understanding of the business aspects of the application he is developing.

Table 2. Secondary Personas (continued)

Persona	Description
Connie – SOA Development Manager	Connie's primary focus is managing assets through the development and test cycles. She analyzes requirements and designs solutions, including Service Interface Specification, prototypes service implementation, works with development and test / mediation teams to implement the solutions. Connie also develops prototypes of applications and helps define dependencies on service. She publishes "Golden Master" code to the Operations manager, creates Documents of Understanding (DoU) and subscription requests, and creates Service Level Definitions (SLDs). Connie evaluates specifications, evaluates the SLD and creates SLAs per the conditions of the DoU.

Table 3. Buyer Personas

Persona	Description
Allen – System Management Architect	Allen's primary responsibility is to always know the status of the existing monitoring tools and services. He is aware of current monitoring and management technology and ensures that the service architecture has a high level of reliability in production. Allen detects the differences (without error) between the designed services architecture and the actual services architecture. He prototypes potential changes to the architecture and has a realistic view of their effects. Allen is able to show the relationship between the business process and the services architecture.
Kathryn – LOB Manager	<p>Kathryn's primary focus is to drive up revenue from her customer facing services. The goal is an increase greater than 20%. She fosters a strong working relationship with the CIO and senior IT management. She negotiates and restructures contracts with outsourced contracting and consulting groups to better reflect development and support needs for her rolling 24 month planning cycle.</p> <p>In the short term, Kathryn wants to know only the significant problems with her services. For these critical incidents, she needs to:</p> <ul style="list-style-type: none"> • Proactively see the critical incidents before any external customer calls about them. • Understand the incident in terms of the impact to the external customer, and not be overwhelmed with technology information. • Feel comfortable knowing that if there is an incident, the IT personnel are aware of it and already working to resolve it. • Know when business services are trending toward violation of the SLA. <p>In the long term, Kathryn needs to know how well her applications are providing services to, and producing revenue from, the external customers. She ensures that the IT team is monitoring the critical resources for her application, and obtains service availability and IT cost information from the IT team. She works with IT administrators and architects to model the business processes, and helps IT understand the service applications, and their customers' steps and their experiences. Kathryn helps the IT team prioritize which incidents are critical to their customers, and works to ensure that she is getting the best value for her IT dollar (for example, by comparing her internal IT team to an IBM IGS outsource group).</p>

Receiving a situation event related to a deployed service operation

Scenario overview

Annette, the level 2 Operator, is monitoring the Tivoli Enterprise Portal Navigator Physical view for deployed web service operations. She notices a blue situation event indicator on the Enterprise and Windows system nodes. She sees that the IBM Tivoli Monitoring Situation Event Console has a blue informational situation event related to a service fault. She decides to handle this potential problem proactively and goes to the service-to-service topology to triage the situation by looking at business impacts and root causes.

Problem determination

Annette, the level 2 Operator, knows that a fault situation can be triggered if web service faults occurred, based on a defined filtering criteria or an application server problem. She takes the following steps to identify the potential problem.

Step 1. Examining the details of the situation

To access the Situation Event Console, Annette clicks the Enterprise node in the Navigator tree display. In the Situation Event Console, she right-clicks in the row of interest (not the link icon) and clicks **Situation Event Results** to open the Situation Event Results workspace. She wants to see the details of the situation and examine if it is a customized fault situation. For more information about using a predefined fault situation that is provided with ITCAM for SOA, see “The Fault_610 situation” on page 201.

Step 2. Using Expert Advice for guidance on the situation

Annette looks at the yellow highlighted row in the Current Situation Values view. The **Service Type** column has a *Provider* value. She notices that the **Fault Count** column is 2, indicating that two faults occurred during the last time period. She identifies that the situation started because the fault count was greater than 0.

She looks at the Expert Advice view for anymore details about the severity of the situation event. She reads the **Suggested Actions** area and decides to use the service-to-service topology to diagnose the potential problem.

Step 3. Launching into the service-to-service topology from the context of the situation

In the Initial Situation Values view, Annette right-clicks the link indicator that is beside the yellow fault situation. In order to open the Operational Flow for Operation topology workspace, she clicks **Operational Flow for Operation**. She uses the Operation Flow and Interaction Detail topology views to identify the operation flows that contain the specific operation, the details about that operation flow, and what is upstream so that she can further investigate the problem.

Step 4. Using the Operational Flow for Operation workspace to understand where web services traffic is coming from and to examine the business impact

In the Operation Flow view portion of the Operational Flow for Operation topology workspace, Annette sees the flow in which the operation aggregate participates. The specific operation aggregate is preselected. She hovers the cursor over the event indicator icon of the preselected operation, which indicates that it has a status.

To verify if any values are getting close to exceeding a threshold, Annette checks the metrics that are associated with the call relationship. She positions her cursor over a call relationship and views the Metrics notebook and the default **Chart** tab. The metrics are organized into metric type and interception point regarding the provider and the requester. So that she can view the numbers that are behind each of the metrics, she clicks the **Table** tab. Then, to see any noticeable changes, she looks at the **Fault Count** designation. After viewing the operation flow details and metrics, she drills down so that she can examine further.

She examines the Interaction Details view, which displays all instances of the operation. To examine the details, she hovers the cursor over an instance. She does the same action for the other operation instances that are part of the flow and checks the metric values again.

She checks whether any business processes are aligned with the operation. She receives an information message stating that business processes are associated with the operation.

Then, to see the fault details, she goes to the Performance Summary workspace.

Step 5. Navigating from the Performance Summary workspace to the Faults Summary workspace to examine the fault details before deciding a solution

Annette navigates to the Faults Summary workspace and examines the Fault Details table view for a real-time view of the data that is being received. She notices an application server error code and decides to provide Jim with her findings in the form of a problem ticket.

Step 6. Escalating the situation for further observance

Annette acknowledges the current problem, reviews her notes, and decides to deal with this issue proactively by opening a problem ticket and routing it to Jim for further observance and investigation.

Monitoring application servers and verifying the service flow through the server

Scenario overview

Annette, the level 2 Operator, is monitoring the application servers by observing the Tivoli Enterprise Portal Navigator Physical view. She notices that the Services Management Agent Environment node, and in particular the *D4:Appserver28* node has a red critical event indicator attached to it. Realizing that a defined situation is true and that a problem exists, Annette tries to resolve the issue quickly.

Problem determination

Annette is aware that a **MaxResponseTimeCritical_610** situation is triggered when the maximum elapsed response time for a web service request exceeds a specified threshold value. She remembers an email stating that all of the web services that run on *Appserver28* are highly important because they deal with customer transactions. She realizes that this situation has a high impact and completes the following steps to identify the problem.

Step 1. Examining the details of the situation

Annette places the mouse cursor over the red critical event indicator on the *D4:Appserver28* node. In the flyover window that lists open situation events, she

clicks the link indicator for the **MaxResponseTimeCritical_610** situation. In the Situation Event workspace, she examines the details of the situation.

Step 2. Using Expert Advice for guidance on the situation

In the Situation Event workspace, Annette looks at the **MaxResponseTimeCritical_610** situation row that is highlighted with red in the Current Situation Values view. The **Service Type** column has a *Requester* value. She notices that the **Max Elapsed Time** column value exceeds the threshold limit of 10000 milliseconds (10 seconds).

She looks at the **Suggested Actions** area of the Expert Advice view for more details about the situation event. Then, so that she can diagnose the problem with this situation, she decides to use the service-to-service topology.

Step 3. Linking directly to the Operational Flow for Application Server workspace

From the Situation Event workspace, Annette clicks the row link indicator of the *Requester* row that is highlighted in red, and directly links to the Operational Flow for Application Server workspace (using the predefined workspace link that IBM Tivoli Composite Application Manager for SOA provides), and starts to examine the services that are running on the *AppServer28* application server.

Step 4. Examining metric values

From the Operational Flow - *AppServer28* topology workspace, in the aggregate view, Annette sees all of the flows that have at least one operation instance hosted on the *AppServer28* application server. The operations that are associated with the *AppServer28* application server are highlighted in blue. Annette uses the **Zoom Slider** feature to focus on the operation that has the status indicator. She examines the metrics at the aggregate level for all the operations. Then, she positions the mouse cursor over a call relationship line, and views the Metrics notebook and the default **Chart** tab. To view the numbers behind each of the metrics, she clicks the **Table** tab and observes the following metrics:

- Maximum Response Time
- Average Message Size
- Maximum Message Size

Annette notices that the **Maximum Response Time** threshold was exceeded by a significant amount for a single service operation and that the problem is linked to this operation. The application is important, so she decides to verify the metric values over various time spans using the **Time Span** icon from the toolbar above the Operational Flow - *AppServer28* view.

Step 5. Verifying the metric values over various time spans

From the Operational Flow - *AppServer28* topology workspace, Annette clicks the **Time Span** icon from the toolbar. The Select the Time Span window opens with the **Real time** default setting activated and displaying the most recent complete interval of data. For a specific start and end time, she selects **Custom**. In the **Custom parameters** area, she selects a **Start Time** and **End Time** spanning 6 hours and clicks **OK**. The new time frame triggers a refresh of the data in the Operation Flow and Interaction Detail views. She recreates the same time frame, only now, from a day earlier. She verifies the data from the new time frame and examines the results. She is concerned about the load on the application server. She decides to open a problem ticket.

Step 6. Honoring the terms of the service level agreement and opening a problem ticket

Annette honors the terms of her service level agreement (SLA), completes her problem ticket, and forwards it to Jim for further investigation.

Making a priority assessment of problem tickets and immediately addressing the critical applications

Scenario overview

Jim, the Middleware/Application Support, Subject Matter Expert (SME), checks his incident queue to see whether any problem tickets are assigned to him. He notices several tickets, and views each one so that he can make an initial priority assessment. Some applications are more critical to certain lines of business than others. For this reason, he must first address the problems with the web services that affect the high-visibility applications.

Problem determination

Jim, the Middleware/Application Support, Subject Matter Expert (SME), notices that one of the problem tickets from Annette involves some important applications that use a set of web services and another service running on a specific application server. This ticket raises some concern, so he accepts the tickets and begins to work on the problems.

Step 1. Beginning the investigation of problem tickets

Jim normally monitors the performance views to detect potential problems. Some web services are unexpectedly overloaded with requests, explaining why some things are not running smoothly. He begins to investigate any problem tickets concerning the critical applications that use these web services.

From the Navigator Physical view of the Tivoli Enterprise Portal desktop client, he navigates to the Services Management Agent Environment node and clicks the **Performance Summary** node to open the Performance Summary workspace. He checks the web service traffic, the health and unavailability of the application server, and starts to monitor the performance.

For more information about this workspace, see “Performance Summary workspace” on page 41; for more information about this view, see “Average Response Time by Operation view” on page 40.

Step 2. Checking web service traffic and the health of the critical applications

Jim wants to assure the performance of the critical business applications and see a real-time status of the associated web services. He drills down on this problem by linking to a monitoring product from the Performance Summary workspace for further analysis. He right-clicks the *Requester* row in the Services Inventory table view, which was identified in Annette's problem ticket, and selects **Link to -> Request Analysis (WebSphere Agent)**. The Request Analysis workspace opens and is displayed under the WebSphere Agent subnode in the Navigator Physical view. In this workspace, he views response time information, application server information, and web server metrics. He sees that all is well and closes the problem ticket.

Step 3. Addressing another problem ticket about services running on a specific application server

Jim opens another problem ticket and sees that services are running on the *AppServer28* application server and that Annette viewed the data in the Situation Event workspace. He knows that he must view the application server for which the web services are running. Because the application server runs on the Windows operating system, Jim revisits his notes and knows that several minor problems in the past with critical applications running on this system occurred in the past. He goes to the Navigator Physical view, looks from the operating system level, and sees that everything looks fine, but loaded. Next, to view the services, he drills down to the specified application server.

Step 4. Navigating to the Operational Flow for Application Server workspace to examine services and their metric values

Jim goes to the Navigator Physical view and selects the Services Management Agent Environment node. In order to open the Operational Flow for Application Server workspace, he clicks **Operational Flow**. He sees all of the flows that have at least one operation instance that is hosted on *AppServer28* in the aggregate view. In total, 28 services must be examined. He starts by looking at the metrics that are associated with each service. Like Annette, he examines the application server details, metric values, and business processes.

He further examines the operation flow and notices that the **Maximum Response Time** value for the *lookupCustomer1* operation is high. He links back from the Operational Flow for Application Server workspace topology to the Performance Summary view that is filtered exactly to the operation that he looked at. He sees that the number of requests is high and is overloading the system.

Step 5. Adding another application server and rerouting web services traffic

Jim decides to add another application server to the environment and reroute the web services traffic. The system is experiencing too many service calls and the processor must be increased. Jim creates a problem ticket and identifies that a new application server with a new host name must be created for the web services environment.

Step 6. Checking the system performance and verifying that the problem ticket is deployed

When Jim is notified that the application server is online, he verifies that the critical red event indicator is eliminated from the *D4: AppServer28* node and the Windows operating system node on the Navigator Physical view.

He then closes the problem ticket.

Examining service interactions in the SOA

Scenario overview

Allen, the Services Architect, examines the current service interactions to ensure that the following criteria are met:

- They reflect his model
- They are correctly associated with the relevant business processes
- The deployments correspond with expected or observed traffic
- Any services that must be registered in WebSphere Service Registry and Repository are correctly administered

He also provides support for Annette by proactively monitoring the performance views to identify potential problems.

Problem determination

Allen, the Services Architect, oversees the entire SOA and uses the All Flows view of the service-to-service topology to find ways to improve the flow of operations. To assemble the architecture of the SOA, he relies on WebSphere Integration Developer (WID). With the architecture in operation, he decides to compare his WebSphere Integration Developer model to the actual operation to verify that all is well.

Step 1. Navigating to the all flows view to compare operations with the current model

Allen constantly tries to improve the flow of operations. He always tries to validate whether his service flows are efficient and completing. He checks an overview of all the services. In the Physical Navigator, from the **View** field, he clicks **ITCAM for SOA** and launches the custom navigator. Then he right clicks the only node in the navigator and selects **WorkspaceOperational Flows**.

The Operational Flows workspace opens. It depicts all of the operational flows in his SOA. In the top portion of the workspace, he can view the overall flow of operations, their relationships to each other, and service operation invocations as call relationships. To view invocations among service operations at the operation instance level of detail, he uses the Interaction Details pane.

He notes how some things change across the architecture and how these changes affect his model.

Step 2. Observing the entire SOA for deployed operations

Based on his knowledge of services from the last time that he observed them, Allen notices that some of the services were swapped out and replaced. This change resulted in a drop in performance and the amount of coverage that was applied to some of the most critical services. He examines the configurations that were made to the infrastructure and verifies that the expectancy level from these modifications is behaving according to design. He then moves on and looks at some other aspects of the SOA.

Step 3. Examining different aspects of the SOA, including business processes

Allen reviews the service-to-service topology display for one of the major defined SOA business processes. He knows that it is a critical process dealing with transactions. In the aggregate portion of the Operational Flows workspace, he uses the search function to search for the critical business process. He checks his notes and knows that *lookUpCustomer* is an operation that deals with customer transactions. He enters *lookU** in the **Name** field. The node matches are automatically selected for him in the view. He sees multiple matches and uses the **Find next** and **Find previous** buttons to navigate between the matches, one node at a time, until he locates the operation aggregate. He finds *lookUpCustomer* in the topology and right-clicks and selects **Show Business Processes**. In the read-only Business Process window, he sees that the business process that is associated with the *lookUpCustomer* operation is named *GetCustomerProcess* and that the accompanying business process namespace is *http://lc.retail.samples.wsm.ibm.com..* He then closes the Business Process window and returns to the topology.

Step 4. Leveraging the integration of service registry integration topology

To view the Service Overview table and see how the various services are defined, Allen clicks the link. While looking at the Services Overview table view and comparing **Observed** and **Registered** operations, he notices two kinds of discrepancies:

- Some defined services are not observed.
- Some observed services do not have a registered counterpart in the repository.

From the Service Overview table view, he notices that several services are not observed. He selects *lookUpCustomerPay*. In order to open the Service Port Details view, he right-clicks the row and selects **View Service Port Details**. He uses this view to visualize the structure of his service in regard to service ports, operations, web services, and application servers.

For the services that are not observed, he checks to see whether they were deployed to the application server. He notes that several of these fall into this category.

For the observed services that do not have a registered counterpart, Allen adds them to a list to be reviewed and processed by the change board, so that he can determine how they were deployed without being registered.

Although he is viewing a service registry integration topology definition of a service and where it is deployed, he wants to see all of the monitored operation flows for the application servers and their observed traffic patterns so that he can ensure that they are not overloaded.

He uses a link to navigate from the service registry integration topology to the Operational Flow for Application Server workspace in the service-to-service topology.

Step 5. Ensuring that the application servers are not overloaded

In the Service Port Details view, Allen right-clicks the *AppServer28* application server instance and selects **Operational Flow for Application Server** to open the Operational Flow for Application Server workspace.

He examines node details and verifies metrics. He uses the aggregate view to verify overall patterns of traffic in terms of service operations and mediations, and then uses the Interaction Details view to look at specific deployments of a particular operation. He acknowledges the operational flows that pass through the selected monitored application server runtime environment and decides to speak to Jim and other members of the development team about some issues.

Step 6. Understanding new issues and putting new processes in place

Allen is concerned that services were added to the environment and were not defined in the WebSphere Service Registry and Repository. To understand how this happened, he sets up a meeting with the development team and Jim (the Middleware Application Support SME). He wants to establish a process to prevent this issue from happening again.

As far as the services that were not observed are concerned, Allen hears back from the web services SME. The SME emphasizes that the services were not working properly and that he intends to work on getting them up and running properly as soon as possible.

Allen contemplates some new mediation ideas to simplify the development. By providing a clearer separation between the business logic and the routing and management capabilities, he thinks that he can remove some complexity from the application, and provide simple integration logic in the Enterprise Service Bus (ESB) that is consistent for all applications.

Diagnosing a process group problem

Scenario overview

Olivia receives an alert that the status of Process Group A is red. Alternatively, Olivia receives a ticket regarding slow response time with Process Group A.

Problem determination

Olivia performs the following steps:

1. Olivia navigates to the Process Group Summary view and confirms that Process Group A is red.
2. Olivia places the mouse cursor over Process Group A for more details.
3. Olivia double-clicks Process Group A and is taken to the Operation Flow for Process Group Topology.
4. Olivia places the mouse cursor over an operation with a yellow status icon (indicating a potential problem with the operation). A flyover window with details, including the Module name, Component/Import/Export name, and Component/Import/Export implementation is displayed.
5. Olivia examines the flyover window and wants more information about the calls to this operation.
6. Olivia double-clicks an operation with a status indicator of red. The topology instance view is displayed in the bottom pane.
7. Olivia places the mouse cursor over the call relationship to display a flyover window containing additional metrics.
8. If the component is a BPEL process, mediation flow, or BPD, Olivia can right click the operation instance and select **Show Details** to display a window showing the parts of the component (for example, mediation primitives in a flow) and metrics for them. For a BPEL process or SCA human task, she can also use this window to link to the BPC Explorer (if the linking is configured).
9. Olivia notices that the fault count is high, so she suspects something is wrong in the application.
10. Olivia right clicks the operation instance and chooses **Link To...** She is presented with a list of workspaces or other interfaces that can help her find the information she needs to solve the problem or hand it off to the appropriate SME.
11. Because Olivia suspects a problem with the application itself, she chooses the **Server Summary** link.
12. Olivia is taken to the Business Process Management Summary page, which displays the status of the BPM application server. She confirms her suspicions that an application problem exists. She routes the ticket to Miles, the Subject Matter Expert (SME) with her findings so far, including the name of the troubled ear file.

BPM impact analysis

Scenario overview

Olivia receives an alert indicating that Application Server 1 is unhealthy.

Problem determination

Olivia performs the following steps:

1. Olivia navigates to the Application Server Summary workspaces and sees the status icon for Application Server 1 is red.
2. Olivia wants to determine the priority of fixing the Application Server problem. She right-clicks the application icon and links to the ITCAM for SOA Service Group Summary workspace.
3. The ITCAM for SOA Service Group Summary workspace shows that five different business processes depend on the health of Application Server 1. Olivia realizes that a high priority problem exists and immediately begins to troubleshoot.

Additional usage scenarios and task examples

Additional scenarios and examples of how to use the latest features in this release of ITCAM for SOA can be found elsewhere in this guide. For illustrated examples of the following tasks, see Chapter 8, “Workspace for monitoring service health,” on page 149:

- Creating groups
- Displaying groups in the Group Summary workspace
- Configuring for unavailability.
- Including the Group Summary view in a workspace

For examples of completing the following tasks, see Chapter 9, “Creating custom workspaces and links,” on page 185:

- Creating a dynamic workspace link to a workspace containing an Operational Flows workspace.
- Creating a dynamic workspace link from Situation Event Results workspace to a new workspace that contains an Operational Flow view.

.

For examples showing how to complete the following tasks, see Chapter 10, “Situations,” on page 199:

- Creating your own situations.
- Creating custom situations for measuring service unavailability

For examples of how to calculate status for operation aggregates and groups, see Appendix B, “Determining status for operation instances, operation aggregates, and groups,” on page 331.

Chapter 3. Workspaces

Workspaces in Tivoli Enterprise Portal provide access to the collected data for your monitored services. Each workspace serves a unique purpose and displays a specific set of data that is collected as a result of monitoring parts of a resource. Workspaces also provide an overview of all resource data. To help you monitor your web services, ITCAM for SOA provides predefined workspaces that display monitoring data in topology graphs, tables, and charts.

Workspaces are split into *views* that display data in a meaningful way. A view can be a table, a graph or chart, a notepad, a web browser session, an event console, or a Take Action view from which you can send commands to the operator console.

One of the views typically displayed in the Tivoli Enterprise Portal is referred to as the *Navigator*. Use the Navigator to select a workspace. When you select items in the Navigator, the workspace displays views that are relevant to your selection. The Navigator view includes the following tabbed options:

- “The Navigator Physical view” on page 24
- “The Navigator ITCAM for SOA view” on page 26

You can access additional workspaces using link menus within the views and within the Navigator itself.

Some workspaces contain table views. The columns in the table show the metrics that are collected by the monitoring agents and displayed in the workspace. The rows in the table can contain links to related workspaces that provide more detailed information.

Data values that fall outside established thresholds are highlighted in the views with a colored background or icon. Scroll bars are tinted if table cells outside the viewable portion of the table contain attribute values outside the thresholds. Hover help is available to provide more information while you move your cursor over table cells and column headings.

In topology views, some operation instance objects are highlighted in the display, for one of several reasons:

- The operation or an instance of the operation aggregate is hosted on a selected application server.
- The operation aggregate is a member of a selected service or process group.
- The operation is the target of a workspace link.
- The operation is displayed as a result of the user refocusing the topology display.

The reason for highlighting is included in the flyover window for each object. In the same way, if an operation is shown as disabled, the flyover window explains that the associated agent is offline.

Predefined workspaces

ITCAM for SOA provides a set of predefined workspaces that operate in Tivoli Enterprise Portal.

The following predefined workspaces are included with ITCAM for SOA:

- “Application Server Services Management workspace” on page 39
- “DataPower Console workspace” on page 46
- “Faults Summary workspace” on page 44
- “Message Arrival workspace” on page 37
- “Message Summary workspace” on page 42
- “Operational Flows workspace” on page 92
- “Operational Flow for Application Server workspace” on page 95
- “Operational Flow for Operation workspace” on page 93
- “Performance Summary workspace” on page 41
- “Performance Summary for Requester Identity workspace” on page 51
- “Requester Identities for Operation workspace” on page 50
- “Requester Identity Monitoring Configuration workspace” on page 49
- “The Group Summary workspace” on page 152
- “Services Management workspace” on page 55
- “Services Management Agent workspace” on page 36
- “Services Management Agent Environment workspace” on page 40

The Navigator Physical view

The Navigator Physical view provides the hierarchical view of your enterprise. You can see a high-level overview of the status of your network environment. You can navigate to specific monitored resources to check activity and investigate problems. At every level, event indicators alert you to changes in system or application conditions.

The Navigator Physical view displays your enterprise as a map of operating systems, agents, and monitored resources. Figure 3 on page 25 is an example of a typical Navigator Physical view.

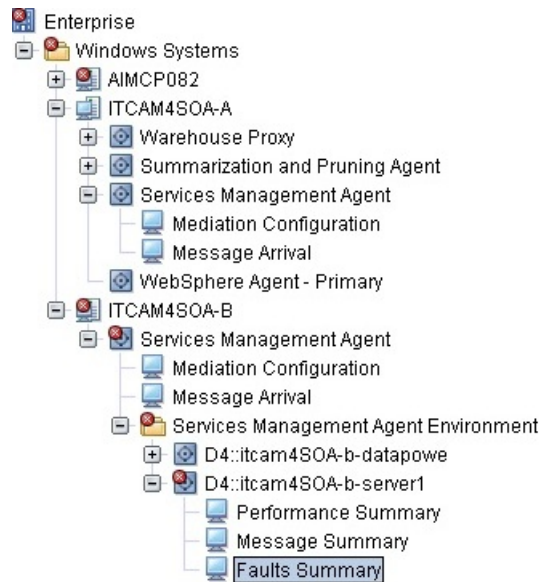


Figure 3. Sample Navigator Physical view

When you click the Enterprise node at the top level of the Navigator Physical view, the Enterprise Status workspace is displayed (see “Enterprise Status workspace” on page 35). This node represents the domain of all of the managed systems that are supported by this Tivoli Enterprise Monitoring Server. For more information about this workspace, see the help for Tivoli Monitoring.

Nodes under the Enterprise node represent the monitored hosts, organized by operating system. For each host where ITCAM for SOA is installed, you can access the Services Management Agent node. Click the node to access the Services Management Agent workspace (see “Services Management Agent workspace” on page 36). You can also right-click the node and select the Requester Identity Monitoring Configuration workspace (see “Requester Identity Monitoring Configuration workspace” on page 49).

The following ITCAM for SOA nodes are located under this node:

- Message Arrival. Click the node to access the Message Arrival workspace (see “Message Arrival workspace” on page 37).
- Services Management Agent Environment. The content of this node depends on whether one or more than one ITCAM for SOA data collectors are installed on the host.

If only one data collector is installed, click the Services Management Agent Environment node to display the Application Server Services Management workspace (see “Application Server Services Management workspace” on page 39). You can also right-click the node and select the Operational Flow for Application Server workspace link (see “Operational Flow for Application Server workspace” on page 95). Under the node, there are three additional nodes, providing access to their own workspaces:

- Performance Summary (see “Performance Summary workspace” on page 41)
- Message Summary (see “Message Summary workspace” on page 42)
- Faults Summary (see “Faults Summary workspace” on page 44)

If more than one data collector is installed, click the Services Management Agent Environment node to display the “Services Management Agent Environment workspace” on page 40. Subnodes under this node represent the data collectors,

and are labeled with unique names in the format *D4:<computer>-<server>*. Each subnode displays its own Application Server Services Management default workspace. You can also right-click the subnode and select the Operational Flow for Application Server workspace link (see “Operational Flow for Application Server workspace” on page 95). Under each subnode, you can find the Performance Summary, Message Summary, and Faults Summary nodes for the data collector.

From a workspace link in the Services Inventory table view of the Performance Summary workspace, you can access the following workspaces:

- Requester Identities for Operation
- Operational Flow for Operation
- Operational Flow for Application Server

From a workspace link in the Requester Identities for Operation workspace, you can access the Performance Summary for Requester Identity workspace.

The Navigator ITCAM for SOA view

Use the ITCAM for SOA Navigator view to access workspaces that cover information from your entire SOA environment.

The view contains one Services Management node. You can use the node to access these predefined workspaces:

- The Group Summary workspace displays a high-level summary view of the overall health of service and process groups. You can organize services in your environment into groups, and view the overall message volume, service unavailability, performance, and status for each group. See Chapter 8, “Workspace for monitoring service health,” on page 149. To access the Group Summary workspace, click the Services Management node.
- The Services Management workspace displays WebSphere Service Registry and Repository integration topology data. It includes service registry integration (static) topology and table views. The views show the relationship of SOA service definitions, their deployments, and relationships to business processes as defined in Discovery Library Adapter (DLA) books. See Chapter 6, “Workspace for service registry integration,” on page 53. To access the Services Management workspace, right-click the Services Management node, and select **Workspaces > Services Management**.
- The Operational Flows workspace displays all operation interactions in your environment in a *service-to-service topology* format. See Chapter 7, “Workspaces for service-to-service topology,” on page 71. To access the Operational Flows workspace, right-click the Services Management node, and select **Workspaces > Operational Flows**.

Accessing the Navigator ITCAM for SOA view

To view the Navigator ITCAM for SOA view, you can select its tab in the Navigator window. If the tab is not present, select **ITCAM for SOA** in the **View** list in the Navigator window.

However, this view might not be available in the list when you log in to Tivoli Monitoring. In this case, you have to assign this view to your user name.

To assign the Navigator ITCAM for SOA view to your user name, complete the following steps:

1. From the Tivoli Enterprise Portal toolbar, click **Administer Users**. The Administer Users window is displayed.
2. In the **Users** area, find your user ID and highlight it in the table.
3. To display the list of assigned and available views, click the **Navigator Views** tab.
4. In the **Available Views** area, select the **ITCAM for SOA** view. Then, click the arrow to move the view into the list of **Assigned Views**.
5. Click **OK**.

For more information about assigning views to your user name, see your Tivoli Enterprise Portal documentation. If you do not have access to this function, contact your Tivoli Enterprise Portal Server administrator for assistance.

Accessing workspaces by using links

You can access additional workspaces by using links from other workspaces. To access the links, either click the workspace link icon, or right-click rows in tables and icons in topologies and select the **Link To** menu.

The following examples show typical uses of workspace links.

Analyzing a triggered situation

You might observe a situation event that is triggered because one or more data collector metric values exceeded a threshold. To display the situation event hover help for the node, in the Navigator Physical view, hold your cursor over the event indicator icon.

In the hover help, to link to the Situation Event Results workspace, click the link indicator.

In the Situation Event Results workspace, examine the rows in the Initial Situation Values table view. If the situation is based on the ITCAM for SOA Services Inventory_610 or Services Inventory Requester Identity_610 attribute groups, it has columns for the service port name and namespace, operation name and namespace, and the Service Type. The Service Type column contains a value of either *Provider* or *Requester*.

After you examine the metrics that caused the situation event, you can investigate the possible cause of the situation event:

1. Click the link indicator in the row of interest.
2. Select one of the following links:
 - For Service Type of *Provider*, Operational Flow for Operation.
 - For Service Type of *Requester*, Operational Flow for Application Server.

The operational flow workspace is then displayed. You can use those topology views to identify specific operational flows for further investigation.

These workspace links are predefined for the situations that are provided with this release. If you define your own situations, you must also create these workspace links when required for your environment. For details about the Situation Event Results workspace, refer to your IBM Tivoli Monitoring documentation. To see examples, see the links that are provided with the predefined situations.

Chapter 9, “Creating custom workspaces and links,” on page 185 also has an example of how to create links to Operational Flow workspaces.

See “Linking from the Situation Event Results workspace” on page 195 for information about creating links from the Initial Situation Values view of the Situation Event Results workspace to a new workspace. You can use this same procedure to create links from the Initial Values view to the predefined Operational Flow for Operation and Operational Flow for Application Server workspaces, specifying those workspaces as the target workspace rather than a new workspace.

You can use the same procedure to create a link for a row in the Current Situation Values view in the Situation Event Results workspace to the Operational Flow for Operation and Operational Flow for Application Server workspaces, by starting the Link wizard from the Current Situation Values view.

Examining a single operational flow from the Operational Flows workspace

When you display the Operational Flows workspace, all of the monitored flows in the environment are displayed. The resulting topology view might show many operation aggregates and operational flows, and you might want to examine a particular operational flow.

To examine a single operational flow, you have to populate the Interaction Detail portion of the view. To populate it, use one of the following methods:

- Double-click the operation aggregate icon.
- Right-click the operation aggregate icon and select **Show Interaction Detail**.

In the Interaction Detail part of the view, right-click an operation instance and select the **Link To -> Operational Flow for Operation** workspace link. A single operational flow is displayed.

Examining a single operational flow from the Operational Flow for Application Server workspace

When you display the Operational Flow for Application Server workspace, all of the monitored flows that contain an operation in the selected application server runtime environment are displayed. The resulting topology view might show many nodes and operational flows, and you might want to examine a particular operational flow.

To examine a single operational flow, you have to populate the Interaction Detail portion of the view. To populate it, use one of the following methods:

- Double-click the operation aggregate icon.
- Right-click the operation aggregate icon and select **Show Interaction Detail**.

In the Interaction Detail part of the view, right-click an operation instance and select the **Link To -> Operational Flow for Operation** workspace link. A single operational flow is displayed.

Viewing requester IDs for an operation instance

If you monitor web services for a particular requester identity, you might want to view the requester identities associated with an operation instance in any of the Operational Flow workspaces. For more information about requester identities, see “Requester Identity Monitoring Configuration workspace” on page 49.

From the Operational Flow workspace, right-click the operation instance and select the **Requester Identities for Operation** workspace link. This workspace displays the list of monitored requester identities that called the selected operation instance.

Important: To view the associated requester identity list, the monitoring feature to monitor by requester identity must be enabled.

Linking to performance summary data from an Operational Flow workspace

When you view topology data in an Operational Flow workspace, to understand the web services that are being monitored, or to help you resolve problems, you might want to examine additional information in metrics, bar charts, and table views.

In any Operational Flow topology workspace, to examine such information, do the following steps:

1. Select an operation instance icon.
2. Click the link to the Performance Summary workspace under the appropriate Services Management Agent Environment node or *D4*: subnode in the Navigator Physical view.

The resulting Services Inventory table view contains monitored data associated with your selected operation instance.

If you select a DataPower mediation operation instance and that operation instance is being monitored for multiple DataPower display groups or subnodes, you are prompted to select the Performance Summary workspace for the available subnodes that you want to access.

Link from a Performance Summary workspace to operational flows and requester identity workspaces

In the Performance Summary workspace, you can access other workspaces from the Services Inventory table view. The available workspaces depend on whether the Service Type is *Requester* or *Provider*.

- If the **Service Type** column has a value of *Requester*, you can click the Operational Flow for Application Server link.
- If the **Service Type** column has a value of *Provider*, you can click either the Operational Flow for Operation link or the Requester Identities for Operation link. From the Requester Identities for Operation workspace, you can click other links to access the Performance Summary for Requester Identity workspace.

Linking to other monitoring agent workspaces

ITCAM for SOA enables workspace links to other monitoring agents through the use of menu options from the Performance Summary workspace and all Operational Flow workspaces. You can improve your problem analysis by linking to other monitoring products that might have data that helps you to isolate a problem.

Links to other monitoring agent workspaces are available if the application support for the other monitoring agent is installed on the Tivoli Enterprise Portal Server.

IBM Tivoli Monitoring verifies whether a monitoring product is installed. Tivoli Monitoring manages the links by filtering out any workspace links that reference

the workspaces for monitoring products that are no longer installed. However, if another monitoring product is installed, but the agent is not deployed to a particular computer, it is possible that the workspace link is still displayed in the **Link To** menu. In this case, when you select the link, an error message is displayed stating that the target workspace cannot be found.

For example, ITCAM Agent for WebSphere Applications is installed on *MachineA*, but not *MachineB*. ITCAM for SOA provides a workspace link from the Interaction Detail portion of the Operation Flow view to the ITCAM Agent for WebSphere Applications workspace. In the Interaction Detail portion of the view, you select an operation instance that was discovered on *MachineB*. The **Link To** menu for that operation instance lists the ITCAM Agent for WebSphere Applications workspace. However, when you click that link, an error message is displayed because ITCAM Agent for WebSphere Applications is not deployed on *MachineB*.

ITCAM for SOA links to the supported monitoring agents on the same computer as the application server that is being monitored by the ITCAM for SOA data collector. The *Local IP Address* value is used in the link definition as the context for the IP address.

The following workspace links are available for the supported monitoring products:

- IBM Tivoli Monitoring operating system agent workspaces:
 - Windows OS agent
 - Linux OS agent
 - UNIX OS agent
- ITCAM for WebSphere:
 - WebSphere Agent
 - Request Analysis
 - Application Health Status
- ITCAM Agent for J2EE:
 - JBoss Application Server
 - Request Analysis
 - Application Health Status
 - BEA WebLogic Application Server
 - SAP NetWeaver Server
 - WebSphere Application Server Community Edition

Multiple application servers: If several application servers are defined under the agent node, the IBM Tivoli Monitoring workspace selection window prompts you for a specific application server.

- ITCAM Agent for WebSphere Applications:
 - WebSphere Agent
 - Request Analysis
 - WebSphere Agent Summary
 - Server Summary

Multiple application servers: If several application servers are defined under the agent node, the IBM Tivoli Monitoring workspace selection window prompts you for a specific application server.

Server Summary: If ITCAM Agent for WebSphere Applications or ITCAM Agent for J2EE is installed, you can access the Business Process Manager Summary

workspace. These agents are available in ITCAM for Applications and ITCAM for Application Diagnostics. They provide the operational monitoring for WebSphere Application Servers and other Java EE servers. For more information about this link, see “From a node in the Interaction Detail view to the Business Process Manager Summary workspace” on page 99.

- IBM OMEGAMON® XE for z/OS
 - Address Space Overview
- IBM OMEGAMON XE for Messaging
 - OMEGAMON XE for Messaging Message Flow Statistics

Multiple message brokers: If several message brokers are defined under the agent node, the Workspace selection window prompts you for the specific broker. The resulting workspace is displayed for your selected message flow name.

ITCAM for SOA and IBM Tivoli OMEGAMON XE for Messaging monitor message flows through the WebSphere Message Broker. The same message flow name is observed in both environments. This name is derived from the **Service Port Name** column in the Services Inventory_610 attributes table or from the service port name attribute for an operation instance in the Operational Flow workspaces. For operations monitored on non-SOAP WebSphere Message Broker nodes, ITCAM for SOA passes the name to the Message Flow Statistics workspace in IBM Tivoli OMEGAMON XE for Messaging.

For operations that are monitored on WebSphere Message Broker SOAP nodes, ITCAM for SOA passes the execution group name to the Message Flow Statistics workspace. The execution group name is derived from the Application Server Name column in the Services Inventory_610 attributes table or from the attributes for an operation instance in the Operational Flow workspaces.

This workspace accepts the message flow name or execution group name and filters the view to display only data that is related to that specific message flow name or execution group name.

Linking to the DataPower WebGUI console

If you view a DataPower mediation operation instance on any Operational Flow workspace, you can access the DataPower WebGUI console within the Tivoli Enterprise Portal. The DataPower Console workspace is displayed for the DataPower appliance that is being monitored by the Tivoli Enterprise Monitoring Agent.

If several display groups or subnodes are monitoring the DataPower mediation operation instance, you are prompted to select the D4: subnode that you want to associate with the DataPower Console workspace.

You can also link to the DataPower WebGUI Console from any row of the Services Inventory view of the Performance Summary workspace where the Application Server Environment column contains the *DataPower* value, and the Performance Summary workspace is showing data for a single DataPower appliance.

For more information about the DataPower WebGUI console workspace and the link symbols that it supports, see Chapter 4, “Workspaces for configuration and metric summaries,” on page 35.

Linking from a Services Management topology view to the Operational Flow for Application Server workspace or Performance Summary workspace

You might be viewing a definition of a service and where it is deployed in the Services Management workspace that displays service registry integration, and you want to see the monitored operational flows or performance statistics for the application server.

To view the information, from a specific application server instance, you can link from one of the following items in the Services Management workspace:

- A row in the the Services Overview table.
- An application server instance in the Service Details topology view.
- A service port instance in the Service Details topology view.
- An application server instance in the Service Port Details topology view.
- A service port instance in the Service Port Details topology view.

To view the monitored operational flows for the application server, select the workspace link to the Operational Flow for Application Server workspace.

To view the performance statistics for the application server, select the workspace link to the Performance Summary workspace.

Viewing historical data

You can view historical data in Tivoli Enterprise Portal if history collection is configured and enabled for the attribute groups used by each of these ITCAM for SOA workspaces and views:

- Performance Summary workspace
- Performance Summary for Requester Identity workspace
- Message Summary workspace
- Operational Flows workspace
- Operational Flow for Application Server workspace
- Operational Flow for Operation workspace
- Operation Flow for Service Group view
- Operation Flow for Process Group view

For procedures to configure historical data collection for ITCAM for SOA attribute groups, see the *IBM Tivoli Composite Application Manager for SOA Installation Guide*.

In addition to these Tivoli Enterprise Portal workspaces, the Eclipse-based IBM Web Services Navigator tool that is provided with ITCAM for SOA helps you to see a cross-computer topology view of the message traffic between services in the SOA environment. This tool retrieves and combines data from log files on the various monitored computers in your environment, or from historical data that was written into the Tivoli Data Warehouse. For more information about the IBM Web Services Navigator tool, see the *IBM Tivoli Composite Application Manager for SOA Tools* guide.

Workspace and link version dependencies

The following workspaces and the links to them are available only for ITCAM for SOA monitoring agents version 6.1.0 fix pack 1 or later:

- The Requester Identities for Operation workspace
- The Requester Identity Monitoring Configuration workspace
- The Performance Summary for Requester Identity workspace

The following workspaces and the links to them are used with data only from ITCAM for SOA monitoring agents version 7.1.0 or later:

- The Operational Flows workspace
- The Operational Flow for Application Server workspace
- The Operational Flow for Operation workspace
- The Operation Flow for Service Group view
- The Operation Flow for Process Group view
- The Group Summary workspace
- The DataPower Console workspace

Predefined workspace links from rows in the Services Inventory table are available only for ITCAM for SOA monitoring agents version 7.1.0 or later.

The **Server Summary** link to ITCAM Agent for WebSphere Applications is available only for ITCAM for SOA monitoring agents version 7.2 or later.

Chapter 4. Workspaces for configuration and metric summaries

ITCAM for SOA provides a set of predefined workspaces that contain information about application servers and metrics for services running on them. You can also use these workspaces to configure ITCAM for SOA monitoring on the servers.

To access these workspaces, use the Navigator Physical view. For details, see “The Navigator Physical view” on page 24.

Enterprise Status workspace

The Enterprise Status workspace gives you a high level, enterprise-wide summary of recent message and situation activity.

Access the workspace by clicking the Enterprise node, which is at the highest level of the “The Navigator Physical view” on page 24.

The following views are included:

Situation Event Console

Displays a row for situations where the status changes to any of the following events:

- Open
- Closed
- Problem
- Acknowledged

To see the Event Details - Similar by Situation Name workspace, click a row link indicator. To see the other Event Detail workspaces, right click this indicator.

My Acknowledged Events

Displays the events that are assigned to the current user. This view shows both open and closed events.

To see the Event Details - Similar by Situation Name workspace, click a row link indicator. To see the other Event Detail workspaces, right click this indicator.

Open Situation Counts

Displays a bar chart for every situation in your monitored network that has become true in the past 24 hours. The bar size indicates the number of times a situation event has been opened during this time.

Message Log

Displays a row for situations where the status changes to any of the following events:

- Open
- Closed
- Problem
- Acknowledged

You can also right click the Enterprise node and select the Managed System Status workspace. This workspace lists all the monitoring agents on your managed network and shows their status.

Services Management Agent workspace

The Services Management Agent workspace displays the current configuration details for the ITCAM for SOA data collectors that are configured in application server instances on a host.

To access the workspace, in the Navigator Physical view, click a Services Management Agent node.

You can update the configuration data that is displayed in this workspace using Take Action commands that are provided with the agent. For more information, see Chapter 11, “Take Action commands,” on page 223.

The **Data Collector Global Configuration** table includes the environment and name of the application server where the data collector is running, and flag settings to enable or disable data collection, tracing, and logging. In this table, you can take the following actions:

- Turn monitoring off by using the “DisableDC_610 Take Action command” on page 243
- Turn monitoring on by using the “EnableDC_610 Take Action command” on page 245
- Use the “updateLogging_610 Take Action command” on page 249 to control the value in the **Debug Log Level** column
- Use the “updateTracing_610 Take Action command” on page 251 to control the value in the **Data Collector Tracing On/Off** column

The **Data Collector Monitor Control Configuration** table includes information about the service ports and operations that are monitored for a specific application server in the specified environment. The **Message Logging Level** column indicates the level of logging that is configured for each unique combination of service ports and operation. Valid message logging levels are:

Table 4. Message logging levels

Message Logging Level	Comment
None	No information about the message is logged.
Body	Only the body of the message is logged.
Header	Only the header portion of the message is logged.
Full	Both the header and body of the message are logged.

By default, all services are monitored. To change the monitoring settings, you can take the following actions:

- Use the “AddMntrCntrl_610 Take Action command” on page 229 to enable monitoring for a port name and operation name. This command also sets the logging level.
- Use the “DelMntrCntrl_610 Take Action command” on page 241 to disable monitoring for a port name and operation name.
- Use the “UpdMntrCntrl_610 Take Action command” on page 252 to change the logging level for a service port and operation that are already monitored.

The Data Collector Monitor Control Configuration table must always have at least one entry. If there is only one entry in the table and you delete it using the `DelMntrCntrl_610` Take Action command, the operation completes successfully with a return code of 0, but the default configuration is automatically restored to the table to ensure that at least one monitor control is in effect at all times.

The **Data Collector Filter Control Configuration** table includes filtering information for specific combination of service port name, operation name, and client IP addresses. The only available action is *reject*, which rejects messages that match the filtering criteria and acts as a selective control over which messages are allowed to pass through the interception point. The rejection causes a SOAP fault to be sent back to the calling service.

Use the “AddFltrCntrl_610 Take Action command” on page 226 to add a filter. Use the “DelFltrCntrl_610 Take Action command” on page 238 to remove a filter.

Data collector filter controls are not supported for DataPower and WebSphere Message Broker application server environments. To filter messages in these environments, use the filtering features that are provided with these products.

Data collector filter controls cannot be used to reject messages for SCA components deployed to IBM Business Process Manager Server, WebSphere Process Server, or WebSphere Enterprise Service Bus.

Message Arrival workspace

The Message Arrival workspace provides a summary of message arrival situations.

To access the workspace, in the Navigator Physical view, open the Services Management Agent node and click the Message Arrival node.

The views that are provided with this workspace display the current count of arriving messages that are observed by the data collectors for each application server.

About message arrival situations

A typical message arrival situation tracks the number of messages that arrive from a combination of service name, operation name, service port namespace, operation namespace, and remote IP address, during a specified time interval. If the number of messages exceeds a predefined threshold during that time interval, the situation is triggered. For example, this situation might be used in a denial of service attack scenario, in which an unusually large amount of message traffic arrives from a source. You might define a workflow policy to automatically handle the triggering of this situation, rejecting the flow of messages from that particular source until the amount of message traffic returns to normal.

When triggered, a message arrival situation emits a pure event and the situation remains active until you reset it. You can automate this reset process by creating a second message arrival situation that is triggered when message traffic falls under a specified threshold for a specified time interval, when traffic returns to normally expected levels. This second situation is configured to clear the previously triggered situation.

This is how the predefined situations, named `MessageArrivalCritical_610` and `MessageArrivalClearing_610`, are designed to operate. Importantly, these

predefined situations are initially configured to monitor *all* combinations of service port name, operation name, service port namespace, operation namespace, and remote IP address, with nominal threshold and time interval settings. To use the Message Arrival workspace, you must use these predefined situations as templates to create situation pairs (one for triggering when excess traffic occurs, and the other for clearing the situation when traffic returns to expected levels) that suit your monitored environment. You can create as many situations and situation pairs as you require. Each situation or situation pair that you create is configured with unique triggering thresholds and time intervals for your environment to monitor a specific combination of service port name, operation name, service port namespace, operation namespace, and remote IP address.

For more information about the predefined Message Arrival Critical and Message Arrival Clearing situations, see:

- “The MessageArrivalCritical_610 situation” on page 207
- “The MessageArrivalClearing_610 situation” on page 205

Message arrival situations in the workspace

Create one or more message arrival situation pairs by copying the predefined MessageArrivalCritical_610 and MessageArrivalClearing_610 situations and modifying the parameters to create new situations for your environment. When you distribute these situations to your managed systems, they are displayed as rows in the Message Arrival Details table view. Initially, the predefined MessageArrivalCritical_610 and MessageArrivalClearing_610 situations are automatically distributed and are displayed in the view. Additional situations that you create and distribute to this managed system are also displayed in the table view. Keep in mind the following considerations:

- Message arrival situations that you create are not included in the Message Arrival Details table view until you distribute them to the managed system.
- When you create message arrival situations, use **all** of the same attributes from the Message Arrival Threshold_610 attribute group that are used in the predefined MessageArrivalClearing_610 or MessageArrivalCritical_610 situations. You can do so by making a copy of the predefined situations and modifying it. This action ensures that all of the required attributes are used and that the monitoring criteria are correctly displayed as rows in the Message Arrival Details table view.
- After making copies of the predefined MessageArrivalClearing_610 or MessageArrivalCritical_610 situations and creating and distributing your own message arrival situations, you must remove these predefined situations from the distribution so that they are no longer displayed in the Message Arrival Details table view and influence the data that is displayed in this workspace.
- **Creating situations using delta and percent functions:** If you create a situation using the Current Message Count attribute in the Message Arrival Threshold_610 attribute group, you cannot use delta or percent functions. The monitoring agent supports only comparisons against the actual value. If you create and distribute such a situation using delta or percent functions, the situation is not displayed in the Message Arrival Details view and is ignored.
- When you create message arrival situations, be aware that certain combinations of service port name, operation name, service port namespace, operation namespace, and remote IP address are not valid. For the valid combinations of these attributes for which you can create situations, see Table 36 on page 227.

Message Arrival Details table view

The Message Arrival Details table view displays the details of the predefined Message Arrival Critical and Message Arrival Clearing situations that are enabled for this agent. The table displays which combinations of service port name, operation name, namespaces, and remote IP address are included in the counts, the actual number of messages that are currently arriving, and the threshold settings and time intervals for the situations.

For more information about the attributes that are displayed in this table view, see “Message Arrival Threshold_610 attributes” on page 269.

Message Arrival by Service view

The Message Arrival by Service view displays a bar graph showing the number of messages that are currently arriving (within the defined time interval) for each situation that is displayed in the Message Arrival Details table view, sorted by service port name. If the service port name is represented by the asterisk (*) wildcard character in the Service Port Name column of the Message Arrival Details table view, this wildcard character is also displayed as the service port name in the Message Arrival by Service view. The bar chart displays the message count for each service port name twice: once for the provider, and once for the requester.

The data displayed in this bar chart is updated after every 30-second sampling interval. If you refresh this workspace after an interval where no message traffic was detected, no data is displayed in the bar chart.

Message Arrival by Operation view

The Message Arrival by Operation view displays a bar graph showing the number of messages currently arriving (within the defined time interval) for each situation displayed in the Message Arrival Details table view, sorted by operation name. If the operation name is represented by the asterisk (*) wildcard character in the **Operation Name (Unicode)** column of the Message Arrival Details table view, this wildcard character is also displayed as the operation name in the Message Arrival by Operation view. The bar chart displays the message count for each operation name twice: once for the provider, and once for the requester.

The data displayed in this bar chart is updated after every 30-second sampling interval. If you refresh this workspace after an interval where no message traffic was detected, no data is displayed in the bar chart.

Application Server Services Management workspace

The Application Server Services Management workspace displays summary message and performance information for all services monitored by the data collector.

To access the workspace, in the Navigator Physical view, click the Services Management Agent Environment node if there is only one data collector on the host. If there is more than one data collector, you can access this workspace by clicking the D4 subnode representing a data collector.

The views in the workspace are based on the Services Inventory_610 table (see “Services Inventory_610 attributes” on page 271). This table contains summarization calculations for the monitored services traffic. The calculations are done for each monitoring interval. For each interval, the traffic is analyzed to

calculate the fields that are available within the table. After the monitoring interval is complete, the record is marked as complete. All queries of the Services Inventory table include a check for complete records only.

Average Response Time by Operation view

The Average Response Time by Operation view displays the average response time, in milliseconds, for each unique combination of service port and operation being monitored, with the message being intercepted as it leaves the server or as the client response is received. This view is also displayed in the Performance Summary workspace. For more information about this view, see “Performance Summary workspace” on page 41.

Number of Messages by Operation view

The Number of Messages by Operation view displays a bar chart of the number of messages that were received, sorted by operation name. This view is also displayed in the Message Summary workspace, but the data is shown for each unique combination of service port name, operation name, and service type (provider or requester). For more information about this view, see “Message Summary workspace” on page 42.

Average Message Size by Operation view

The Average Message Size by Operation view displays a bar chart of the average size, in bytes, of messages per operation name. This view is also displayed in the Message Summary workspace, with the data shown for each unique combination of service port name, operation name, namespaces, and service type (provider or requester).

Services Management Agent Environment workspace

The Services Management Agent Environment workspace displays aggregate summary information for services monitored by all the data collectors on the host.

To access the workspace, in the Navigator Physical view, click the Services Management Agent Environment node if there is more than one data collector on the host.

The workspace is similar to the Application Server Services Management workspace, with the following exceptions:

- The title bar is shown as *Services Management Agent Environment* instead of *Application Server Services Management*.
- The Number of Messages by Operation view is replaced by the Fault Summary by Operation view.
- The data that is displayed in the bar charts is a summary of the data for all of the data collectors for that Tivoli Enterprise Monitoring Agent.

Average Response Time by Operation view

The Average Response Time by Operation view displays the average response time, in milliseconds, for each unique combination of service port, operation, and service type (*Provider* or *Requester*) being monitored, with the message being intercepted as it leaves the server or as the client response is received. This view is also displayed in the Performance Summary workspace. For more information about this view, see “Performance Summary workspace” on page 41.

Fault Summary by Operation view

The Fault Summary by Operation view displays a bar chart summarizing the number of faults for each unique combination of service port, operation, and service type (*Provider* or *Requester*).

Average Message Size by Operation view

The Average Message Size by Operation view displays a bar chart of the average size, in bytes, of messages per operation name. This view is also displayed in the Message Summary workspace, with the data shown for each unique combination of service port, operation, and service type (*Provider* or *Requester*).

Performance Summary workspace

The Performance Summary workspace provides an inventory of currently active and monitored web services, a bar graph showing the average response time (as measured by elapsed time) of the services, and a detailed table view of the individual response times for each combination of service port and operation.

To access the workspace, in the Navigator Physical view, select a Performance Summary node for a data collector.

Average Response Time by Operation view

The Average Response Time by Operation view displays the average response time, in milliseconds, for each unique combination of service port, operation, and service type (*Provider* or *Requester*) that is being monitored, with the message being intercepted as it leaves the server or as the client response is received. This view is also displayed in the Services Management Agent Environment workspace (for more information, see “Services Management Agent Environment workspace” on page 40).

If the requester and provider are both in the same application server, this bar chart displays each operation twice: once for the provider, and once for the requester. You can display historical data in this view, but be aware that historical data is displayed in multiple rows in the bar chart for each time interval in the requested time span.

When the monitoring agent is first started, at least one monitoring interval must be completed before any data is displayed in the view. After the first monitoring interval is complete, this bar chart displays the average response time for all services that had a valid average response time within the monitoring interval. If the web service has no traffic during the monitoring interval, then the average response time is indicated with a value of -1 and is not included in the bar chart. For more details on the data that is displayed in this view, see “Services Inventory_610 attributes” on page 271.

Important: For Business Process Definitions (BPDs), the response time is always measured as 0.

Services Inventory view

The Services Inventory view provides a table view of the service ports and their associated operations, and information about the associated application server, and the associated system. This table also displays the number of messages within each monitoring interval that have valid response times, and additional data, including

average response time, maximum response time, minimum response time, and standard deviation for the average response time.

This table displays only the data for the most recently completed 5-minute interval. To create a query, you can copy the query that produces the display, and then modify the new query by removing the interval completion check to show the current interval in addition to the completed interval. You can display historical data for this view for each 5-minute interval that occurred within the specified start and end time. Consult your IBM Tivoli Monitoring documentation for assistance with these Tivoli Enterprise Portal tasks.

When the monitoring agent is first started, at least one 5-minute interval must be completed before any data is displayed in the view. After the first 5-minute interval is complete, this table displays the inventory of services, and the performance-related numbers from the Services Inventory table. If the service has no traffic during the interval, then the average response time is indicated with a value of -1. For more details about the data that is displayed in this view, see “Services Inventory_610 attributes” on page 271.

The data in this table is based on the time at the monitoring agent, which might not be the same as the time at the hub Tivoli Enterprise Monitoring Server. Always ensure that your computer system clocks are synchronized. The beginning and ending times that specify the monitoring interval are expressed in Greenwich Mean Time (GMT).

Important: For Business Process Definitions (BPDs), values in the following columns are always set to 0:

- Elapsed Time Message Count
- Average Elapsed Message Round Trip Time
- Max Elapsed Time
- Min Elapsed Time
- Elapsed Message Round Trip Time Std Dev

Message Summary workspace

The Message Summary workspace provides the details on the number of messages and the average size, in bytes, of the messages, by service port name, operation name, and type (requester or provider).

To access the workspace, in the Navigator Physical view, select a Message Summary node for a data collector.

The data for this workspace comes from the Services Inventory_610 attribute table (for more information, see “Services Inventory_610 attributes” on page 271).

Number of Messages by Service:Operation:Type view

The Number of Messages by Service:Operation:Type view displays the number of messages received for each unique combination of service port, operation, and service type (*Provider* or *Requester*).

If the requester and provider are both in the same application server, this chart displays each operation twice: once for the provider, and once for the requester. The labels on the bar chart contain only the operation name portion of the

combination of service port name, operation name, namespaces, and service type. The hover help for each bar includes the operation name and the number of messages.

You can display historical data in this view, but be aware that historical data is displayed in multiple rows in the bar chart for each 5-minute time interval in the requested time span. For assistance with these Tivoli Enterprise Portal tasks, see your IBM Tivoli Monitoring documentation.

When the monitoring agent is first started, at least one monitoring interval must be completed before any data is displayed in the view. After the first monitoring interval is complete, this bar chart displays the number of messages by operation for all web services within the monitoring interval. If the web service has no traffic during the monitoring interval, a zero is displayed. For more details on the displayed data in this view, see “Services Inventory_610 attributes” on page 271.

The displayed data in this view is also displayed in the Services Management Agent Environment workspace. For more information, see “Services Management Agent Environment workspace” on page 40).

Average Message Size by Service:Operation:Type view

The Average Message Size by Service:Operation:Type view displays a bar chart of the average size, in bytes, of messages received for each unique combination of service port, operation, and service type (*Provider* or *Requester*).

If the requester and provider are both in the same application server, this bar chart displays each operation twice: once for the provider, and once for the requester. The labels on the bar chart contain only the operation name portion of the combination of service port name, operation name, namespaces, and service type. The hover help for each bar includes the operation name and the average message size, in bytes.

You can display historical data in this view, but be aware that historical data is displayed in multiple rows in the bar chart for each 5-minute time interval in the requested time span.

When the monitoring agent is first started, at least one 5-minute interval must be completed before any data is displayed in the view. After the first 5-minute interval is complete, the bar chart displays the average message size for all web services within the 5-minute time interval. If the web service has no traffic during the time interval, then the average message size is indicated with a value of -1 and is not included in the bar chart. For more details about the data that is displayed in this view, see “Services Inventory_610 attributes” on page 271.

The data in this chart is based on the time at the monitoring agent, which might not be the same as the time at the hub of the Tivoli Enterprise Monitoring Server. You must keep your computer system clocks synchronized.

The data that is displayed in this view is also displayed in the Services Management Agent Environment workspace. For more information, see “Services Management Agent Environment workspace” on page 40).

Faults Summary workspace

The Faults Summary workspace provides a general summary of faults by operation.

To access the workspace, in the Navigator Physical view, select a Faults Summary node for a data collector.

Number of Faults by Operation view

The Number of Faults by Operation view displays a bar chart of the number of faults reported for each unique combination of service port, operation, and service type (*Provider* or *Requester*). This chart is an aggregate view of the most recent sampling interval. The view is generated from the Services Inventory table, which contains calculations of the number of faults within the monitoring interval.

If the requester and provider are in the same application server, this bar chart displays each operation twice: once for the provider, and once for the requester. You can display historical data in this view, but be aware that historical data is displayed in multiple rows in the bar chart for each monitoring interval in the requested time span. The hover help information for each bar includes the operation namespace and the number of faults.

When the monitoring agent is first started, at least one monitoring interval must be completed before any data is displayed in the view. After the first monitoring interval is complete, the bar chart displays the number of faults for all web services within the monitoring interval. If the web service has no traffic during the monitoring interval, then a zero is displayed. For more details about the data that is displayed in this view, see “Services Inventory_610 attributes” on page 271.

Because this bar chart is based on the Services Inventory table, and the Fault Details view is generated from data in the Fault Details table, the number of faults might not match the number of faults that are displayed.

Fault Details view

The Fault Details table view displays the details of the faults that are received for each operation, including the date and time when the message was observed, the fault code that is specified in the response message, and message text that is associated with the fault code. This table is a real-time view of the observed data. You might not see the data reflected in the bar chart until the current monitoring interval completes and the bar chart is updated. The fault data that is displayed in this table is for faults that occurred during the most recently completed monitoring interval.

Beginning with ITCAM for SOA 7.2 Fix Pack 1, the Fault Details table view displays additional details about faults that are received for transactions on DataPower SOA appliances. The following columns are displayed in the Fault Details table view:

Application Server Node Name (Unicode)

The node name of the application server where the message was intercepted.

Application Server Cell Name (Unicode)

The name of the application server cell where the message was intercepted.

Application Server Cluster Name (Unicode)

The name of the application server cluster where the message was intercepted.

Error code

The error code of the DataPower transaction.

Tip: If the fault is not generated by a DataPower Appliance, the error code field might be empty.

Error subcode

Additional details about the root cause of the error.

Tip: If the fault is not generated by a DataPower Appliance, the error subcode field might be empty.

Domain

The DataPower domain name.

Transaction Identity

The ID of the transaction. The transaction identity is set on the DataPower appliance.

Requester Identity

The host IP Address or the user ID of the requester. The type of requester information that is displayed depends on the value of the `kd4.ira.fault.reqid.type` property in `KD4.dc.properties` file.

These columns are hidden by default. To display the columns, complete the following steps:

1. Right-click the Fault Details table view and select **Properties**.
2. On the Properties Editor, click the **Filters** tab.
3. Select the check box beneath the following column headings:
 - Application Server Node Name (Unicode)
 - Application Server Cell Name (Unicode))
 - Application Server Cluster Name (Unicode)
 - Error code
 - Error Subcode
 - Domain
 - Transaction Identity
 - Requester Identity
4. Click **OK** to apply your changes to the view and to close the Properties Editor.

For more information about customizing table views, see the *Tivoli Enterprise Portal User's Guide* in the Tivoli Monitoring information center.

The data for this view comes from the Fault Log_610 attribute table (for more information, see "Fault Log_610 attributes" on page 266).

DataPower Console workspace

The DataPower Console workspace provides management of the DataPower appliance through the DataPower WebGUI, launched from an integrated browser view in Tivoli Enterprise Portal.

To access the workspace, you can take one of the following actions:

- In the Performance Summary workspace, in the Services Inventory table, select a row that corresponds to a DataPower mediation operation instance, right click, and select **Link To > DataPower Console**.
- In any Operational Flow workspace, in the Interaction Details view, select a DataPower mediation operation instance, right click, and select **Link To > DataPower Console**

When you first access the DataPower Console workspace, you must enter your login information.

In the Tivoli Monitoring link definition, this workspace requires the link symbols *DPHostname* and *DPPort*. They define the host name and the port for the WebGUI interface.

If you customize the DataPower appliance so that the DataPower WebGUI interface is accessed through another port, you can modify the link definition to provide an alternate port number. The default setting for the port number is *9090*. Use the *DPPort* link symbol to specify the port number.

Changing the port number: When you change the port number in the link definition, it affects all links to the DataPower Console workspace. If multiple DataPower appliances use different port numbers, you must create separate link definitions. The port number is **not** passed along as part of the context for the link.

For information about creating link definitions, see the IBM Tivoli Monitoring documentation. For more information about the DataPower interface, see the *DataPower WebGUI Guide*, and refer to the online help for procedures about accessing the DataPower Console workspace.

Chapter 5. Workspaces for monitoring by requester identity

Typical ITCAM for SOA views show metric data that is aggregated by each service port and operation name pair, without regard for the specific source (the user or business partner) from which the request for the service and operation originated. These origin points, also called *web services requesters*, can be tracked and monitored for the quality of the service being requested. This additional level of end-to-end monitoring of web services helps you to verify that service expectations are being provided to users and business partners.

ITCAM for SOA supports the capability to configure monitoring for one or more *requester identities*, and to display metric data for each monitored combination of service port, operation, and requester identity. This data is displayed in a set of predefined Tivoli Enterprise Portal workspaces and views. These views show metric data and relationships broken down by requester identity within a particular service port and operation pair.

You can identify the origin of a web services request in several ways. This version of ITCAM for SOA supports two methods:

- By *security principal*, or the user ID with which the requester authenticated, for example, *noreply@us.ibm.com*.
- By the host name or IP address of the remote computer from which the request originated.

To monitor message traffic by requester identity, you use several Take Action commands to complete the following configuration tasks:

1. Configure the type of requester identity that you want to monitor. You can select only one of the two options (security principal or host name/IP address), but not both at the same time.
2. Create the list of requester identities that you want to monitor. The monitoring agent must store an entire set of metric data for each monitored requester identity. Therefore, restrict the list to the smallest number of requester identities that must be monitored.
3. Enable data collection by requester identity.

Environments that support monitoring by requester identity

Table 5 on page 48 displays the application server runtime environments and enterprise service bus environments that support monitoring by requester identity using either or both of these types of requester identities:

- User ID
- Hostname or IP address

Refer to your application server documentation for more details on its security support if you decide to monitor requester identity by user ID.

Table 5. Runtime environments that support monitoring by requester identity

Application server runtime environment or enterprise service bus environment	Requester identity type supported:	Comment
IBM WebSphere Application Server	User ID	The user ID is the Java Authentication and Authorization Service (JAAS) principal.
IBM WebSphere Business Integration	Hostname / IP address	These runtime environments support monitoring of the remote host name or IP address except when monitoring SCA components and operations.
IBM WebSphere Process Server		
IBM WebSphere Enterprise Service Bus		
IBM Business Process Manager Server		
IBM WebSphere Message Broker	User ID	For JMS message flows, the user ID is obtained from the JMSXUserID header.
	Hostname / IP address	<p>For WebSphere MQ message flows, the user ID is obtained from the UserIdentifier field in the MQMD message descriptor.</p> <p>For HTTP message flows in WebSphere Message Broker version 6.1.0.2 or later, the host name or IP address is obtained from the HTTPInput header. For WebSphere Message Broker version 6.0.0.5 or later fix pack, monitoring by requester identity is not supported for HTTP message flows.</p> <p>For a message flow starting with a SOAPInput node in WebSphere Message Broker version 6.1.0.2 or later:</p> <ul style="list-style-type: none"> • The host name or IP address is obtained from the HTTPInput header. • If HTTP basic authentication is enabled for the flow, the user ID is obtained from the HTTP headers. • If user name token authentication is enabled for the flow, the user ID is obtained from the user name field of the WS-Security header. • If X.509 token authentication is enabled for the flow, the X.509 token in the WS-Security header is used as the user ID.

Table 5. Runtime environments that support monitoring by requester identity (continued)

Application server runtime environment or enterprise service bus environment	Requester identity type supported:	Comment
IBM DataPower SOA Appliance	User ID	The user ID for this environment is obtained from the result of a AAA policy assertion on the DataPower SOA appliance. For information about setting up your AAA policy, see the DataPower WebGUI guide, and the <i>IBM Tivoli Composite Application Manager for SOA Installation Guide</i> for details on the DataPower AAA policy extraction methods supported by the ITCAM for SOA data collector, and configuring the DataPower SOA Appliance for monitoring requester identities by user ID.
	Hostname / IP address	
IBM CICS Transaction Server	User ID	The user ID for this runtime environment is obtained from the DFHWS-USERID container.
Microsoft .NET	User ID	The user ID for this application server runtime environment is obtained from the CurrentPrincipal.Identity.Name property of the thread on which the web service is running.
	Hostname / IP address	
Apache Axis SOAP Engine with BEA WebLogic	User ID	The user ID for this application server runtime environment is obtained from WS-Security for Java (WSS4J) attributes.
	Hostname / IP address	
JBoss	User ID	The user ID for this application server runtime environment is obtained from WS-Security for Java (WSS4J) attributes.
	Hostname / IP address	

Requester Identity Monitoring Configuration workspace

To create and maintain the list of all the requester identities to be monitored, use the Requester Identity Monitoring Configuration workspace. To access this workspace from the Physical Navigator view, right-click the **Services Management Agent** workspace node and select **Workspace -> Requester Identity Monitoring Configuration**.

Requester Identity Monitoring Status view

The Requester Identity Monitoring Status table view displays a property setting of *On* or *Off* to indicate whether data collection is enabled or disabled for all requester identities that are displayed in the Monitored Requester Identities table view. The default setting is *Off*.

To turn on or turn off your requester identity monitoring configuration property settings, use these Take Action commands:

- “EnableReqIDMntr_610 Take Action command” on page 247 turns on requester identity monitoring.
- “DisableReqIDMntr_610 Take Action command” on page 244 turns off requester identity monitoring.

This view also displays the type of requester identity that is being monitored. Valid values for the type of requester identity include either *User_ID* or *Remote_Hostname*. This value indicates whether the requester identity values that are being captured are user information (user ID) or the remote computer (host name or IP address) making the service request for the collection and aggregation of metrics.

To define the type of requester identity that is being monitored, use these Take Action commands:

- “SetReqIDTypeHostIP Take Action command” on page 247 sets the requester identity type to *Remote_Hostname*.
- “SetReqIDTypeUserInfo Take Action command” on page 248 sets the requester identity type to *User_ID*.

Monitored Requester Identities view

The Monitored Requester Identities single column table view displays the list of requester identities that are being monitored for data collection and aggregation when the property setting is turned on or enabled. If a row in the **Requester Identity (Unicode)** column contains the asterisk (*) wildcard character, all requester identities are monitored for each unique combination of service port and operation name. Initially this table is empty, signifying that no requester identities are configured to be monitored.

Using the asterisk (*) wildcard: With many unique requester identities, usage of this wildcard character can result in a large amount of collected data.

To manage the list of monitored identities, or to remove a requester identity from the list, use these Take Action commands:

- “AddRequesterIdentity_610 Take Action command” on page 233 adds a requester identity to the list.
- “DeleteRequesterIdentity_610 Take Action command” on page 235 removes a requester identity from the list.

Requester Identities for Operation workspace

The Requester Identities for Operation workspace displays the aggregated metric data for a single service. The data is displayed in multiple table rows, one row for each unique combination of service port, operation, and requester identity. This table includes an additional **Requester Identity (Unicode)** column that displays the unique requester identities that are associated with the selected row from the Services Inventory table view.

To access the workspace, right-click a row (with a Service Type value of *Provider*) in the Services Inventory table view. Then, select **Link To -> Requester Identities for Operation**. Alternatively, you can click the row link indicator for the selected row and select **Requester Identities for Operation**.

Using this workspace, you can examine the metrics and relationships by requester identity for a specific service port and operation pair. By default, only requester identities that initiated requests within the last 70 minutes are displayed. To display a list of all requester identities that initiated requests before the last 70 minutes, you must complete a historical data query.

From this workspace, you can also select a specific requester identity and link to the Performance Summary for Requester Identity workspace. That workspace displays table and bar chart views of message size, round-trip response time, message counts, and fault counts for the selected requester identity. To link to the workspace, right-click a row in the Requester Identities for Operation table view and select **Link To -> Performance Summary for Requester Identity**. You can also select the workspace link icon and then the **Performance Summary for Requester Identity** link. For more information, see “Performance Summary for Requester Identity workspace.”

Important: Information is only displayed for requester identities that are monitored. If no requests to this service have such requester identities, the table is empty. To configure the list of monitored requester identities, see “Requester Identity Monitoring Configuration workspace” on page 49.

Performance Summary for Requester Identity workspace

The Performance Summary for Requester Identity workspace provides a table view and bar charts that display the services activity for a specific requester identity that is associated with a particular combination of service port and operation.

To access the workspace, in the Requester Identities for Operation workspace, select a row and link Performance Summary for Requester Identity. For more information, see “Requester Identities for Operation workspace” on page 50.

The requester identity for which data is being displayed is included in the view titles for each view in this workspace.

As with any IBM Tivoli Monitoring workspace, after you configure a requester identity to be monitored, you can create your own workspace links to this workspace.

By default, requester identity metrics are available for a service type of provider only. Beginning with ITCAM for SOA 7.2 Fix Pack 1, you can override this default behavior by setting the `kd4.ira.reqid.requester.enabled` property in the `KD4.dc.properties` file to 1. When the property is set to 1, requester identity metrics are available for both service types; provider and requester.

Remember: Metrics for service requester are only generated when requester identity monitoring is based on the `User_ID`. You can set requester identity monitoring using the “SetReqIDTypeUserInfo Take Action command” on page 248.

For more information about setting the `kd4.ira.reqid.requester.enabled` property, see the “Configuring the aggregation of metrics in the Service Inventory Requester Identity table” section of the *IBM Tivoli Composite Application Manager for SOA Installation Guide*.

Message and Fault Count view

The Message and Fault Count table view displays the number of messages and faults reported for the specified combination of service port and operation for a service type of *Provider*. This table view is an aggregate view of the most recent sampling interval.

When the monitoring agent is first started, at least one monitoring interval must be completed before any data is displayed in the view. After the first monitoring

interval is complete, this table view displays the number of messages and faults by service port and operation combination for all services within the monitoring interval. If the web service has no traffic during the time interval, a zero is displayed.

Response Time by Operation view

The Response Time by Operation bar chart view displays in milliseconds the average, minimum, and maximum response time for the specified combination of requester identity, service port, and operation, with the message being intercepted as it leaves the server.

This bar chart displays each operation only for a service type of *Provider*. The data displayed is for the most recently completed monitoring interval.

When the monitoring agent is first started, at least one monitoring interval must be completed before any data is displayed in the view. After the first monitoring interval is complete, this bar chart displays the average response time for all services that had a valid average response time within the monitoring interval. If the service has no traffic during the monitoring interval, then the response time is indicated with a value of *-1* and is not included in the bar chart.

The labels on the bar chart contain only the operation name portion of the combination of service port name, operation name, and namespaces.

Message Size by Operation view

The Message Size by Operation bar chart view displays the average, minimum and maximum message size, in bytes, of messages that are received for the specified combination of requester identity, service port, and operation. This bar chart displays each operation only for a service type of *Provider*.

When the monitoring agent is first started, at least one monitoring interval must be completed before any data is displayed in the view. After the first monitoring interval is complete, this bar chart displays the average message size for all services within the monitoring interval. If the service has no traffic during the monitoring interval, then the message size is indicated with a value of *-1* and is not included in the bar chart.

The labels on the bar chart contain only the operation name portion of the combination of service port name, operation name, and namespaces.

Chapter 6. Workspace for service registry integration

You can use ITCAM for SOA to displays information about relationships between resources on different hosts.

The Services Management workspace shows the data. You can use it to display these types of information:

- Services and service relationships that are registered with the WebSphere Service Registry and Repository and that are imported into Tivoli Common Object Repository.
- Service metadata that WebSphere Service Registry and Repository uses to define services, service ports, and operations.
- Services that are discovered by ITCAM for SOA agents and imported into Tivoli Common Object Repository.
- Comparisons of registered service definitions to discovered services.
- Cross computer relationships among services and business processes that are modeled in Business Process Execution Language and imported into Tivoli Common Object Repository.

The information about services and service relationships displayed in the Services Management workspace depends on data that is retrieved from the Tivoli Common Object Repository component of ITCAM for SOA. This data is populated in Tivoli Common Object Repository using several Discovery Library Adapters that are provided with the product:

WebSphere Service Registry and Repository Discovery Library Adapter

Discovers the relationships between service ports, operations, services, port types, and metadata documents from WebSphere Service Registry and Repository. WebSphere Service Registry and Repository can process Web Services Description Language (WSDL) and other definition files to determine these static relationships.

Business Process Execution Language for Web Services Discovery Library Adapter

Discovers the relationships between port types, operations, and business processes from Business Process Execution Language files.

ITCAM for SOA Discovery Library Adapter

Discovers the relationships between service ports and operations and the application servers and computer systems on which they are deployed, using data collected by ITCAM for SOA monitoring agents and retrieved from Tivoli Enterprise Monitoring Server.

Before you can display service registry integration data in the Services Management workspace, you must install, configure, and run one or more of these Discovery Library Adapters to populate Tivoli Common Object Repository with service relationship data.

You start with the Services Overview table, listing all services in your environment. You can then navigate to detailed views, which can be displayed as graphical topology or tables, depending on the number of services in the view.

As information for this workspace is obtained from multiple data sources, complete information for any particular service might not be available.

Limitations of service registry integration monitoring

Certain limitations apply to display of service to service relationships based on service registry integration.

Unmatched service port names and operation names: The ITCAM for SOA Discovery Library Adapter discovers the service ports and operations that are observed by the ITCAM for SOA data collectors. In some cases, the service port names, namespaces, and operation names discovered by the data collectors do not match the service ports and operations discovered by the WebSphere Service Registry and Repository Discovery Library Adapter or by the Business Process Execution Language Discovery Library Adapter. For example:

- The ITCAM for SOA CICS data collector is unable to discover the service port namespace or operation namespace from a Web Services Description Language document for CICS web services. Therefore, the CICS service ports and operations discovered by the ITCAM for SOA Discovery Library Adapter do not match the CICS service ports and operations discovered by the WebSphere Service Registry and Repository Discovery Library Adapter and the Business Process Execution Language for Web Services Discovery Library Adapter.
- The Microsoft .NET data collector is not able to discover the service port name in a Web Services Description Language document. Instead, it discovers the service name. Therefore, Microsoft .NET service ports and operations discovered by the ITCAM for SOA Discovery Library Adapter do not match the Microsoft .NET service ports and operations discovered by the WebSphere Service Registry and Repository Discovery Library Adapter.
- When you create Web Services Description Language documents for your WebSphere Message Broker SOAP services, define the service port names and namespaces and operation names and namespaces to follow the conventions that are used by the ITCAM for SOA WebSphere Message Broker data collector. Otherwise, the WebSphere Message Broker SOAP service ports and operations discovered by the ITCAM for SOA Discovery Library Adapter might not match the WebSphere Message Broker services that are discovered by the WebSphere Service Registry and Repository Discovery Library Adapter.

Service port on multiple application servers: If a service port is observed on multiple application servers, the Services Overview table shows all of the operations on each of the application servers, even if the monitoring agents did not observe every operation on each application server. The data is presented this way in the table because the table presents the logical topology view of the port.

For example, suppose that service port *Customer* is observed on two application servers, *server1* and *server2*. The *getCustomer* operation for service port *Customer* is invoked on both application servers, but the *lookupCustomer* operation for this service is invoked only on *server1*.

In this case, the Services Overview table displays two rows for the combination of the *Customer* service port and *getCustomer* operation, one row for application server *server1*, and another row for application server *server2*. There are also two rows for the combination of *Customer* service port and *lookupCustomer* operation: one row for application server *server1*, and another row for application server *server2*.

Applications with SCA operations: SCA modules, components, and operations are displayed only in the Services Overview table and static topology views if they are observed by the ITCAM for SOA data collectors and you import ITCAM for SOA DLA books into Tivoli Common Object Repository. The SCA component and operation rows in the Services Overview table never has a check mark in the *Registered* column because SCA modules in WebSphere Service Registry and Repository are not discovered by the WebSphere Service Registry and Repository DLA.

Non-SOAP Messages in WebSphere Message Broker: The Services Overview table view and the other static topology views do not display information about non-SOAP message flows observed by the ITCAM for SOA data collector in a WebSphere Message Broker application server environment.

Synchronize the Services Overview table and Operational Flow workspaces: The topology data in the Services Management workspace is retrieved from Tivoli Common Object Repository, which is populated with data in Discovery Library Adapter books. To keep the observed services in the Services Overview table updated with the topology information displayed in the more dynamic Operational Flow workspaces, run the ITCAM for SOA Discovery Library Adapter on a regular basis and bulk load its book into Tivoli Common Object Repository.

DataPower hostname for data collector, not appliance, is displayed: When a service port and operation are observed on a DataPower appliance by the ITCAM for SOA data collector, the Services Overview table and other static topology views show the host name and IP address of the computer where the data collector is running instead of the host name and IP address of the DataPower appliance. This is consistent with the host name and IP address shown in the Services Inventory view of the Performance Summary workspace for the data collector node associated with the monitored DataPower appliance.

Services Management workspace

The Services Management workspace contains information on service to service relationships, gathered based on ITCAM for SOA monitoring as well as WebSphere Service Registry and Repository data. This workspace provides views to help you with the following tasks:

- View the services and service relationships that are registered with WebSphere Service Registry and Repository (WSRR) that are discovered by the WSRR DLA and imported into the Tivoli Common Object Repository.
- View service metadata documents from the WebSphere Service Registry and Repository.
- View the service ports and operations that are discovered by the ITCAM for SOA DLA and imported into the Tivoli Common Object Repository.
- Compare a set of registered service definitions with a set of discovered services, to verify that services are implemented and operating as designed.
- View the business processes that use activities that are implemented by web services that are discovered by the ITCAM for SOA DLA or the WSRR DLA. The business process and service relationship data is discovered by the BPDL DLA and imported in to the Tivoli Common Object Repository.

To access the Services Management workspace, in the “The Navigator ITCAM for SOA view” on page 26, right-click the Services Management node and select **Workspaces > Services Management**. For more information, see “The Navigator ITCAM for SOA view” on page 26.

When you open the workspace, the Services Overview table is displayed (see “Services Overview table view”).

The table views in this workspace contain data strings in the UTF-8 character set, supporting data that is represented in different languages. Unlike other workspaces, the **(Unicode)** designation does not appear in the column titles for the table. The views that are provided with this workspace are described in the sections that follow.

Services Overview table view

The Services Overview table view displays services that are defined in WebSphere Service Registry and Repository or discovered by ITCAM for SOA. This view is the initial view for the Services Management workspace.

Table 6 lists the columns for the table view.

Table 6. The names and descriptions of the column headings for the Services Overview table view.

Name	Description
Service Port	For web services, the name of the Web Services Description Language service port, otherwise the service port name assigned to the service.
Operation	For web services, the name of the Web Services Description Language operation, otherwise the operation name assigned to the service.
Service	The name of the registered Web Services Description Language service.
Application Server	The name of the application server on which the service port is deployed.
Computer System	The name of the computer on which the application server is running.
Observed	Indicates that the service port and operation were observed by the ITCAM for SOA monitoring agent on the provider (or server) side of a request.
Registered	Indicates that the service is defined in the WebSphere Service Registry and Repository.

When you right-click a row in the table, the resulting menu provides a list of resource actions and views. The following resource relationship views are available from the Services Overview table view:

- **View Service Details** Displays the Service Details view for the service in the selected row. If no service information is available, this menu item is not available. See “Service Details view” on page 58.
- **View Service Port Details** Displays the Service Port Details view for the service port in the selected row. See “Service Port Details view” on page 61.
- **View Business Processes for Service** Displays the Business Process view for the service. If no service information is available in the selected row, this menu item is not available. If there is no business process information available for this service, an empty view is displayed. See “Business Processes for Service view” on page 63.
- **View Business Processes for Service Port** Displays the Business Process view for the service port in the selected row. If there is no business process

information available for this service port, an empty view is displayed. See “Business Processes for Service Port view” on page 65

- **View Metadata** Displays the Metadata dialog, which might contain a list of metadata documents for the service, service port, and operation for the selected row of the Services Overview table view. If only one metadata document is available, it is displayed. See “Viewing metadata” on page 68.

You can also navigate to the Performance Summary and Operational Flow for Application Server workspaces for the application server. To navigate to these workspaces, right click a row in the table for which a server name is displayed in the **Application Server** column, and select the **Link To** action.

Static topology views

Detail views in the Services Management workspace can display a graphical topology. It is known as *static topology*, because it includes the fixed relationships between services, as registered in WebSphere Service Registry and Repository.

The Tivoli Enterprise Portal provides basic topology functions. For example, zoom state, panning placement, and the ability to display the topology in a table form. You can also use Next and Back navigation arrows to move through a series of topology views and tables. For more information about workspace functions, see the Tivoli Enterprise Portal online help.






The network resources in the ITCAM for SOA view are displayed in a logical topology, and represented by icons. The relationships between resources are represented by solid line arrows.

Resource type icons

In static topology views, different icons represent different resource types

Table 7 describes the icons.

Table 7. The icons and descriptions that are displayed in the static topology.

Icon	Description
	A service port that is observed by the ITCAM for SOA agent data collector or defined in the WebSphere Services Registry and Repository.
	An operation observed by the ITCAM for SOA agent data collector or defined in the WebSphere Services Registry and Repository.
	A Web service defined in the WebSphere Services Registry and Repository.
	An application server monitored by an ITCAM for SOA data collector.
	A business process that is defined in Business Process Execution Language.

Relationships between resource types

Relationships between resource types are displayed in the static topology with a solid line and an arrow.

Table 8 describes the defined resource relationships.

Table 8. The style and the defined resource relationships in the static topology.

Relationship Type	Line Type	Resource Relationships	Description
Contains	Solid line with arrow	<ul style="list-style-type: none">• Service → Service Port• Business Process → Operation	The containment relationship.
Hosts	Solid line with arrow	Application Server → Service Port	Indicates that the application server is hosting the runtime environment that the service port is deployed on.
Defined within	Solid line with arrow	Operation → Service Port	Indicates that the definition for the operation is defined in the Web Services Description Language for the service port.

Service Details view

The Service Details view is a static topology and table view that displays the service ports, operations, web services, and application servers that are related to a specified service.

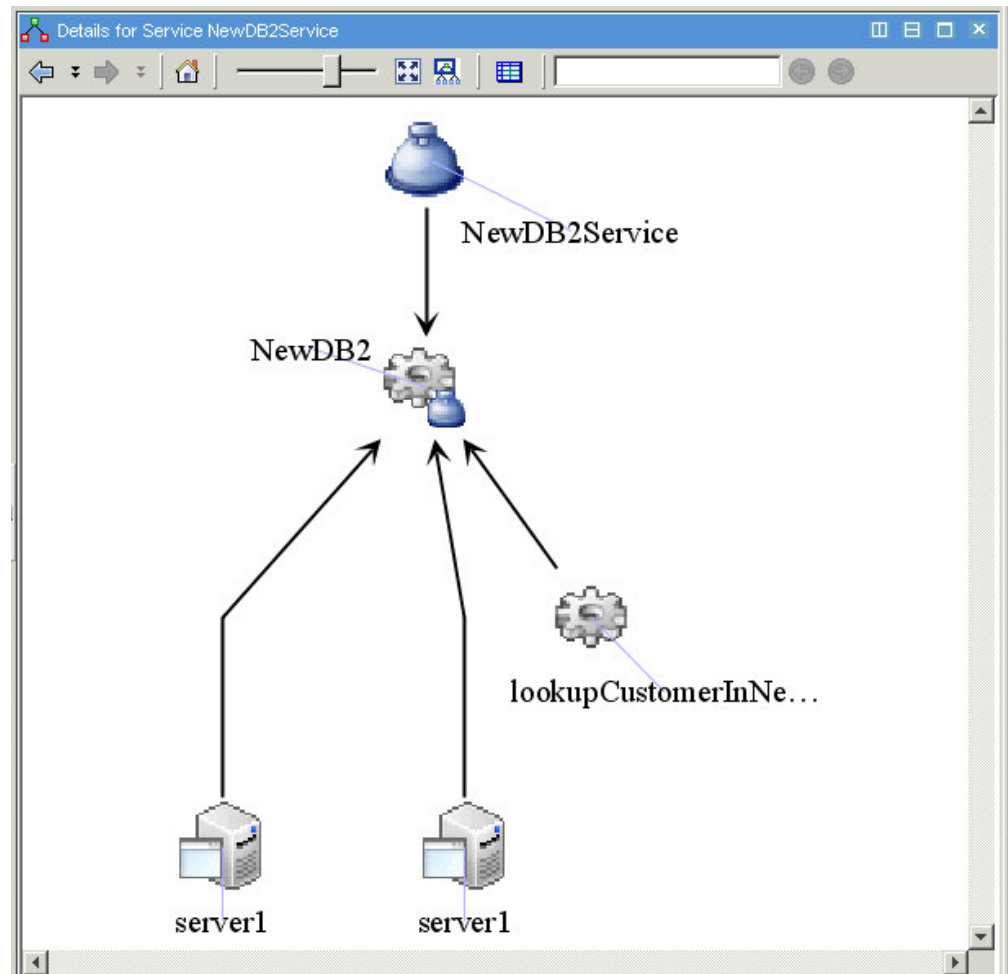


Figure 4. The Service Details view in the Services Management workspace

To receive additional information for a resource, place the mouse cursor over the resource icon to display a flyover window.

When you right-click a resource in the topology or a row in the table, the menu provides a list of resource actions and views. The following resource relationship views are available from the Service Details view:

- **View Service Port Details** Displays the Service Port Details view for a selected service port. See “Service Port Details view” on page 61.
- **View Business Processes for Service** Displays the Business Process view for the selected service. See “Business Processes for Service view” on page 63.
- **View Business Processes for Service Port** Displays the Business Process view for the selected service port. See “Business Processes for Service Port view” on page 65.
- **View Metadata** Displays the Metadata window, which might contain a list of metadata documents for the service, service port or operation. If only one metadata document is available, it is displayed. See “Viewing metadata” on page 68.

You can right-click an application server icon and selecting the **Link To** action to navigate directly to the Performance Summary workspace or to the Operational

Flow for Application Server workspace for that application server. If multiple application servers are associated with the selected service port, you are prompted to select an application server.

If a large amount of resources is included in the view (by default, 50 or more), a table view is displayed instead of the topology. You can narrow your scope by using the column sorting and filtering capabilities. To change the number for which a table view is displayed, or to display the table or topology view at all times, see “Setting the threshold for static topology” on page 69.

For more information about the other menu actions and the capabilities for column sorting and filtering, see the online help for Tivoli Enterprise Portal.

The Service Details table view contains these column headings:

Table 9. The names and descriptions of the column headings for the Service Details table view.

Name	Description
Name	The name of the resource.
Resource Type	The type of resource and its accompanying icon.
Namespace	The namespace used to fully qualify the resource.
Port Type	The definition of a set of operations for the service port that can be deployed as a group.
Computer System	The name of the computer on which the application server is running.
Source	The origin of the relationship indicated by the solid line arrows in the topology view, that indicates a directional relationship between the source and the destination.
Destination	The end point of the relationship indicated by the solid line arrows in the topology view, that indicates a directional relationship between the source and the destination.

Use the **Show/Hide additional attributes** button on the table toolbar to turn on the columns to show more information about the services. In addition to the previously mentioned column headings, the Service Details table view has the following attributes:

Table 10. Additional attributes for the Service Details table view.

Name	Description
Description	The description of the resource.
Node Name	The name of the node to which the application server belongs.
Cell Name	The name of the cell to which the application server belongs.
Cluster Name	The name of the cluster to which the application server belongs.
Vendor Name	The name of the vendor that makes the application server.
Product	The application server licensed program name.
Port Type Namespace	The namespace that fully qualifies the port type.
Computer System IP Address	The IP address for the computer system on which the application server is running.

Service Port Details view

The Service Port Details view is a static topology and table view that displays service ports, operations, web services, and the application servers defined in the Tivoli Common Object Repository as being related to a specified service port.

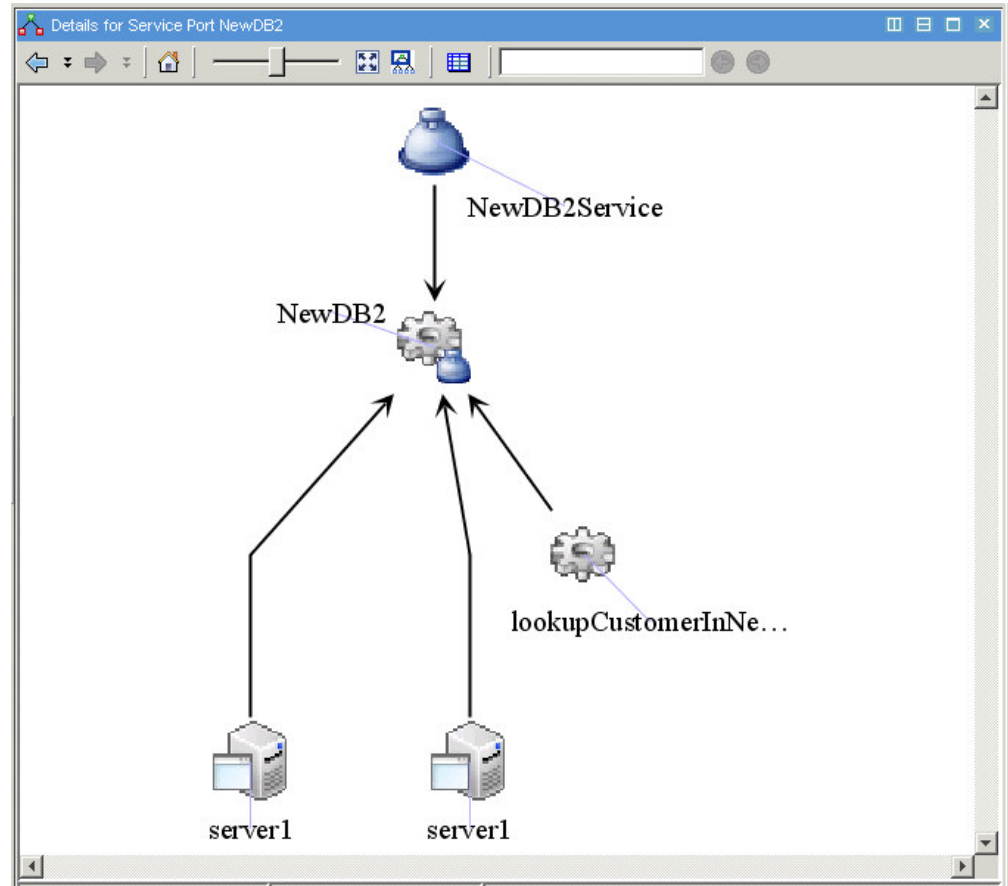


Figure 5. The Service Port Details view in the Services Management workspace

To receive additional information for a resource, place the mouse cursor over the resource icon to display a flyover window.

When you right-click a resource in the topology or a row in the table, the menu provides a list of resource actions and views. The following resource relationship views are available from the Service Port Details view:

- **View Service Details** Displays the Service Port Details view for a selected service port. See “Service Details view” on page 58.
- **View Business Processes for Service** Displays the Business Process view for the selected service. See “Business Processes for Service view” on page 63.
- **View Business Processes for Service Port** Displays the Business Process view for the selected service port. See “Business Processes for Service Port view” on page 65.
- **View Metadata** Displays the Metadata window, which might contain a list of metadata documents for the service, service port or operation. If only one metadata document is available, it is displayed. See “Viewing metadata” on page 68.

You can right-click an application server icon and selecting the **Link To** action to navigate directly to the Performance Summary workspace or to the Operational Flow for Application Server workspace for that application server. If multiple application servers are associated with the selected service port, you are prompted to select an application server.

If a large amount of resources is included in the view (by default, 50 or more), a table view is displayed instead of the topology. You can narrow your scope by using the column sorting and filtering capabilities. To change the number for which a table view is displayed, or to display the table or topology view at all times, see “Setting the threshold for static topology” on page 69.

For more information about the other menu actions and the capabilities for column sorting and filtering, see the online help for Tivoli Enterprise Portal.

The Service Port Details table view contains these column headings:

Table 11. The names and descriptions of the column headings for the Service Port Details table view.

Name	Description
Name	The name of the resource.
Resource Type	The type of resource and its accompanying icon.
Namespace	The namespace used to fully qualify the resource.
Port Type	The definition of a set of operations for the service port that can be deployed as a group.
Computer System	The name of the computer on which the application server is running.
Source	The origin of the relationship indicated by the solid line arrows in the topology view, that indicates a directional relationship between the source and the destination.
Destination	The end point of the relationship indicated by the solid line arrows in the topology view, that indicates a directional relationship between the source and the destination.

Use the **Show/Hide additional attributes** button on the table toolbar to turn on the columns to show more information about the services. In addition to the previously mentioned column headings, the Service Port Details table view has the following attributes:

Table 12. Additional attributes for the Service Port Details table view.

Name	Description
Description	The description of the resource.
Node Name	The name of the node to which the application server belongs.
Cell Name	The name of the cell to which the application server belongs.

Table 12. Additional attributes for the Service Port Details table view. (continued)

Name	Description
Cluster Name	The name of the cluster to which the application server belongs.
Vendor Name	The name of the vendor that makes the application server.
Product	The application server licensed program name.
Port Type Namespace	The namespace that fully qualifies the port type.
Computer System IP Address	The IP address for the computer system that the application server runs on.

Business Processes for Service view

The Business Processes for Service view shows service relationships for a business process. This topology or table view displays all of the service ports and operations for a specific service that participate in business processes, according to Tivoli Common Object Repository.

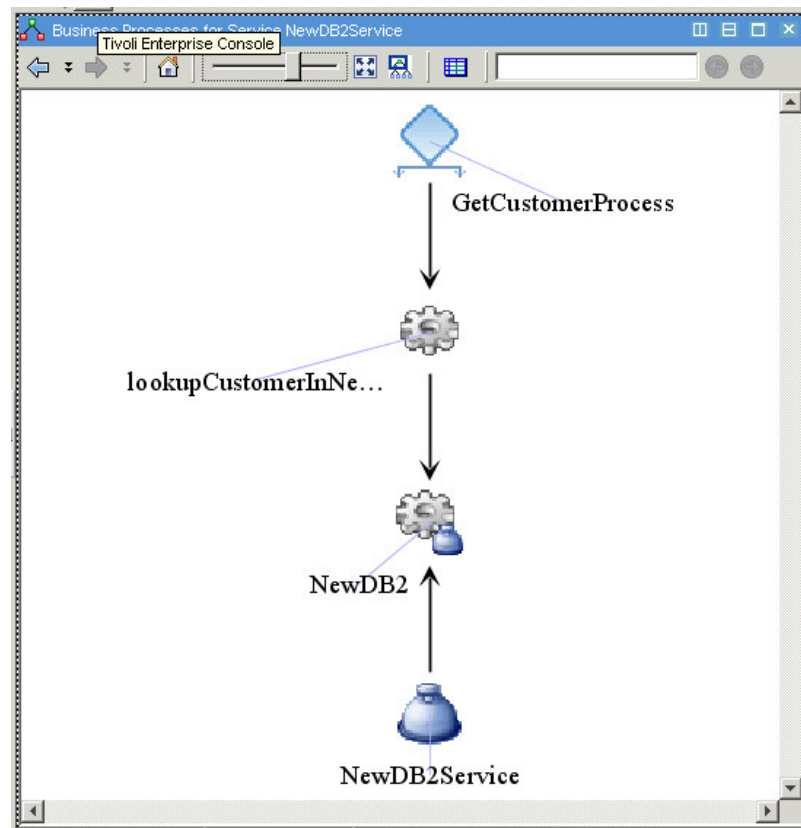


Figure 6. The Business Processes for Service view in the Services Management workspace

To receive additional information for a resource, place the mouse cursor over the resource icon to display a flyover window.

When you right-click a resource in the topology or a row in the table, the menu provides a list of resource actions and views. The following resource relationship views are available from the Business Processes for Service view:

- **View Service Details** Displays the Service Port Details view for a selected service port. See “Service Details view” on page 58.
- **View Service Port Details** Displays the Service Port Details view for a selected service port. See “Service Port Details view” on page 61.
- **View Business Processes for Service Port** Displays the Business Process view for the selected service port. See “Business Processes for Service Port view” on page 65.
- **View Details for Business Process** Displays the Business Process Details view for the selected business process. See “Business Process Details view” on page 66.
- **View Metadata** Displays the Metadata window, which might contain a list of metadata documents for the service, service port or operation. If only one metadata document is available, it is displayed. See “Viewing metadata” on page 68.

Select **Properties** from the menu to set a limit on the number of nodes that you can view in the topology. If the node threshold is exceeded, the topology automatically switches to the table view. The default setting is 50 but you can modify or turn off the threshold. Refer to the online help procedures to edit the properties.

If a large amount of resources is included in the view (by default, 50 or more), a table view is displayed instead of the topology. You can narrow your scope by using the column sorting and filtering capabilities. To change the number for which a table view is displayed, or to display the table or topology view at all times, see “Setting the threshold for static topology” on page 69.

For more information about the other menu actions and the capabilities for column sorting and filtering, see the online help for Tivoli Enterprise Portal.

The Business Processes for Service table view contains these column headings:

Table 13. The names and descriptions of the column headings for the Business Processes for Service view.

Name	Description
Name	The name of the resource.
Resource Type	The type of resource and its accompanying icon.
Namespace	The namespace used to fully qualify the resource.
Port Type	The definition of a set of operations for the service port that can be deployed as a group.
Source	The origin of the relationship indicated by the solid line arrows in the topology view, that indicates a directional relationship between the source and the destination.
Destination	The end point of the relationship indicated by the solid line arrows in the topology view, that indicates a directional relationship between the source and the destination.

Business Processes for Service Port view

The Business Processes for Service Port view displays service port relationships for business processes. This topology or table view displays the service port, its associated service, and all the operations that participate in business processes according to Tivoli Common Object Repository.

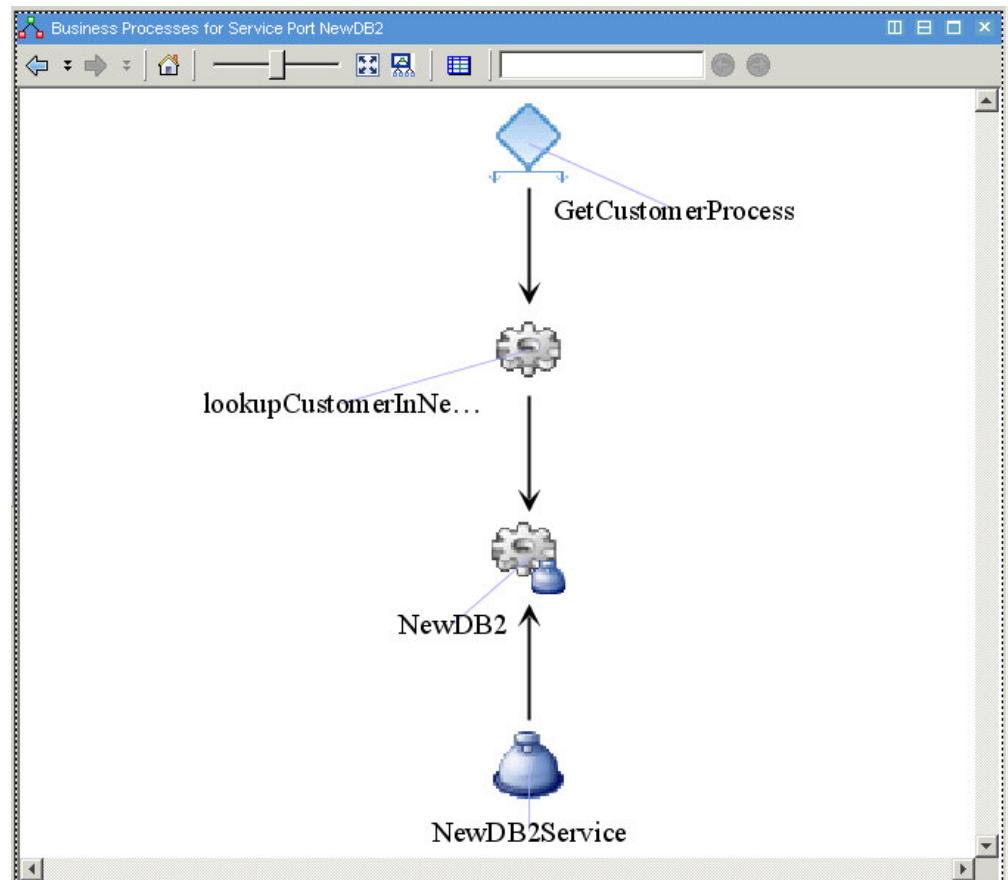


Figure 7. The Business Processes for Service Port view in the Services Management workspace

To receive additional information for a resource, place the mouse cursor over the resource icon to display a flyover window.

When you right-click a resource in the topology or a row in the table, the menu provides a list of resource actions and views. The following resource relationship views are available from the Business Processes for Service Port view:

- **View Service Details** Displays the Service Port Details view for a selected service port. See “Service Details view” on page 58.
- **View Service Port Details** Displays the Service Port Details view for a selected service port. See “Service Port Details view” on page 61.
- **View Business Processes for Service** Displays the Business Process view for the selected service. See “Business Processes for Service view” on page 63.
- **View Details for Business Process** Displays the Business Process Details view for the selected business process. See “Business Process Details view” on page 66.

- **View Metadata** Displays the Metadata window, which might contain a list of metadata documents for the service, service port or operation. If only one metadata document is available, it is displayed. See “Viewing metadata” on page 68.

If a large amount of resources is included in the view (by default, 50 or more), a table view is displayed instead of the topology. You can narrow your scope by using the column sorting and filtering capabilities. To change the number for which a table view is displayed, or to display the table or topology view at all times, see “Setting the threshold for static topology” on page 69.

For more information about the other menu actions and the capabilities for column sorting and filtering, see the online help for Tivoli Enterprise Portal.

The Business Processes for Service Port table view contains these column headings:

Table 14. The names and descriptions of the column headings for the Business Processes for Service Port view.

Name	Description
Name	The name of the resource.
Resource Type	The type of resource and its accompanying icon.
Namespace	The namespace used to fully qualify the resource.
Port Type	The definition of a set of operations for the service port that can be deployed as a group.
Source	The origin of the relationship indicated by the solid line arrows in the topology view, that indicates a directional relationship between the source and the destination.
Destination	The end point of the relationship indicated by the solid line arrows in the topology view, that indicates a directional relationship between the source and the destination.

Business Process Details view

The Business Process Details view is a static topology and table view that displays the operations, service ports, and services that are related to a specific business process according to Tivoli Common Object Repository.

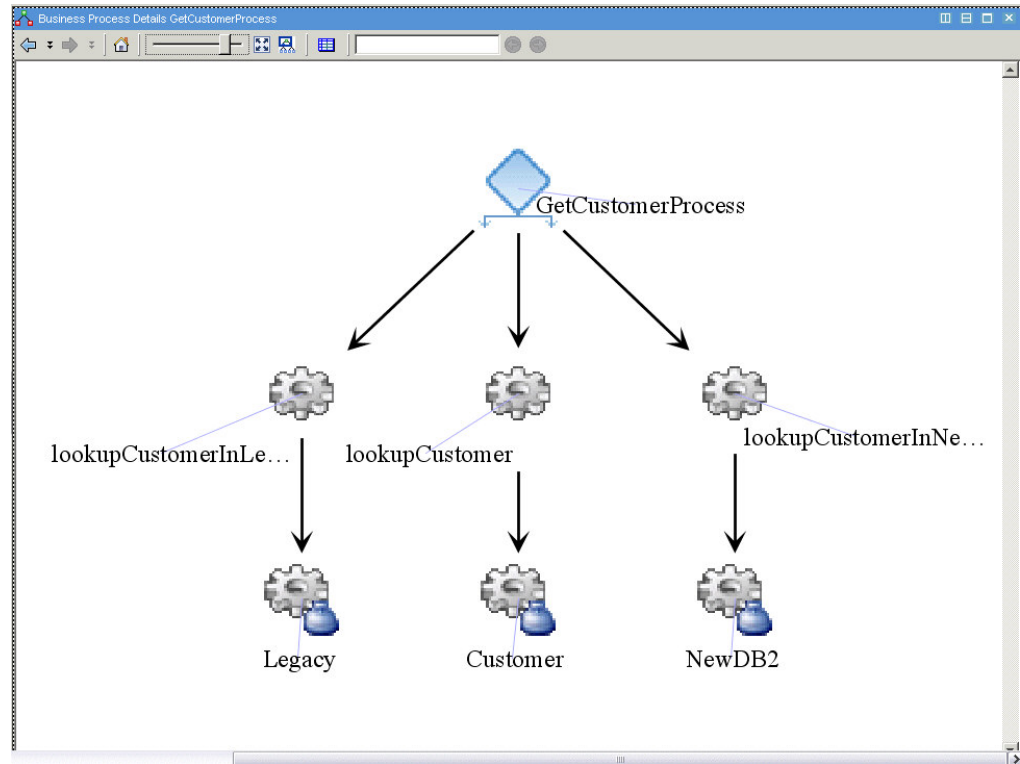


Figure 8. The Business Process Details view in the Services Management workspace

To receive additional information for a resource, place the mouse cursor over the resource icon to display a flyover window.

When you right-click a resource in the topology or a row in the table, the menu provides a list of resource actions and views. The following resource relationship views are available from the Business Process Details view:

- **View Service Details** Displays the Service Port Details view for a selected service port. See “Service Details view” on page 58.
- **View Service Port Details** Displays the Service Port Details view for a selected service port. See “Service Port Details view” on page 61.
- **View Business Processes for Service** Displays the Business Process view for the selected service. See “Business Processes for Service view” on page 63.
- **View Business Processes for Service Port** Displays the Business Process view for the selected service port. See “Business Processes for Service Port view” on page 65.
- **View Metadata** Displays the Metadata window, which might contain a list of metadata documents for the service, service port or operation. If only one metadata document is available, it is displayed. See “Viewing metadata” on page 68.

If a large amount of resources is included in the view (by default, 50 or more), a table view is displayed instead of the topology. You can narrow your scope by using the column sorting and filtering capabilities. To change the number for which a table view is displayed, or to display the table or topology view at all times, see “Setting the threshold for static topology” on page 69.

For more information about the other menu actions and the capabilities for column sorting and filtering, see the online help for Tivoli Enterprise Portal.

The Business Process Details table view contains these column headings:

Table 15. The names and descriptions of the column headings for the Business Process Details table view.

Name	Description
Name	The name of the resource.
Resource Type	The type of resource and its accompanying icon.
Namespace	The namespace used to fully qualify the resource.
Port Type	The definition of a set of operations for the service port that can be deployed as a group.
Source	The origin of the relationship indicated by the solid line arrows in the topology view, that indicates a directional relationship between the source and the destination.
Destination	The end point of the relationship indicated by the solid line arrows in the topology view, that indicates a directional relationship between the source and the destination.

Viewing metadata

Metadata might be available for a service, operation, or port. This information, which might be Web Services Description Language documents, XSD documents, or WS-Policy documents, is displayed in a metadata selection window. The metadata documents are associated with the definition of that resource as defined in WebSphere Service Registry and Repository.

Your selection is displayed in Extensible Markup Language (XML) format. The document view is read-only and there is no linking available within the document to other defined imports. You can open multiple XML documents and they remain open until the window is closed or you click **Close**. If there is only one metadata document associated with the resource, the metadata selection window is bypassed and the document opens automatically.

From the Services Overview table view or the appropriate resource in a view, complete the following steps:

1. Select a row from the table or the appropriate resource and right click to view the menu.
2. From the menu, click **View > Metadata**.
3. If there is one metadata document, it is displayed. If there are several metadata document, a Metadata Selection window is displayed. Select a document from the list and click **View**. The read-only document is displayed.

Important: If a service is not registered in WebSphere Service Registry and Repository, or if the WebSphere Service Registry and Repository DLA is not configured to include Web Services Description Language documents in the data that it discovers, the **View -> Metadata** menu item might not be available.

Setting the threshold for static topology

For each of the detail views in the Services Management workspace, you can set a limit on the number of nodes that you can view in the topology. If more resources are included in a view, it displays a table. The default setting is *50*, but you can modify or turn off the threshold.

To edit the properties, perform the following steps:

1. From any view in the Services Management workspace, select a resource or the view itself, and right-click.
2. From the menu, click **Properties**.
3. From the **Properties** tree, click the view name. The properties view opens.
4. In the **Configuration** tab, select a threshold from the **Configure threshold for automatic table mode switch** list. These are the available thresholds:
 - **Never** (always display a topology view)
 - **50**
 - **100**
 - **200**
 - **Always** (always display a table view)
5. Click **OK**.

Chapter 7. Workspaces for service-to-service topology

ITCAM for SOA provides a set of *service-to-service* topology workspaces and views. Using these views, you can see the interactions among deployed services, and the relationships that services have with each other.

Use the service-to-service topology views to analyze the effect of a problem, understand its scope, and isolate the source of the problem. When you understand the relationships between deployed services, you can see which services are interacting, the metrics that are associated with those relationships, and the status that is associated with any services in the overall flow.

As a services architect, you can also take advantage of the unique perspective of service-to-service topology views to validate the behavior of your SOA design. You can compare your SOA design with its observed behavior to look for discrepancies, places where services are interacting unexpectedly, or where services are not being called as expected.

Service-to-service topology uses workspace linking to integrate different perspectives on the operational SOA. You can link to other relevant views of the SOA behavior, both within ITCAM for SOA and within other monitoring product workspaces and views.

Service-to-service topology concepts

An *operation*, in the context of service-to-service topology, represents a specific function that is provided by a *service*. One or more operations represent the service in the topology.

The specific function that is provided by an operation can be different, depending on how the service is implemented. For example, for a web service, an operation represents a Web Services Description Language defined operation.

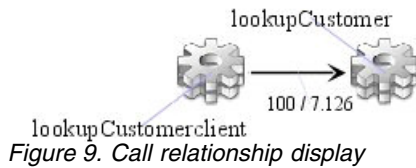
An *operation instance* is the deployment of an operation to an application server, and is uniquely defined by a number of attributes:

- The service port name and namespace
- The operation name and namespace
- The mediation type, (for example, DataPower, WebSphere Message Broker, or Service Component Architecture)
- The application server environment where the service port and operation are deployed, (for example, WebSphere Application Server, JBoss, and others)

An *operation aggregate* represents a set of operation instances implementing the same function. In a distributed environment, an aggregate can consist of a number of instances, and some of them can be in different geographic locations. An operation aggregate is uniquely identified by the service port name and namespace, operation name and namespace, and the mediation type.

As operation instances call, or are called by, other operation instances, the resulting call activity is monitored by the ITCAM for SOA data collector. This call information is then obtained by the SOA Domain Management Server, which in turn derives a set of *call relationships*. A call relationship is represented in topology

views by a line with an arrow.



These call relationships provide the basis for service-to-service topology, and can include the following types of information:

- The relationships between one or more operations (aggregates or instances) and a particular target operation (aggregate or instance) that is called
- The relationships between a particular operation (aggregate or instance) and the set of target operations (aggregates or instances) that are called
- The identity of application servers or DataPower appliances on which calls are detected
- The summarized metrics between an operation (aggregate or instance) and a target operation (aggregate or instance) that is called

An operation (aggregate or instance) can represent one of the following supported types of mediations:

Service Component Architecture (SCA) mediation

An operation that is included in an SCA mediation module, and that is observed by the ITCAM for SOA data collector in the IBM Business Process Manager Server, WebSphere Process Server, or WebSphere Enterprise Service Bus runtime environment.

DataPower mediation

An operation that is detected by the ITCAM for SOA data collector in the DataPower runtime environment.

WebSphere Message Broker mediation

An operation in a SOAP message flow that is detected by the ITCAM for SOA data collector in the WebSphere Message Broker runtime environment.

In the service-to-service topology views, you can see *operational flows*. An operational flow is made up of one or more *call relationship paths*. A call relationship path is a set of caller operations and a set of target operations, and aggregated metrics on those call relationship paths. For example, *Operation 1* calls *Operation 2*, which in turn calls *Operation 3*.

A typical call relationship involves one operation instance (the requester, or client) calling another operation instance (the provider, or server), with both requester and provider sides of the relationship being monitored by an ITCAM for SOA monitoring agent. However, even if one of these operation instances is not being monitored, the monitoring agent can still obtain some metric information for these unmanaged instances, and include them in the service-to-service topology display. Unmanaged instances can be displayed in service-to-service topology views in these cases:

- When a service is called by requesters outside of your monitored environment
- When an ITCAM for SOA version 6.1 data collector is monitoring the operation
- When a third party service is called from within your monitored environment

- When the ITCAM for SOA monitoring agent is configured to use monitor controls to monitor a subset of operations in an operational flow

How service-to-service topology works

Figure 10 illustrates a simple flow that represents the type of observed SOA message traffic that can be displayed in Tivoli Enterprise Portal service-to-service topology workspaces or in the IBM Web Services Navigator views.

Important: The actual service-to-service topology is displayed differently in workspaces and views.

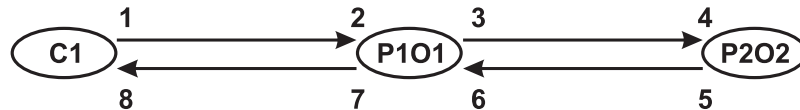


Figure 10. A simple operational flow example

The numbers on the lines between the operations identify interception points where ITCAM for SOA can detect a message, to obtain operation relationship data and transaction flow data. There are four possible interception points for an operation:

- Client Request: interception points 1 and 3
- Client Response: interception points 6 and 8
- Provider Enter: interception points 2 and 4
- Provider Leave: interception points 5 and 7

Operational flows in Tivoli Enterprise Portal represent the relationship between a calling operation (requester) and a target operation (provider). In Figure 10, *C1* represents a stand-alone client web service application that has a relationship with the *P1O1* web service. *P1O1* is a service provider, but it also acts as a web service requester because it sends requests to the *P2O2* web service. For the relationship between *C1* and *P1O1*, *C1* is the requester side of the relationship and *P1O1* is the provider side of the relationship. Similarly, for the relationship between *P1O1* and *P2O2*, *P1O1* is the requester side of the relationship, and *P2O2* is the provider side of the relationship.

In this example, because *C1* is not a web service provider, it does not have a service port name and namespace, or an operation name or namespace. However, if an ITCAM for SOA data collector is monitoring *C1*, the data collector detects the service port and operation to which *C1* is sending its request. For this operational flow, the *C1* application is identified as an operation aggregate with a service port name of *P1 client* and an operation name of *O1 client*.

If a stand-alone client application calls multiple operations, multiple operation instances, and operation aggregates are displayed in the service-to-service topology views for the client application, one for each operation that the stand-alone client calls.

One-way services: A web service can be called as a one-way service, and then can call other services. In this case, the ITCAM for SOA topology displays the called service and the caller for other services as different operations. For example, assume that *P1O1* calls the *Log* operation and the *Log* operation is a one-way

service. If the *Log* operation also calls the *LogToDisk* operation, you see this set of operational flows in the service-to-service topology views:

```
P101 -> Log
LogToDiskClient -> LogToDisk
```

Operational flows and service transaction flows

The operational flow workspace views that are displayed in Tivoli Enterprise Portal show aggregate and instance operational flows, but not service transaction flows. An operational flow indicates that *C1* calls *P1O1*, which calls *P2O2*. However, this flow does not imply that *P1O1* calls *P2O2* every time that *C1* calls *P1O1*.

The Transaction Flows view that is displayed by the IBM Web Services Navigator indicates which operation instance relationships were used for a particular transaction. Here are two service transaction flow examples that use the operation instances in Figure 10 on page 73:

- **Transaction A:** At 2:00 pm EST, *C1* sends a request to *P1O1* on application server *X*, and *P1O1* returns a response to *C1*. On Figure 10 on page 73, the flow is 1->2, 7->8.
- **Transaction B:** At 2:05 pm EST, *C1* sends a request to *P1O1* on application server *X*. During the processing of the operation, *P1O1* sends a request to *P2O2* on application server *Z*, and *P2O2* provides a response back to *P1O1*. *P1O1* then returns a response to *C1*. On Figure 10 on page 73, the flow is 1->2, 3->4, 5->6, 7->8.

The IBM Web Services Navigator can display operational flows and individual service transaction flows from historical data in the Tivoli Data Warehouse or from data collector metric log files. For more information about using Web Services Navigator, see the *IBM Tivoli Composite Application Manager for SOA Tools* guide.

Topology data in the SOA Domain Management Server

ITCAM for SOA provides another function in support of service-to-service topology. To retrieve data about relationships, service ports and operations, deployment environments, and request and response metrics, the SOA Domain Management Server function queries the Tivoli Enterprise Monitoring Agent tables.

SOA Domain Management Server also queries IBM Tivoli Monitoring for information about situations that are triggered against resources that are monitored by ITCAM for SOA. Metric data for the past 24 hours is stored in the SOA Domain Management Server database, and requests for a time frame greater than the last 24 hours cause historic metric information to be retrieved from the data warehouse.

All of this data is correlated and stored in an internal database and used to derive the nodes and links that are used in the service-to-service topology views.

Calculating the status of operations

The SOA Domain Management Server calculates and maintains the status of operation instances. The status of an operation instance is based on the severity of open situations that are related to the operation instance, and can have one of the following values:

- Fatal
- Critical

- Minor
- Warning
- Harmless
- Informational
- Unknown
- Normal

An operation instance with a status such as *Critical*, *Warning*, and so on, is displayed in the topology views with a *status decorator*, or color indicator on the topology icon. If the status is *Normal*, then no status decorator is displayed in the view. The status decorators are the same as the ones that are displayed in the Physical Navigator view for nodes with open situations.

The SOA Domain Management Server periodically polls for the list of open situations, and calculates the status of operation instances based on this algorithm:

- If at least one *Fatal* situation event opens for the operation instance, the status is set to *Fatal*.
- If no *Fatal* situation events open but there is at least one *Critical* situation event open for the operation instance, the status is set to *Critical*.
- If no *Fatal* or *Critical* situation events open but there is at least one *Minor* situation event open for the operation instance, the status is set to *Minor*.
- If no *Fatal*, *Critical*, or *Minor* situation events open but at least one *Warning* situation event is open for the operation instance, the status is set to *Warning*.
- This algorithm continues through the various states, setting the status of the operation instance to the highest level of open situation.
- If no situation events open for the operation instance, the status is *Normal*. However, no status decorator is displayed for a *Normal* status.
- For an instance whose agent is offline, the operation instance is displayed with an inactive icon to indicate it is not available.

Figure 11 on page 76 shows an example of an open situation. In the Navigator Physical view, the highlighted subnode is displayed with a status decorator icon, indicating that a *Critical* situation is open. In the Operational Flow for Application Server view for the selected application server, the highlighted operation aggregate also displays a *Critical* status decorator. The flyover window for the operation aggregate includes the Instance Status attribute indicating that one operation instance in the aggregate has an abnormal state. To display the abnormal operation in the Interaction Detail portion of the view, double-click this operation aggregate. The situation information at the bottom then indicates a fault condition.

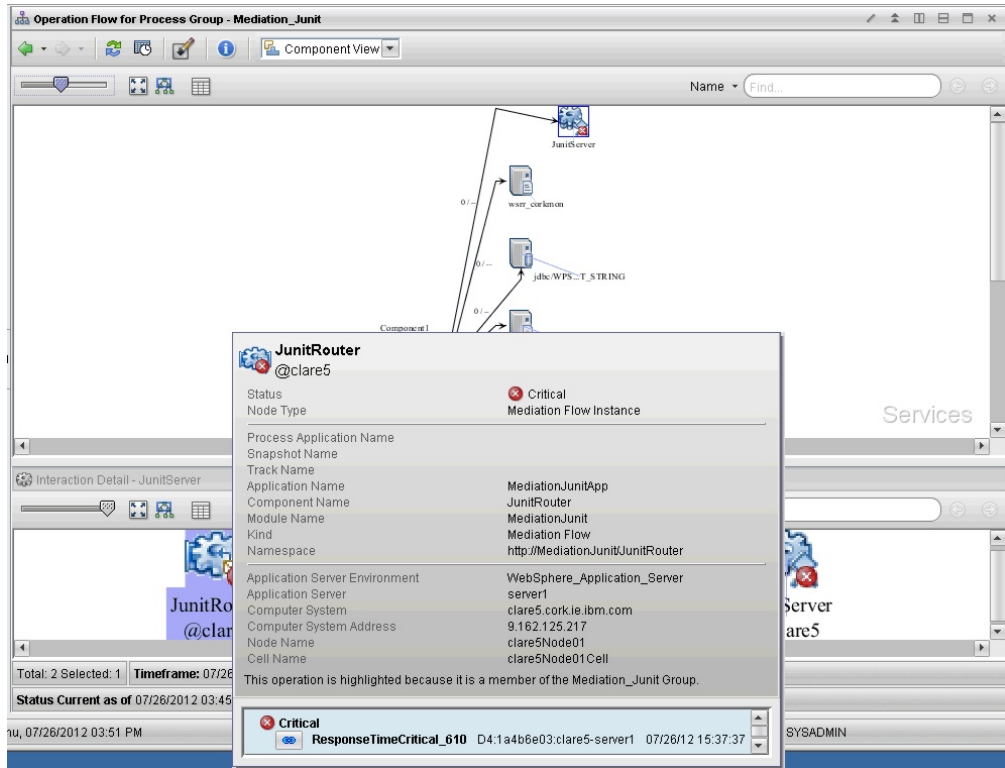




Figure 11. Status example when there is an open situation

From the service-to-service topology views, you can see the list of open situations for an operation instance. To get more details about a selected situation, click the link icon to display the Situation Event Results workspace for that situation.

If the data collector node for an operation instance is offline, the operation

aggregate node is displayed with an *unknown* status decorator: . The

instance node is displayed as inactive: .

Operation status at the aggregate level

The SOA Domain Management Server also tracks the overall health of an operation at the operation aggregate level, and provides this type of information for each operation aggregate node for display in topology views:

- Of the total number of operation instances, the number of operation instances with a status of *Fatal*, *Critical*, *Minor*, *Warning*, *Harmless*, or *Unknown*, or whose data collector subnode is offline.
- A status indicator that reflects the overall health of the operation aggregate. Operation aggregate status is represented by a smaller set of status levels that summarize the status of its instances. For more information about how status is determined, see Appendix B, "Determining status for operation instances, operation aggregates, and groups," on page 331.

The list of open situations is not provided for an operation aggregate node.

Offline: Instances that are offline are classified as *Unknown* for the purposes of determining aggregate status. Any situation status that they might have is ignored.

Unmanaged: The operation instance and operation aggregate status of unmanaged clients and unmanaged operations is always *Normal*.

For more details about how status is calculated to operation instances and operation aggregates, see Appendix B, “Determining status for operation instances, operation aggregates, and groups,” on page 331.

Operation status in the DataPower environment

If a DataPower mediation operation instance is being monitored for multiple data collector nodes in the Navigator Physical view, for all data collector nodes that are monitoring that operation instance, the status indicator for the operation instance is based on the open situations that are associated with that operation.

The configuration of your DataPower environment might affect how the status of DataPower mediation operation instances is displayed in topology views. The following examples demonstrate where the status might not be displayed as expected:

- DataPower subnodes can be used to aggregate data from multiple domains, and situations are triggered based on the aggregated metrics. For example, suppose that you define a subnode *SN1* mapped to two domains, *dom1* and *dom2*. The *dom1* is on DataPower appliance *DP1*, and the *dom2* domain is on DataPower appliance *DP2*. For this example, suppose that operation instance *P1O1* is observed in both *dom1* and *dom2* domains.

Suppose that no messages for *P1O1* in the *dom1* domain exceed the maximum message size threshold. If the size of a message for *P1O1* in the *dom2* domain exceeds the maximum size threshold, the *MaxMessageSize_610* situation with a *Critical* severity is triggered. Because the operation instance status is based on the severity of the *MaxMessageSize_610* situation for *P1O1* in subnode *SN1*, both operation instances for *P1O1* (one in *dom1* and the other in *dom2*) are displayed in the topology views with a *Critical* status, even though the traffic for *P1O1* in the *dom1* domain is not exceeding the threshold in the *MaxMessageSize_610* situation.

- Similarly, suppose that the *dom1* domain is also mapped to a second subnode, *SN2*. Even if no open situations for the *P1O1* operation instance exist in subnode *SN2*, because the *MaxMessageSize_610* situation is open against *P1O1* in subnode *SN1*, the status for the *P1O1* operation instance for the *dom1* domain in subnode *SN2* is also *Critical* because there is at least one subnode with a critical situation open for *P1O1*.

For this example, you can view the list of open situations using the Operational Flow workspace, and see that the *P1O1* operation instance for the *dom1* domain has no open situations for subnode *SN2* but *P1O1* does have a *MaxMessageSize_610* situation open against it for subnode *SN1*.

Additional considerations

The following considerations are important when viewing the status of operation instances and aggregates:

- Only situations that are open for these attribute groups are evaluated when determining the operation status:
 - Services Inventory_610
 - Services Inventory Requester Identity_610 attributes
 - Fault Log_610 attributes
- The **Service Type** attribute in the situation event must have the value of *Provider* for the situation to affect the status of an operation in the service-to-service

topology views. Situations for which the **Service Type** attribute is set to *Requester* do not identify a specific operation instance.

- The status is always the current status even if the selected time frame for the metric data is not the current interval.
- SOA Domain Management Server also keeps a record of when the list of open situations was last updated with status changes. This time stamp is included in updates to the service-to-service topology views so that you can verify that the status is current.
- SOA Domain Management Server uses only situations that are defined for ITCAM for SOA data collector subnodes that are included in the Navigator Physical view when calculating operation status.
 - If you create a logical navigator view, from the Navigator Physical view, copy your ITCAM for SOA data collector subnodes to your logical navigator instead of creating new nodes.
 - If you create ITCAM for SOA nodes (instead of copying them) in your logical navigator and define situations for those nodes, those situations are not factored into the calculation of operation status.
- SOA Domain Management Server polls IBM Tivoli Monitoring for the list of open situations. By default, this polling occurs every 2.5 minutes. In certain cases, the status might not match what you see in the Physical Navigator. For example, the status might be different if you display an Operational Flow workspace before SOA Domain Management Server detects a new open situation or a situation that is closed.

Summarized relationship metrics

Summarized metrics for relationships are captured and displayed in the service-to-service topology views. The following metrics are included:

- The number of messages
- The average, minimum, and maximum message lengths for request messages and for response messages
- The average, minimum, and maximum response time
- The number of faults

Additional considerations for relationship metrics

Keep in mind these additional considerations:

- **Updates to service-to-service topology views:** Metrics for the most recently completed monitoring interval are displayed by default unless you use the Select Time Span window to specify a historical time period. The default monitoring interval is 5 minutes. To ensure that all data for a completed monitoring interval is collected by the ITCAM for SOA data collector, updates to the service-to-service topology views are delayed before the metric data from the most recently completed monitoring interval is displayed. The default delay time is 2 minutes.
- **Historical data older than 24 hours:** The SOA Domain Management Server database holds the last 24 hours of call relationship metric data. To display call relationship metric data that was collected before the last 24 hours, you must enable history collection. For information about this procedure, refer to the *IBM Tivoli Composite Application Manager for SOA Installation Guide*.
- **Unmanaged requester:** For a call relationship that involves an unmanaged requester (client), no metrics are available for the requester response interception point.

- **Unmanaged operation:** For a call relationship that involves an unmanaged operation, no metrics are available for the provider enter and leave interception points.
- **No message traffic or metrics not available:** If no traffic is detected by the ITCAM for SOA data collectors for a relationship during the selected time frame, a value of 0 is displayed for the *provider leave* and *requester response* message count and fault count metrics, and no value is shown for the other metrics. However, if metric data is not available during the selected time frame, no value is displayed for any of the metrics. Metric data is not available under the following conditions:
 - The call relationship did not exist during the selected time frame.
 - The specified time frame results in metric data from the data warehouse being requested, but no metric data is found in the warehouse for that time frame. In this case, ITCAM for SOA cannot determine whether the lack of data means that no message traffic was detected during the time frame or when no metric data was available, so no value is displayed for the metrics.
 - The ITCAM for SOA monitoring agent that is monitoring one or both sides of the call relationship is offline during the specified time frame.
- **One-way services:** For one-way services, metrics are only observed for the *provider enter* interception point. For *requester response* and *provider leave* interception points, the message count and fault count metrics have a value of 0 to indicate that no response was sent. In the case where no metric data is available for the selected time frame, no metric values are displayed for any of the interception points.

If the ITCAM for SOA data collector detects an operation calling a one-way service, but that one-way service is not being monitored by ITCAM for SOA, the call relationship is not displayed in the service-to-service topology views.

- **Synchronizing monitored servers:** To ensure that you retrieve consistent metric data across all of your ITCAM for SOA monitoring agents, on the servers where the agents are installed, keep the date and time synchronized using, for example, a Network Time Protocol server.
- **DataPower consideration:** You might have multiple instances of the ITCAM for SOA data collector monitoring the same DataPower appliance and set of domains. For example, one instance of the data collector is on *host1* and another instance of the data collector is on *host 2*, and both instances are monitoring the DataPower appliance with *dp-server1*).

In this case, multiple instances of the call relationship metrics are stored in the data warehouse, and the resulting multiple sets of metrics are aggregated when you display historical data from the warehouse in the service-to-service topology views. In this example, the message count is doubled because there are two instances of the data collector monitoring the same set of DataPower appliance domains.

Topology display elements

Topology views consist of various graphical elements. These elements represent the operations in an SOA infrastructure, their statuses, and relationships.

Resource type icons for operation aggregates in the view

The Operation Flow view of the service-to-service topology provides a high-level overview of operation interactions; it shows one or more service flows that illustrate call relationships among operation aggregates.

Table 16 describes the most common icons that represent the types of operation aggregate resources in the Operation Flow view.

Table 16. The most common icons and descriptions that are displayed in the service-to-service topology for the Operation Flow view.
















Icon	Description
	The operation aggregate represents all the instances of a specific operation, identified by a unique combination of operation name, operation namespace, service port, and service port namespace, running on all monitored application servers.
	The unmanaged client aggregate covers both the unmanaged clients and unmanaged (operation) providers. This resource type is always the caller in a call relationship.
	The unmanaged operation aggregate is an aggregate of all the instances of any operation running on an application server that is <i>not</i> being managed by the ITCAM for SOA monitoring agent. This resource type is always the target in a call relationship.
	An aggregate of all the instances of a specific DataPower mediation operation, identified by a unique combination of operation name, operation namespace, service port, and service port namespace, running on any monitored DataPower appliance. Because this icon represents a mediation, it includes a background image of a pipe, to signify that the mediation processes the message traffic and then passes it on to the next operation aggregate.
	An aggregate of all the instances of a specific SCA mediation operation, identified by a unique combination of operation name, operation namespace, service port, and service port namespace, running on any monitored SCA environment. Because this icon represents a mediation, it includes a background image of a pipe, to signify that the mediation processes the message traffic and then passes it on to the next operation aggregate.
	An aggregate of all the instances of a specific WebSphere Message Broker mediation operation, identified by a unique combination of operation name, operation namespace, service port, and service port namespace, running on any monitored Message Broker. Because this icon represents a mediation, it includes a background image of a pipe, to signify that the mediation processes the message traffic and then passes it on to the next operation aggregate.
	An aggregate of all the instances of a BPEL process.

Table 16. The most common icons and descriptions that are displayed in the service-to-service topology for the Operation Flow view. (continued)

Icon	Description
	An aggregate of all the instances of an SCA human task.
	An aggregate of all the instances of an SCA mediation flow.
	An aggregate of all the instances of an SCA mediation subflow.
	An aggregate of all the instances of a database system, which is an external system connected to an SCA mediation flow. It searches values from a database and stores them in the message.
	An aggregate of all the instances of a WebSphere Service Registry and Repository Policy Resolution, which is an external system connected to an SCA mediation flow. It dynamically configures a mediation flow by using mediation policies retrieved from WebSphere Service Registry and Repository.
	An aggregate of all the instances of an SCA Java component.
	An aggregate of all the instances of an SCA outbound flat file adapter.
	An aggregate of all the instances of an SCA inbound flat file adapter.

Resource type icons for operation instances in the view

The Interaction Detail portion of the Operation Flow view displays the detail about the operation instances that are part of the selected operation aggregate. Table 17 describes the icons that represent the types of resources in the Interaction Detail portion of the view.

Table 17. The icons and descriptions that are displayed in the service-to-service topology for the Interaction Detail portion of the view.



Icons	Descriptions
	The operation instance is a specific operation, identified by a unique combination of operation name, operation namespace, service port, and service port namespace, running on a specific monitored application server.
	The unmanaged client is the client (or clients) of an operation. Each operation instance in the topology graph might have an unmanaged client calling it. This resource type is always the caller in a call relationship.

Table 17. The icons and descriptions that are displayed in the service-to-service topology for the Interaction Detail portion of the view. (continued)














Icons	Descriptions
	The unmanaged operation is an instance of any operation running on a specific application server that is not being managed by the ITCAM for SOA monitoring agent. This resource type is always the callee in a call relationship.
	This is a specific DataPower mediation instance of an operation on a DataPower appliance that is uniquely associated with this type of mediation. Because this icon represents a mediation, it includes a background image of a pipe, to signify that the mediation processes the message traffic and then passes it on to the next operation.
	This is a specific SCA mediation instance operation on a specific SCA environment that is uniquely associated with this type of mediation. Because this icon represents a mediation, it includes a background image of a pipe, to signify that the mediation processes the message traffic and then passes it on to the next operation.
	This is a specific WebSphere Message Broker mediation instance of an operation on a specific Message Broker that is uniquely associated with this type of mediation. Because this icon represents a mediation, it includes a background image of a pipe, to signify that the mediation processes the message traffic and then passes it on to the next operation.
	An instance of a BPEL process.
	An instance of an SCA human task.
	An instance of an SCA mediation flow.
	An instance of an SCA mediation subflow.
	An instance of a database system, which is an external system connected to an SCA mediation flow. It searches values from a database and stores them in the message.
	An instance of a WebSphere Service Registry and Repository Policy Resolution, which is an external system connected to an SCA mediation flow. It dynamically configures a mediation flow by using mediation policies retrieved from WebSphere Service Registry and Repository.

Table 17. The icons and descriptions that are displayed in the service-to-service topology for the Interaction Detail portion of the view. (continued)

Icons	Descriptions
	An instance of an SCA Java component.
	An instance of an SCA outbound flat file adapter.
	An instance of an SCA inbound flat file adapter.

Status indicators for resource type icons

ITCAM for SOA determines operation status based on situations that are associated with specific operation instances. For more information about determining the status, see “Calculating the status of operations” on page 74.

To obtain the most effective status representation in your Tivoli Enterprise Portal workspaces and views, you must ensure that situations are defined and deployed in a way that makes the most sense for your SOA. At higher levels, you must create and maintain your service groups in a way that also makes the most sense for your SOA. To the extent that both the situations and the service groups are defined correctly for your environment, the resulting status is also correct and representative.

The operational flow views include the following status icons on operation instance and operation aggregate levels.

Table 18. Operation instance and operation aggregate status icons













Operation instance level status		Operation aggregate level status	
	Fatal		Fatal
	Critical		Critical
	Minor		Warning
	Warning		

Table 18. Operation instance and operation aggregate status icons (continued)

Operation instance level status		Operation aggregate level status	
	Harmless	(No state symbol is used when there are no situations opened)	(Normal)
	Informational		If the status for an operation aggregate is assumed to be <i>Normal</i> for purposes of calculating the service group status, but there is at least one operation instance with an open situation or an offline subnode, the purple diamond icon () is displayed in the service-to-service topology views, indicating an <i>Abnormal</i> status for the operation aggregate.
(No state symbol is used when there are no situations opened)	(Normal)		
	Unknown		Unknown

Offline: Operation instances whose monitoring agents are offline are classified as *Unknown* for the purposes of determining the operation aggregate status. Any situation status that is associated with these operation instances is ignored.

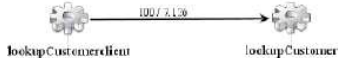
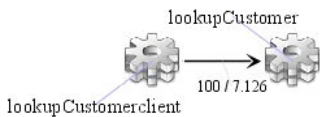
Unmanaged: Status indicators are not displayed for unmanaged client aggregates or unmanaged operation aggregates. Their implied status is *Normal*.

Relationships between resource types

The service-to-service topology displays an operational relationship flow with a solid line and an arrow, and any available metrics.

Table 19 on page 85 describes the defined call relationship for an operation instance and operation aggregate.

Table 19. The style and the call relationship in the service-to-service topology.

Relationship Type	Line Type	Resource Relationship	Description
Call	<p>A solid line with an arrow and metrics.</p> <p>This image represents an operation instance call path.</p>  <p><i><Server Enter message count> / <Server Leave average response time in seconds></i></p> <p>The value of either or both of these metrics might not be known. In these cases a dash (-) is used. For more information about metric data that is not available for display, see “Summarized relationship metrics” on page 78.</p> <p>This image represents an operation aggregate call path.</p> 	<ul style="list-style-type: none"> • Operation instance-> Operation instance • Unmanaged client-> Operation instance • Operation instance->Unmanaged operation 	<p>A directed edge or link that connects two operation instance nodes or two aggregates representing a call relationship between the two nodes where the source node calls the destination node.</p>

SCA component display

ITCAM for SOA version 7.2 or later displays SCA components and operations within operational flow workspaces.

Important: If a topology workspace displays an SCA component or operation, but some metrics and links are absent, check the version of the ITCAM for SOA agent installed on the application server where this component or operation runs. If the version is lower than 7.2, upgrade the agent to the latest version.

Component and operation views

You can switch between component and operation topology views for SCA components.

Some SCA components, such as BPEL processes and Java components, might implement multiple operations. The operations are defined in the interface of a component.

In component view, a single node represents each SCA component. The flyover window and details window (see “Viewing details” on page 110) contain information for the entire component.

All incoming and outgoing call relationships for all operations in the component are displayed on the component node.

If two or more operations in a component have the same call relationship to another node, only one link is displayed while there are no performance metrics (that is, actual calls are not being made).

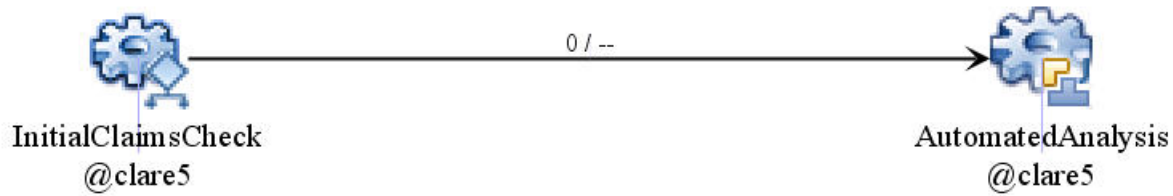


Figure 12. A call relationship without metrics

When performance metrics become available, each call relationship with metrics is displayed as a separate link.

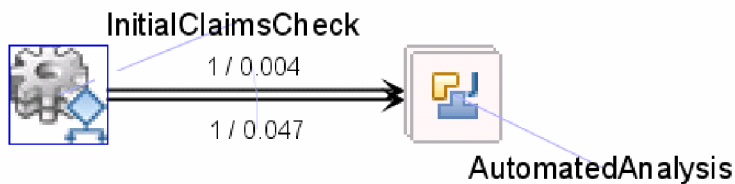


Figure 13. A call relationship with metrics; two operations in one component

To view the operation information for a call relationship, hold the mouse cursor over the link arrow. In the flyover window, the label above the arrow is the target operation name.

In operation view, every operation is represented by a separate node. The flyover window and details window contain information for the single operation.

Calls between operations (within the same component or between different components) are represented by links.

Sometimes a component might export an operation that another component imports, but no calls are monitored between them. Such a connection is called a *logical link* and represented by a dash line.

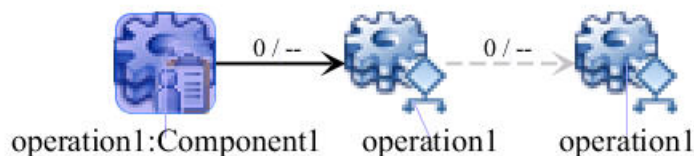


Figure 14. A call link and a logical link in Operation view.

To switch between component and operation view, use the **Component view/Operation View** list in the toolbar of the topology workspace window.



Figure 15. The **Component view/Operation View** list in the toolbar.

Example

An example application is developed using IBM Integration Designer. A Stand-alone Human Task component named `StartClaimsCheckProcess` is linked to a BPEL Process component named `InitialClaimsCheck`. This process component is linked to a Java component named `AutomatedAnalysis` and a BPEL process component named `DetailedClaimsAnalysis`. This second BPEL process component is linked to another Stand-alone Human Task component named `AssessAccidentDescription`.

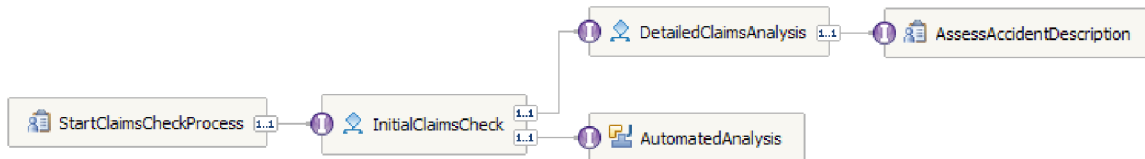


Figure 16. Example application in Integration Designer

By default, a Topology workspace for the application displays the component view. In this view, each component is represented by a single icon.

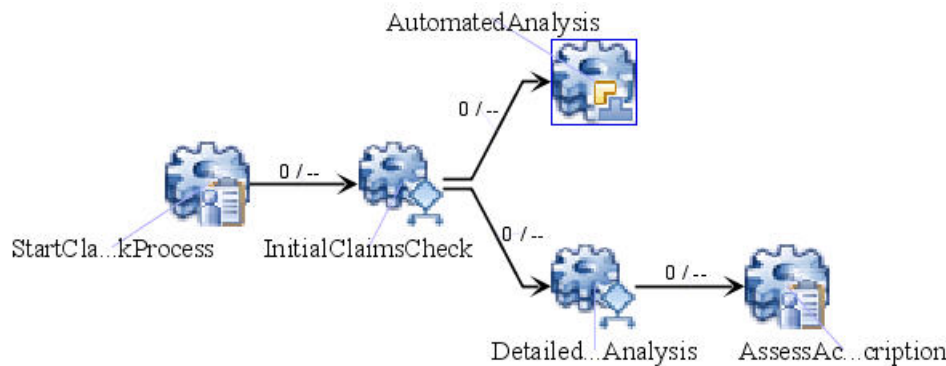


Figure 17. Example application in the Topology workspace, component view

When you switch the workspace to the operation view, each operation within the components is represented by its own icon.

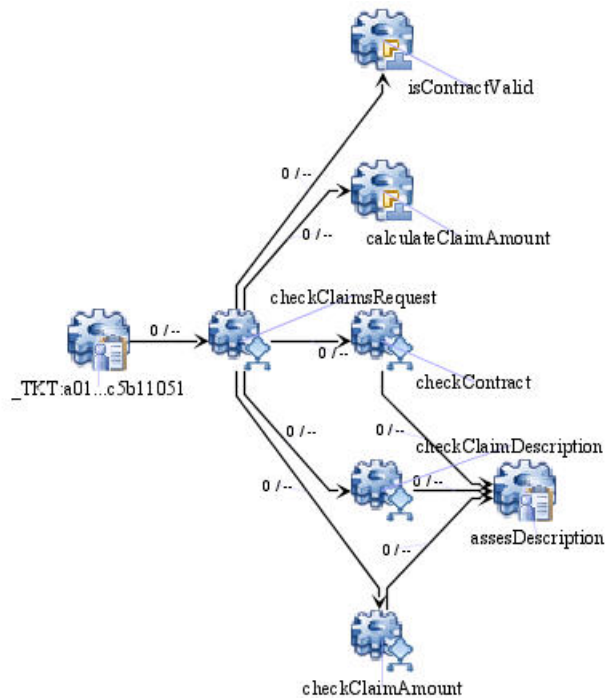


Figure 18. Example application in the Topology workspace, operation view

Figure 19 shows which operations belong to every component in the application.

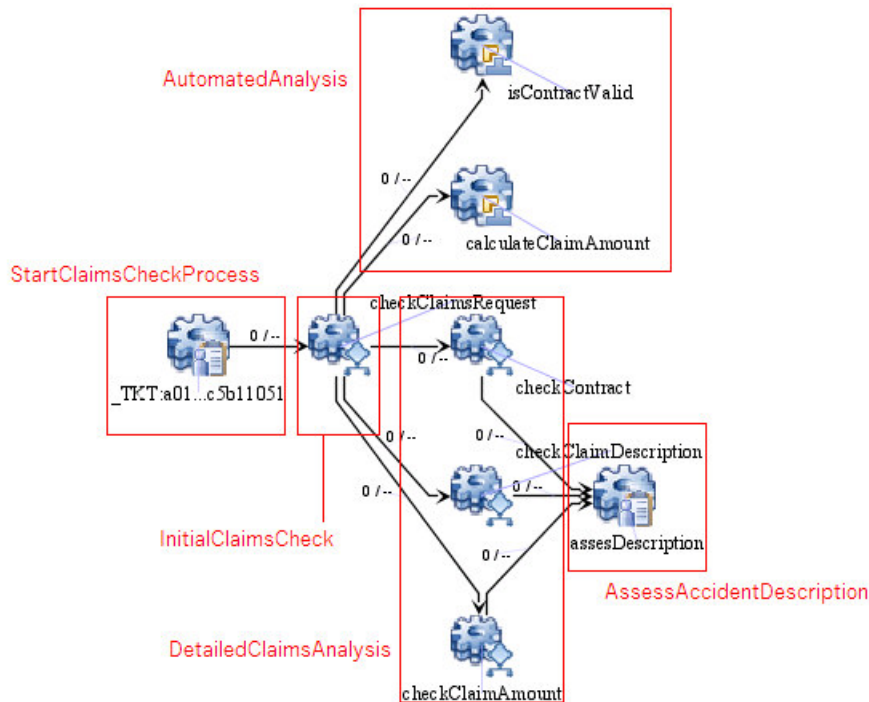


Figure 19. Affiliation of operations in the example application

Static and dynamic data

ITCAM for SOA uses both static and dynamic data to display SCA component links.

The agent gathers *static data* when an application is installed and started on the BPM server. ITCAM for SOA uses static data to display all SCA component and

operation nodes and the links between them (except the links that are based on web service and HTTP bindings). A link that is based on static data is displayed when all the applications involved in the link are first started.

To display all other links, and external system nodes, ITCAM for SOA uses *dynamic data*. This data is gathered when calls are intercepted. A link that is based on dynamic data becomes available only after an actual call is made.

All other ITCAM for SOA topology information and all metrics are based on dynamic data.

BPEL components

A *Business Process Execution Language* (BPEL) component implements a model of a business process workflow, consisting of individual activities. BPEL components are created and edited in IBM Integration Designer.

In the component view, the operational flow views display a BPEL component as a single icon. In the operation view, each operation defined in the component interface is displayed as a separate icon.

In the details window you can see the list of activities within the component or operation, the execution metrics for each activity, and information about the type and number of errors that happened in a recent period. For more information about the details window, see *Viewing Details*.

Call relationships between the BPEL component or its operations and other components and services are displayed as links.

Links between BPEL components and other SCA components are displayed based on static data, when all the applications involved are installed and started. Other links are displayed based on dynamic data, after at least one call is monitored. When ITCAM for SOA monitors new calls, it adds new links.

Mediation flow components

A *mediation flow* component processes data to mediate between two or more services. It performs various analysis and type conversion, and routes information to different services depending on content.

A mediation flow component consists of one or more *mediation primitives*, which represent single steps in information processing. A mediation flow can include *mediation subflows*, which are separate flows consisting of primitives. A subflow can include other subflows.

Some mediation primitives call an external database or a registry.

In the component view, the operational flow views display a mediation flow component as a single icon. Each mediation subflow and each external system is displayed as a separate icon.

In the operation view, each operation that is defined in the interface of a mediation flow is displayed as a separate icon. Mediation subflows and external systems are displayed as separate icons, in the same way as they are displayed in a component view.

In the details view you can see the list of mediation primitives and subflows within the flow, operation, or subflow, as well as performance metrics for them. For more information about the details window, see [Viewing Details](#).

Call relationships between the mediation flow and other components, services, and external systems are displayed as links.

If a mediation flow passes the information to other subflows, BPEL processes, and SCA components, ITCAM for SOA displays the link at all times when the components are installed, based on static data.

If a mediation primitive interacts with a database or registry, ITCAM for SOA also displays the link when the component is installed, based on static data. If the component is then reconfigured to use another external system, ITCAM for SOA displays the new link and external system icon when a call is made, based on dynamic data. Therefore, in this case, new links and icons are displayed, in the view.

Links to other services are displayed based on dynamic data, when a call is made.

Other SCA components

SCA components other than BPEL processes and mediation flows are also displayed in topology views.

In the component view, the operational flow views display an SCA component as a single icon. In the operation view, each operation that is defined in the component interface is displayed as a separate icon.

Call relationships between the SCA component or its operations and other components and services are displayed as links.

Links between SCA components, except links based on web service and HTTP bindings, are displayed when the applications are installed based on static data.

Other links are displayed based on dynamic data after at least one call is monitored. When ITCAM for SOA monitors new calls, it adds new links.

Important: SCA adapters provide interaction with a data source, such as a file or database. Such data sources are not included in the topology. The topology views include links only between adapters and SCA components.

Business Process Definition display

A Business Process Definition (BPD) is a model of a business process workflow, consisting of individual activities. The activities can depend on user input. BPDs are created and edited in the Process Designer, an Eclipse-based integrated development environment.

The operational flow views display a BPD process as a single icon.

In the details view you can see the list of activities and gateways within the component, and the execution metrics for each activity and gateway. For more information about the details window, see [Viewing Details](#).

ITCAM for SOA displays the following call relationships for a BPD:

- Call to a BPD from an unmanaged client, (for example, a service on another system). In the present version, a call from the Web interface is also displayed as a call from an unmanaged client.
- Call from a BPD to another BPD.
- Call from a BPD to an SCA component, (including a BPEL process), and from an SCA component to a BPD.

Important: The current version of ITCAM for SOA displays call metrics for calls from a BPD to an SCA component. It displays the message count, but not the response time, for calls from an SCA component to a BPD. It does not display metrics for any other calls from or to a BPD (including calls from a BPD to another BPD). For more information about viewing call metrics, see “Viewing metrics” on page 107.

Calls from an SCA component to BPD are displayed based on static data, when the applications are installed and started. All other relationships involving a BPD are displayed based on dynamic data, after at least one call is monitored. For more information about static and dynamic data, see “Static and dynamic data” on page 88.

Calls between a BPD and other services are not displayed in the present version.

Important: If you change a BPD while it is running, the displayed metrics for this BPD will be incorrect. If you start a new instance of a changed BPD, a new node is added to the topology; the node might look like a duplicate. If you have changed a BPD, delete the node that represents the BPD.

Operational Flow workspaces

Service-to-service topology information is displayed in these workspaces:

- “Operational Flows workspace” on page 92
- “Operational Flow for Operation workspace” on page 93
- “Operational Flow for Application Server workspace” on page 95
- “The Group Summary workspace” on page 152. For information about displaying topology in this workspace, see “Displaying the topology view for a group” on page 161.

Within any operational flow workspace, a split pane divides the topology into two logical levels of detail. The upper pane contains the *Operation Flow* view, displaying the overall flow of operations and their relationships to each other. This view depicts service operation calls as call relationships in a topology graph. This view aggregates the data across all instances of an operation. The aggregation of operation instances is called an *operation aggregate*. An operation aggregate is uniquely identified by the combination of service port name, service port namespace, operation name, operation namespace, and the mediation type. The Operation Flow view provides metrics and status indicators to help you with isolating problems and analyzing the effect of problems on your deployed services.

When you double-click a node in the Operation Flow view or right-click a node in the Operation Flow view and select Show Interaction Detail, the lower pane, called the *Interaction Detail* view, displays that portion of the topology in more detail.

The Interaction Detail view displays calls among deployed service operations and other nodes at the *operation instance* level of detail. In An operation instance, like an

operation aggregate, is uniquely identified by the combination of service port name, service port namespace, operation name, operation namespace, and mediation type for a particular application server environment where the service port and operation are deployed. Metric labels are displayed in the view with links that represent the call relationships. Status indicators denote operation instances that are in an abnormal state as a result of the situation events that are associated with them. Metric values that are associated with each call relationship link are also displayed.

The Interaction Detail view displays only a subset of the operational flow. For the selected operation, the Interaction Detail view displays these items:

- The instances of the operation
- The set of operation instances that call the operation
- The set of operation instances that it calls

If the number of objects in a view exceeds the maximum that is specified in the view properties, the topology is displayed in table mode rather than graphical mode.

In any operational flow workspace, you can use the **Component view/Operation View** list in the toolbar to switch between component and operation views. This setting applies to both the Operation Flow and Interaction Detail views and it determines the way SCA components are represented. All other nodes are not affected. For more information about the component and operation views, see “Component and operation views” on page 85).

A search function is also provided to help you locate specific objects in the topology display.

Operational Flows workspace

The *Operational Flows* workspace is one of several workspaces that are associated with the Services Management node in the Navigator ITCAM for SOA view. The workspace displays all of the operational flows that are observed by the data collectors that monitor the SOA and the details about those operational flows.

Figure 20 on page 93 displays an example of the Operational Flows workspace.

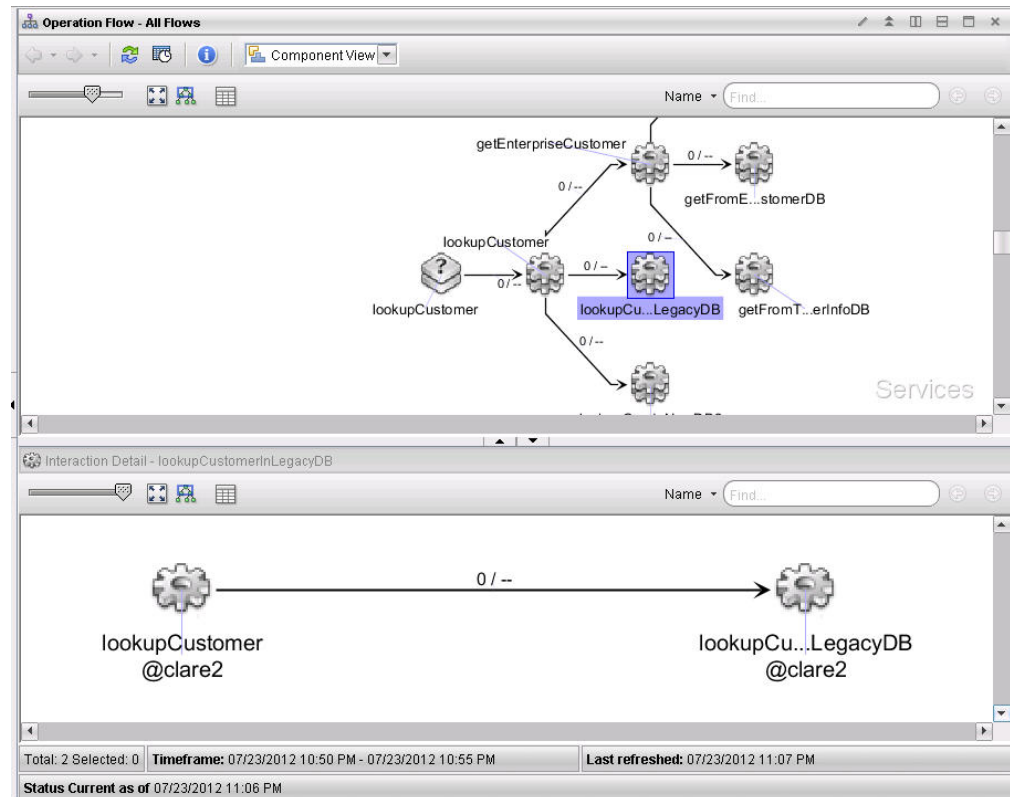


Figure 20. The Operational Flows workspace

Figure 21 displays an example of the Interaction Detail portion of the Operational Flows workspace, with additional metric summary information that helps you to understand and manage your SOA environment. To view such metric information, hold the mouse cursor over a line representing a call relationship.

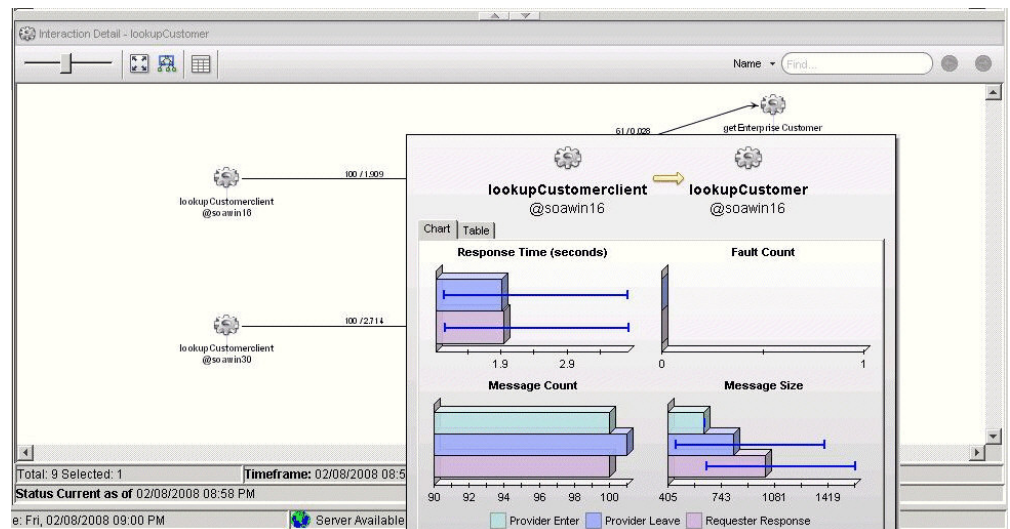


Figure 21. Metric information displayed with the Operational Flows workspace

Operational Flow for Operation workspace

This workspace displays the operational flow that contains a specific operation and the details about that flow. This workspace is available only as the target of a

workspace link in the context of a specific operation. This workspace is associated with the Performance Summary workspace for the data collector that monitored the operation.

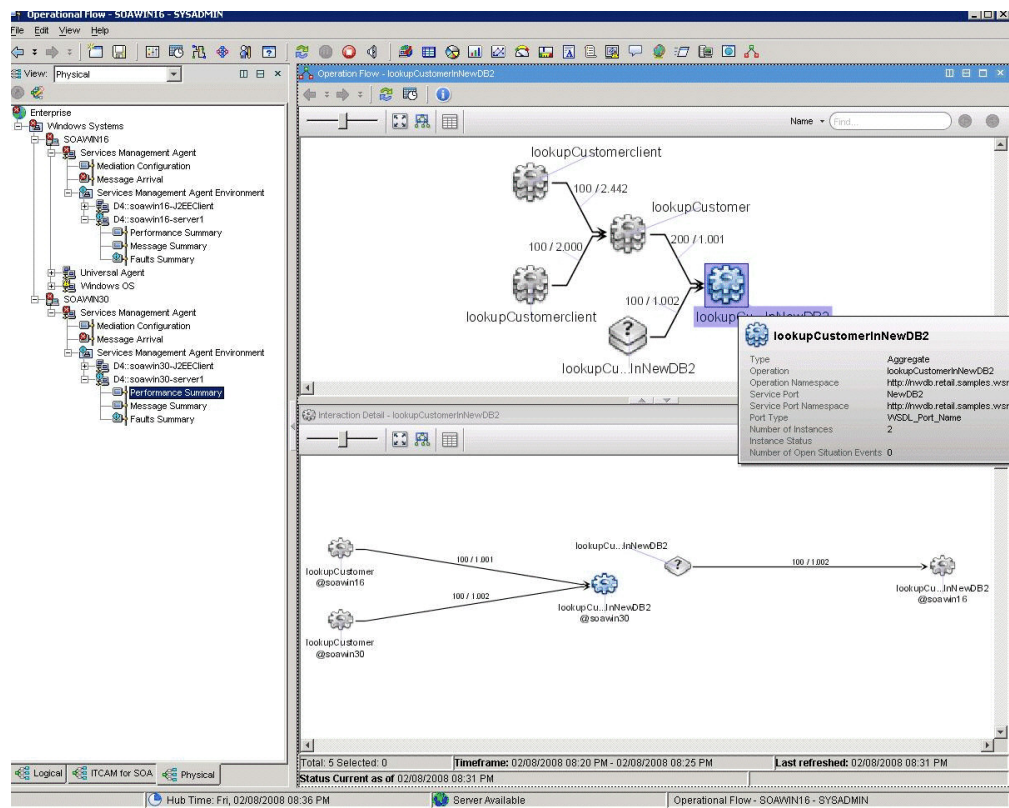


Figure 22. The Operational Flow for Operation workspace

This workspace has the same format as the Operational Flows workspace and can be used to complete the same actions.

The operation that you select as the context for the link into this workspace is preselected in the Interaction Details portion of the view and also highlighted in the Operation Flow and Interaction Details portions of the Operation Flow for Operation view.

The selected operations are highlighted in both the Operation Flow and Interaction Detail portions of the view. The reason for the highlighting is explained in the flyover window for the operation.

Similar to the Operational Flows workspace, the title of the Operation Flow view is followed by the name of the operation aggregate instance, for example, *dbLookup*. When you double-click an operation aggregate, the name of the aggregate is included in the title of the Interaction Detail portion of the view.

To link to this workspace, the operation that you selected as the context for the link must have a Service Type value of *Provider* if you are linking from a row in the Services Inventory table of the Performance Summary workspace, or from a situation in the Situation Event Results workspace. If the value of the Service Type attribute is *Requester*, the Operational Flow for Application Server workspace is displayed instead.

Operational Flow for Application Server workspace

The *Operational Flow for Application Server* workspace is a secondary workspace that is associated with a particular application server that is being monitored by the ITCAM for SOA Tivoli Enterprise Monitoring Agent. This workspace displays the operational flows involving the selected application server environment.

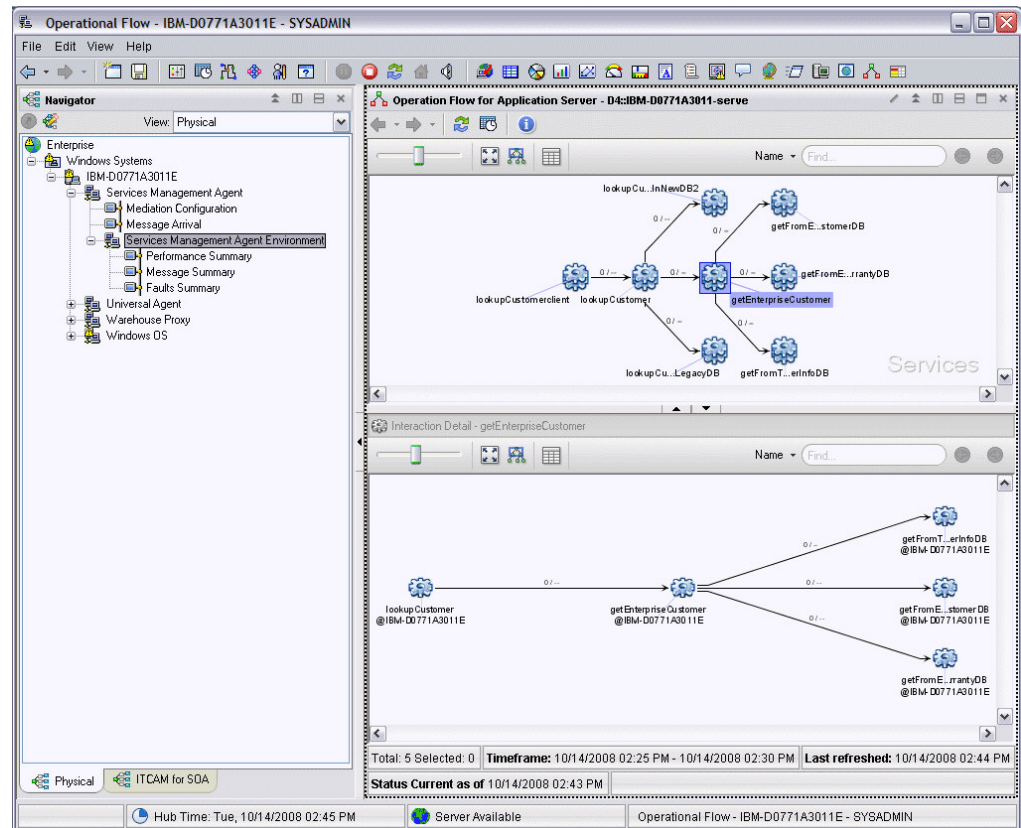


Figure 23. The Operational Flow for Application Server workspace

This workspace has the same format and the same set of views as other Operational Flows workspaces, and can be used to complete the same actions. Similar to the Operational Flow for Operation workspace, this workspace displays only operational flows in which the selected application server participates.

To see the operational flow for operations that are not deployed on the selected application server, select an operation instance in the Interaction Detail portion of the Operation Flow for Application Server view. Then, link from it to its Operational Flow for Operation workspace.

The operations that are associated with the application server are highlighted in both the Operation Flow and Interaction Detail portions of the view. The reason for the highlighting is explained in the flyover window for the operation.

Navigating the service-to-service topology

The service-to-service topology is accessed from workspace links in the “The Navigator Physical view” on page 24 and the “The Navigator ITCAM for SOA view” on page 26.

The Operational Flow workspaces differ in context, in where they are accessed and in their contents.

Table 20 describes the variations of the Operational Flow workspaces, the access points, and the initial content of each view.

Table 20. The Operational Flow workspace variations, access points and initial view content.

Workspace	Access Point	Operational Flow	Interaction Detail
Operational Flow for Operation	From a situation event detail, various ITCAM for SOA agent data collector workspaces.	Displays the flows that the operation aggregate participates in. The operation instance is preselected.	Displays the flows containing operation instances, with the specific operation instance preselected.
Operational Flow for Application Server (D4 subnode)	An alternate workspace to the subnode in the Navigator Physical view.	Displays all of the operational flows that have at least one operation instance hosted on the specified subnode.	Initially empty. You must double-click an operation aggregate from the Operational Flow portion of the view.
Operational Flows (All operational flows)	An alternate workspace to the root node of the Navigator ITCAM for SOA view.	Displays all known operational flows.	Initially empty. You must double-click an operation aggregate from the Operational Flow portion of the view.

In any of the Operational Flow workspaces, you can use predefined links to other ITCAM for SOA workspaces. If ITCAM Agent for WebSphere Applications is installed, some nodes provide links to workspaces from this agent as well.

You can create your own dynamic workspace links using the Tivoli Enterprise Portal Workspace Link Wizard and referencing the supported link symbols.

From the Situation Event Results workspace to an Operational Flows workspace

When you observe a triggered situation event for data collector metrics in the Navigator Physical view, you can follow this basic procedure to help determine the cause:

1. Hold the cursor over the event indicator in the Navigator Physical view to display the situation event hover help for the affected data collector node.
2. In the situation event hover help, click the link indicator to navigate to the Situation Event Results workspace.
3. Examine the metric data in the Initial Situation Values table view of the Situation Event Results workspace to understand what triggered the situation event.

The rows in the Initial Situation Values table view of the Situation Event Results workspace include Service Type values of *Requester* and *Provider* because the operation instance nodes in the topology graph represent the provider side of requests.

4. To begin your investigation of the possible cause of the event, you can select a workspace link to one of the Operational Flow workspaces.

This link is enabled *only* for situations that are associated with the required context of service port name, service port namespace, operation name, and operation namespace, and whose metric data is from an ITCAM for SOA data collector at version 7.1.0 or later.

Service Type = *Provider*: If the situation event is for a row in the table with a Service Type of *Provider*, use the Operational Flow for Operation workspace for the instance that processed the request.

Service Type = *Requester*: If the situation event is for a row in the table with a Service Type of *Requester*, it might not be obvious whether you should investigate the requester side or provider side of the request. In this case, you might link to the Operational Flow for Application Server workspace to identify the set of flows that most likely involve what you need to explore.

From a row in the Services Inventory table view to an Operational Flow workspace

For a particular row of metric data in the Services Inventory table view of the Performance Summary workspace, you might want to view the service port and operation relationship in context of the operational flow, and to see the instance details of who calls that operation and who it calls. Note that this is available for metric data from a data collector subnode in ITCAM for SOA version 7.1.0 or later.

You can click the workspace link in the row of the table for the selected service port and operation relationship. Depending on whether it is for a *Provider* or *Requester* service type, you are taken to either the Operational Flow for Operation workspace or the Operational Flow for Application Server workspace.

From an operation instance in the Operational Flows workspace to the Operational Flow for Operation workspace

The “Operational Flows workspace” on page 92 shows *all* of the monitored flows in the environment, and provides an overview that might contain many nodes.

You can right-click an operation instance in the Interaction Detail portion of the Operation Flow view and select a workspace link to the “Operational Flow for Operation workspace” on page 93 to observe the *single* operational flow at the level in the Navigator Physical view for the ITCAM for SOA data collector that observed the operation instance.

From an operation instance in the Operational Flow for Application Server workspace to the Operational Flow for Operation workspace

At the data collector level, the “Operational Flow for Application Server workspace” on page 95 displays *all* of the monitored flows in the application server runtime environment, but you might need a more focused view.

You can right-click an operation instance in the Interaction Detail portion of the Operation Flow view and select a workspace link to the “Operational Flow for Operation workspace” on page 93 to view a *single* operational flow at the level in the Navigator Physical view for the ITCAM for SOA data collector that observed the selected operation instance.

From an operation instance in any Operational Flows workspace to the Requester Identities for Operation workspace

You might observe a particular operation instance in any Operational Flows workspace and want more information about the monitored requester identities for that operation instance.

If monitoring by requester identity is enabled for requester identities that call the specified operation instance, you can right-click the operation instance and select the workspace link to the “Requester Identities for Operation workspace” on page 50 for the Performance Summary node of the ITCAM for SOA data collector that observed the service port and operation relationship.

The Requester Identities for Operation workspace displays the list of monitored requester IDs that have called the operation instance.

From a DataPower operation instance in any Operational Flows workspace to the DataPower Console workspace

If you are viewing a DataPower operation instance in an Operational Flow workspace, you might want to navigate to the DataPower Console workspace for the associated DataPower appliance that is being monitored.

You can right-click the operation instance and select the workspace link to the DataPower Console to display the DataPower WebGUI within the Tivoli Enterprise Portal.

From a data collector node in the Navigator Physical view to the Operational Flow for Application Server workspace

You might want to see an overview of all the monitored operational flows that were observed to go through a particular application server runtime environment.

You can select the workspace link on the menu item used to select the secondary “Application Server Services Management workspace” on page 39 for a version 7.1.0 or later data collector node in the Navigator Physical view. Use this link to navigate to the “Operational Flow for Application Server workspace” on page 95 for the data collector that observed all the monitored operational flows through that specific application server runtime environment.

From an operation instance in any Operational Flows workspace to the Performance Summary workspace

If you are viewing a particular operation instance in an Operational Flows workspace, you might want to navigate to the associated “Performance Summary workspace” on page 41 to see additional metric information displayed in bar charts and table views. You can use this additional information to further assist you in problem determination at the data collector level.

You can right-click the selected operation instance in the Operational Flows workspace and select the workspace link to the Performance Summary workspace.

From a view in the Services Management workspace to the Operational Flow for Application Server workspace

If you are viewing the static topology definition of a service and the application server where it is deployed, you might want to see an overview of all of the monitored operation flows passing through that application server.

From either the “Service Port Details view” on page 61 or the “Service Details view” on page 58 in the “Services Management workspace” on page 55, you can right-click an application server instance and select the workspace link to the “Operational Flow for Application Server workspace” on page 95 for the data collector that observed the application server instance.

From a node in the Interaction Detail view to the Business Process Manager Summary workspace

If ITCAM Agent for WebSphere Applications is installed, you can drill down from a node in the Interactions Details pane of an Operations Flow workspace to the Business Process Manager Summary workspace in the Agent for WebSphere Applications.

When you right-click on a node, the **Server Summary** link is visible if the Agent for WebSphere Applications is installed and the selected node either an SCA component or a BPD.

The workspace displays summary information for the application server where the SCA component or BPD runs. You can drill down to details for any necessary applications.

Additional features of Operational Flow workspaces

In each of the Operational Flow workspaces, several additional features and controls are available.

Using resource actions in the Operation Flow portion of the view

When you are using the Operation Flow views, the following resource actions are available from the menu after you select an operation aggregate.

Table 21 describes the resource actions that are applicable in the graph or table viewing area.

Table 21. Resource actions for operation aggregates in the Operation Flow view

Resource Action	Resource Type Enablement	Function
Show Interaction Detail	All nodes	The Interaction Detail view is updated to display all the instances of the selected operation aggregate and any call relationships.

Table 21. Resource actions for operation aggregates in the Operation Flow view (continued)

Resource Action	Resource Type Enablement	Function
Refocus	All nodes	The view is centered on the selected aggregate enabling you to see the flow through that node and also updates the instance view to show instances of that aggregate. This resource action is <i>only</i> available in the Operational Flow for Operation workspace.
Show Details	All nodes	The Details window is displayed for the selected node. See Viewing details.
Show Metrics	Call relationships	The Metrics notebook is displayed for the selected link. See Viewing metrics.
Show Business Processes	Operation aggregates	The business process information is displayed. This action is only displayed and is available as determined by the ability of the SOA Domain Management Server to verify the presence of the Tivoli Common Object Repository. See Viewing business processes.
Manage Groups	Operation aggregates	The Groups window is displayed.
Zoom to selected	Operation aggregates	Zooms in the display on the selected operation aggregate and centers it in the view.

Using resource actions in the Interaction Detail portion of the view

When you are using the Interaction Detail portion of any Operation Flow view, the following resource actions are available from the menu after you select an operation instance.





Table 22 on page 101 describes the resource actions that are applicable in the topology graph or table viewing modes.

Table 22. Resource actions for operation instances in the Interaction Detail portion of the view

Resource Action	Resource Type Enablement	Function
Take Action	All nodes	The standard Tivoli Enterprise Portal function that enables you to initiate system commands and other types of actions you might define. The set of actions that is available at any specific time for a node is a function of the affinity of that node. For example, within the Operational Flows workspace that is associated with the data collector node, all the Take Action commands are available. However, the Operational Flows workspace that is associated with the Services Management node on the Navigator ITCAM for SOA view does not have any Take Action commands.
Link To	All nodes	<p>The standard Tivoli Enterprise Portal function using workspace linking from one context to another. ITCAM for SOA provides a set of predefined workspace links for operation instances. These links apply only to managed operation instances.</p> <p>You can define custom link definitions or use predefined links. The predefined links that are available for a node are dependent on the link definitions. When you select the link icon, the links are dynamically evaluated to ensure that they are applicable for a specific context and appropriate for a specific node. See Navigating the service-to-service topology</p> <p>A predefined links is provided to an ITCAM Agent for WebSphere Applications workspace. See “From a node in the Interaction Detail view to the Business Process Manager Summary workspace” on page 99</p> <p>A set of predefined workspace links is provided for DataPower appliance instances. For example, any Operational Flow workspace to the DataPower WebGUI. See “DataPower Console workspace” on page 46.</p>
Launch		The standard Tivoli Enterprise Portal function that enables you to define or initiate launches to external applications.
Delete	All operation instances	Initiates a request to the SOA Domain Management Server to delete the selected operation instance.
Show Details	All nodes	<p>The Details window is displayed for the selected operation instance.</p> <p>See Viewing details.</p>
Show Metrics	Call relationships	<p>The Metrics notebook is displayed for the selected link.</p> <p>See Viewing metrics.</p>

Using toolbar functions

At the top of the workspace the Tivoli Enterprise Portal topology toolbar includes the following functional icons:

- Click the  **Refresh** icon to reissue queries to SOA Domain Management Server for the latest topology data to update the display in the Operational Flow view and the Interaction Detail view.
- Click the  **Time Span** icon to define the time frame for the metric data displayed on the links in the topology display. See “Selecting the time span for topology metrics” on page 120 for more information on this feature.
- Click the  **Informational** icon to instantly access additional information for a selected operation aggregate or instance. For example, you might want to know how the icons in the Operation Flow view relate to those in the Interaction Detail view.
- Use the **Component view/Operation View** list to switch between component view and operation view of BPM elements in the topology.
- Click the  **Configure Unavailability** icon in the Group Summary view or the Operation Flow for Service Group view to display the Configure Unavailability dialog for associating predefined situations with unavailability. See “Configuring for unavailability” on page 180 for more information.

Tivoli Enterprise Portal provides additional basic topology graph functions. For example, zoom state, panning control with scroll bars, fit to view, overview, and the ability to display the topology in table mode. You can also use the **Forward** and **Backward** navigation arrows to move through a series of workspaces, topology views, and tables.

Viewing details in a flyover window

You can hold the cursor over an icon or link in the topology to display a flyover window with detailed information about the object.

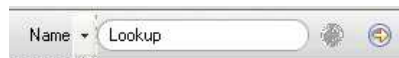
For icons, the flyover window contains object details. See “Viewing details” on page 110.

For links, the flyover window contains call metrics. “Viewing metrics” on page 107.

If you select the **Informational** icon in the Tivoli Enterprise Portal topology toolbar, the flyover window details are amended with more information displayed in blue text. Click the **Informational** icon again to revert back to the original hover help display.

Searching views

Use the search tool



in the toolbar to search for node labels and 16 different attributes from the ITCAM for SOA service model within any view.

Alongside the search field in the toolbar is a **Name** label that indicates the name of the node label or specific attribute that is being searched. Click the down arrow to select from a defined list of available attributes in a particular view or type in your search text.

When you enter the text, the node matches are automatically selected, and the topology graph zooms in on that node. When multiple matches are found, you can use the **Find next** and **Find previous** buttons to navigate between the matches. When either button is active, hold the mouse cursor over the active button to see a flyover window that indicates the number of matches and the current position of the matching nodes. For example, *1 of 4*, where *1* is the current position in the attribute list of the matching nodes and *4* is the number of matches.

If your search text cannot be found, the background color of the entry field changes to yellow.

You can manually enter the asterisk (*) and question mark (?) wildcard characters to find specific node labels. For example, *sys** to find *system* and *sys_01*, or *system?* to find *system1* and *systemZ*. An asterisk (*) at the end is also added automatically, so you can enter *sys* to find *system*.

This search field is not available in table mode. Use the table sorting and filtering capabilities to locate rows that meet particular search criteria.

If you decide to return to a workspace using the Tivoli Enterprise Portal **Forward** and **Backward** navigation buttons, your attribute search is evaluated, but the nodes are not selected in the topology. Your matching attribute or attributes are highlighted in blue within the node tooltip information only. You can also click the entry field to restore the matching selections.

In the Operation Flow view (aggregate view), click the current find attribute down arrow to select from the following attributes:

- Name (searches all node labels and is the default selection)
- Service Name and Namespace (searches for all matches in the operation name, operation namespace, service port, and service port namespace attributes)
- Operation
- Operation Namespace
- Service Port
- Service Port Namespace
- Mediation Type
- Port Type
- Service Groups

In the Interaction Detail view (instance view), click the current find attribute down arrow to select from the following attributes:

- Name (searches all node labels and is the default selection)
- Service Name and Namespace (searches for all matches in the operation name, operation namespace, service port, and service port namespace attributes)
- Operation
- Operation Namespace
- Service Port
- Service Port Namespace
- Mediation Type
- Port Type
- Application Server
- Node Name

- Cell Name
- Application Server Environment
- Cluster Name
- DataPower Domain
- Computer System
- Computer System Address

These lists are displayed with a check mark next to the currently selected attribute.

Viewing topology in table mode

When you display the Operational Flow workspaces, you can select to display the topology data in the default topology view format, or you can switch to table format by selecting the **View as Table** icon in the menu bar of the Operation Flow view.

Use table mode to clarify your view if there are many operation aggregates or operation instances contained in the topology. You can narrow your scope by using the table column sorting and filtering capabilities provided in table mode.

In table mode, you can sort and filter on any of the various columns displayed in the table. For example, for the Instance Status column in the Operation Flow table view, aggregates that have at least one instance that is not in a *Normal* state are shown with one of the non-normal status values for operation instances, such as *Fatal*, *Critical*, *Minor*, and *Warning*. Aggregates with all instances in a *Normal* state show a blank entry in the Instance Status column.

You can choose to filter out any aggregates with instances in a particular state, or display only those operation aggregates with all instances in a *Normal* state, by doing these steps:

1. With the Operation Flow view in table mode, right-click the Instance Status column header and select **Filter -> Edit Filter**.
2. The Edit Filter dialog is displayed. Select one or more check boxes next to the operation instance states to display in the table, and click **OK**. To have the table view display only aggregates with *Normal* status, select the check box next to the blank entry, and click **OK**. The use of a blank entry to indicate that there is no abnormal status is consistent with typical Tivoli Enterprise Portal operation.

Automatically switching to table mode: If the number of objects in the view exceeds the maximum specified in the view properties, the topology is automatically displayed in table mode rather than in graphical mode. The default value for this threshold is 50. To configure this threshold for switching the view to table mode, see Viewing properties.

Operational Flow table view column headings

Table 23 describes the Operational Flow table view column headings.

Table 23. The names and descriptions of the column headings for the Operational Flow table view.

Column Name	Description
Resource Type	The resource type for an operation aggregate.
Name	The node label name of the operation aggregate.

Table 23. The names and descriptions of the column headings for the Operational Flow table view. (continued)

Column Name	Description
Operation	The name of the service operation.
Operation Namespace	The namespace used to fully qualify the operation.
Service Port	The name of the service port.
Service Port Namespace	The namespace used to fully qualify the service port.
Status	The indicator for whether this operation aggregate is associated with one or more operation instances that have status.
Highlighted	Provides a reason why the operation aggregate is highlighted.
Number of Instances	The number of instances represented by the operation aggregate.
Number of Abnormal Instances	The number of instances that have an abnormal status.
Service Groups	Displays the names of all service groups for which the operation aggregate is a member.
Source	The port and operation of the caller; it is used only in call relationships.
Target	The port and operation of the callee; it is used only in call relationships.
Server Enter Message Count	The number of messages seen at the Server Enter interception point; it is used only in call relationships.
Server Leave Average Response Time	The average response time seen at the Server Leave interception point; it is used only in call relationships.

Interaction Detail table view column headings

Table 24 describes the Interaction Detail table view column headings.

Table 24. The names and descriptions of the column headings for the Interaction Detail table view.

Column	Description
Resource Type	The resource type for an operation instance.
Name	The node label name of the operation instance.
Operation	The name of the operation instance.
Operation Namespace	The namespace used to fully qualify the operation.
Service Port	The name of the service port.
Service Port Namespace	The namespace used to fully qualify the service port.
Port Type	The type of service that is being monitored.
DataPower Domain	The DataPower domain name.

Table 24. The names and descriptions of the column headings for the Interaction Detail table view. (continued)

Column	Description
Status	The indicator for whether this operation instance is associated with one or more operation aggregates that have a status. The status types are: <ul style="list-style-type: none"> • Normal • Information • Warning • Critical
Highlighted	Provides a reason why the operation aggregate is highlighted.
Application Server Environment	The type of application server environment.
Application Server	The name of the application server.
Computer System	The hostname for the computer system on which the application server runs.
Computer System Address	The IP address for the computer system on which the application server runs.
Node Name	The name of the node to which the application server belongs.
Cell Name	The name of the cell to which the application server belongs.
Cluster Name	The name of the cluster to which the application server belongs.
Source	The port and operation of the caller; it is used only in call relationships.
Target	The port and operation of the callee; it is used only in call relationships.
Server Enter Message Count	The number of messages seen at the Server Enter interception point; it is used only in call relationships.
Server Leave Average Response Time	The average response time seen at the Server Leave interception point; it is used only in call relationships.

Using the status area

The status area for the service-to-service topology is at the bottom of the workspace.

Table 25 on page 107 describes the Status area.

Table 25. The description of the Status area in the service-to-service topology workspace.

Element	Description
Total: Selected:	The overall count of the operation instances that are associated with the Interaction Detail view of the service-to-service topology. These numbers are meaningful because the Operation Flow view is strictly for navigation. For example, Total: 4 Selected:1.
Timeframe	The time interval reflected in the data that is displayed in the view. The time interval is displayed in the local time at the Tivoli Enterprise Portal.
Last Refreshed:	The time of the last refresh. The time interval is displayed in the local time at the Tivoli Enterprise Portal.
Status Current as of:	The time stamp for the last time the SOA Domain Management Server polled for the current status. The time stamp is displayed in the local time at the Tivoli Enterprise Portal.
Message area	The warning or error messages that are associated with the SOA Domain Management Server responses to the service-to-service topology. For example, KD4UI0002I The view is empty.

Viewing detailed information from Operational Flow workspaces

In each of the Operational Flow workspaces, you can access detailed information about nodes and links.

Viewing metrics

The operation aggregate nodes in the service-to-service topology graphs and all their call relationships have flyover windows that display the attributes for the nodes, data about situation events, and any additional metrics for the call relationship paths.

This information is displayed through the flyover window for any call relationship that is displayed. You can view metrics in more detail in a tabbed notebook window, with a default bar chart view and a table mode.

There is a large set of metrics that is associated with every call relationship at both the operation instance and aggregate level. Metrics are collected at these interception points for the *provider* and *requester*:

- Server Enter
- Server Leave
- Client Response

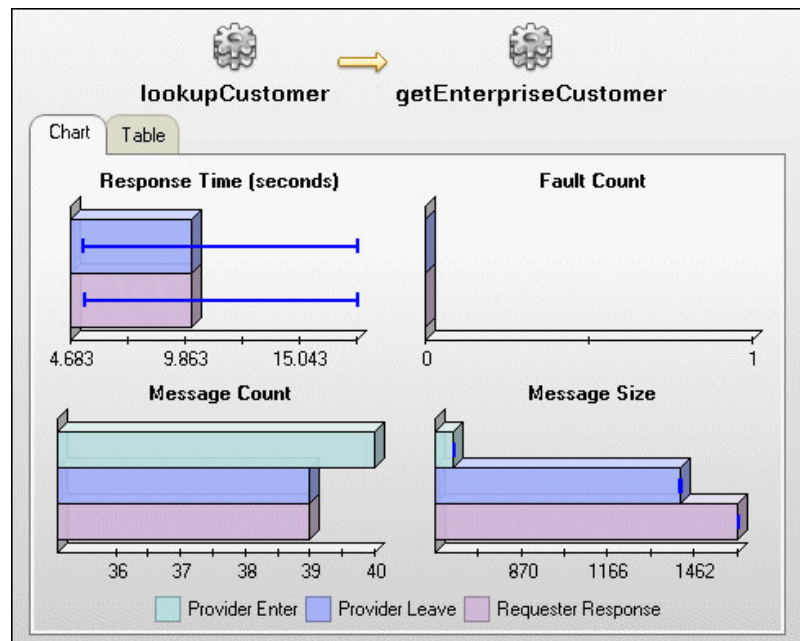
The metrics are organized into metric type and interception point with regard to the provider and the requester. You can use this organization to compare the following key call relationships metrics:

- Message Count
 - Server enter compared to Server Leave
 - Server Leave compared to Client Response
- Message Length minimum, average, and maximum
 - Server enter compared to Server Leave
 - Server Leave compared to Client Response
- Fault Count
 - Server Leave compared to Client Response
- Response Time minimum, average, and maximum
 - Server Leave compared to Client Response

To view metrics in more detail, select a call relationship in the topology graph or from the Resource Type column in the table mode of the Operation Flow view and click **Show Metrics** from the menu. The Metrics notebook is displayed with the **Chart** tab as the default view.

The **Chart** tab displays the call relationship metric data in the following bar charts:

- Response Time (seconds)
- Fault Count
- Message Count
- Message Size



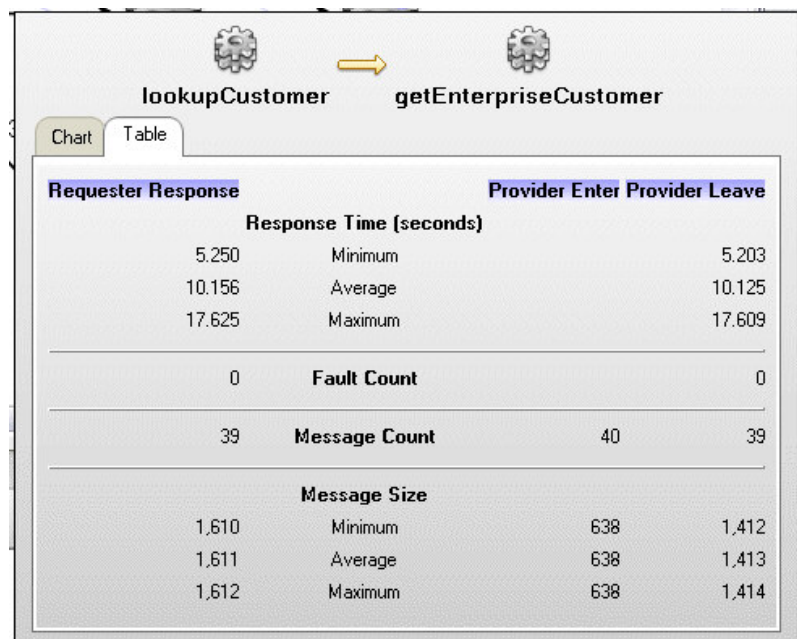
The superimposed blue range indicators within the Response Time bar chart display the average value. The minimum value shows the lowest possible value; the maximum value shows the highest possible value.

The color legend at the bottom of the **Chart** tab displays the colors that are used in the bar charts for different values. The bar chart values are:

- Provider Enter
- Provider Leave
- Requester Response

The **Table** tab displays the call relationship metric data in columns and grouped for each of the following metrics:

- Response Time (seconds)
- Fault Count
- Message Count
- Message Size



Requester Response		Provider Enter	Provider Leave
Response Time (seconds)			
5.250	Minimum		5.203
10.156	Average		10.125
17.625	Maximum		17.609
<hr/>			
0	Fault Count		0
<hr/>			
39	Message Count	40	39
<hr/>			
Message Size			
1,610	Minimum	638	1,412
1,611	Average	638	1,413
1,612	Maximum	638	1,414

The minimum, average, and maximum values for each metric are displayed as appropriate. You can switch between the **Chart** tab and the **Table** tab to view the metric data in either format.

Some call relationship patterns (where the operation being called is unmanaged) might not have values (or might have a value of -1) for some of the metrics during the last monitoring interval. Absent values are indicated with empty cells in the **Table** tab view to differentiate them from metrics that have a value of 0 or are listed as **No Value** in place of a bar in the bar charts.

You can change the default view in the Metrics notebook by clearing the **Display bar chart as the default in the tooltip for call relationship data** check box in the **Configuration** tab of the Properties notebook.

The metrics that are displayed in the service-to-service topology and the Metrics notebook represent the best-available measurement information considering how the data is collected and aggregated across multiple levels; for example, the data collectors, Tivoli Enterprise Monitoring Agents, and the SOA Domain Management Server. These metrics are useful to characterize the interaction among the

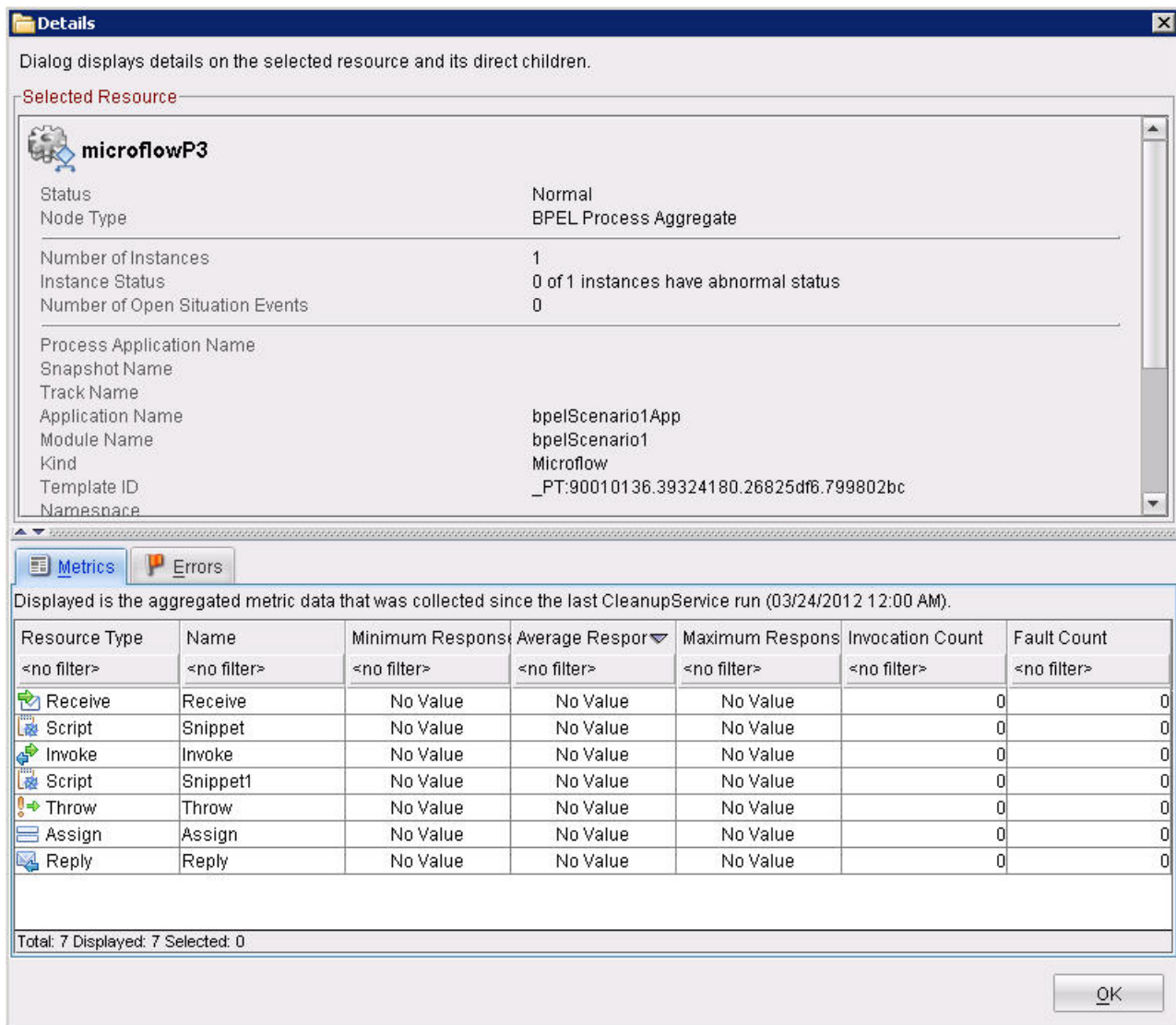
operations in a call relationship and to locate specific focal areas in your SOA. Once you find an area of interest, within the service-to-service topology, you can use links to access other workspaces, for example, the “Performance Summary workspace” on page 41.

Important: The current version of ITCAM for SOA displays call metrics for calls from an SCA component to a Business Process Definition (BPD), but does not display metrics for any other calls from or to a BPD.

Viewing details


To view detailed information about an aggregate or instance node, you can hold the mouse cursor over the icon. A flyover window is displayed.

To view the details in a window, and to view additional details on child resources and errors, right-click the icon (in the table view, right-click the row) and select the **Show Details** menu item. The **Details** window opens.



Dialog displays details on the selected resource and its direct children.

Selected Resource

 **microflowP3**








Status: Normal
Node Type: BPEL Process Aggregate

Number of Instances: 1
Instance Status: 0 of 1 instances have abnormal status
Number of Open Situation Events: 0

Process Application Name:
Snapshot Name:
Track Name:
Application Name: bpelScenario1App
Module Name: bpelScenario1
Kind: Microflow
Template ID: _PT:90010136.39324180.26825df6.799802bc
Namespace:

Metrics **Errors**

Displayed is the aggregated metric data that was collected since the last CleanupService run (03/24/2012 12:00 AM).

Resource Type	Name	Minimum Respons	Average Respor	Maximum Respons	Invocation Count	Fault Count
<no filter>	<no filter>	<no filter>	<no filter>	<no filter>	<no filter>	<no filter>
 Receive	Receive	No Value	No Value	No Value	0	0
 Script	Snippet	No Value	No Value	No Value	0	0
 Invoke	Invoke	No Value	No Value	No Value	0	0
 Script	Snippet1	No Value	No Value	No Value	0	0
 Throw	Throw	No Value	No Value	No Value	0	0
 Assign	Assign	No Value	No Value	No Value	0	0
 Reply	Reply	No Value	No Value	No Value	0	0

Total: 7 Displayed: 7 Selected: 0

OK

Figure 24. The **Details** window.

Summary information

In the **Selected resource** part of the window, you can view the name of the resource (that is, instance or aggregate). For BPM Human Tasks and BPEL processes, click the name to open the BPC Explorer browser window. (If the window does not open, you must set the BPC Explorer URL; see the ITCAM for SOA Installation Guide).

You can also examine the details on the resource. The same details are available in the flyover window.

Depending on the resource type, different information is available. The following attributes might be present for an operation aggregate:

Status The overall status for the aggregate. See “Operation status at the aggregate level” on page 76.

Node type

The type of the node (operation aggregate).

Number of Instances

The number of instances that are part of the aggregate.

Instance Status

This indicator is based on the status of the individual instances within the aggregate. If one or more instances have an abnormal status, this fact is represented by a phrase, such as, “3 of 10 instances have abnormal status”.

The status that is displayed in the service-to-service topology and the Details window might be delayed as opposed to the instantaneous status you see in the Navigator Physical view of the Tivoli Enterprise Portal. Some situations are inherently not applicable to the instances displayed in the service-to-service topology; it is possible you might view an abnormal *D4*: subnode, but the associated instances might be displayed as normal.

Operation

The name of the operation.

Operation Namespace

An additional identifier that, when paired with the operation name, defines a fully qualified and unique identifier for the operation.

Service Port

The name of the service port.

Service Port Namespace

An additional identifier that, when paired with the service port name, defines a fully qualified and unique identifier for the service port.

Port Type

The type of service port, for example a Web Services Description Language port.

Groups

Identifies one or more service or process groups to which the aggregate belongs.

The following attributes might be present for an operation instance:

Status The status is listed as either *Normal* or one of the abnormal status levels (such as *Critical*, *Fatal*, or *Warning*). If the status is not *Normal*, a list of situation events that contribute to the abnormal status is displayed in the

flyover window below the other attributes. This list of events is similar in appearance to the list observed in the Navigator Physical view when you hold the mouse cursor over a node that has a status indicator, but is limited to the specific situation events that relate to that node.

The list of situations is grouped by status severity, for example, *Fatal*, *Critical*, *Minor*, *Warning*, *Harmless*, and *Informational* or *Unknown*. Click the link indicator next to the situation name to open the Situation Event Results workspace. The Initial Situation Values table view is in the Situation Event Results workspace, the values of the attributes when the situation first occurred.

Important: The situation event time is the local time that the situation was opened at the hub of the Tivoli Enterprise Monitoring Server.

Node type

The type of the node (operation instance).

Operation

The name of the operation.

Operation Namespace

An additional identifier that, when paired with the operation name, defines a fully qualified and unique identifier for the operation.

Service Port

The name of the service port.

Service Port Namespace

An additional identifier that, when paired with the service port name, defines a fully qualified and unique identifier for the service port.

Port Type

The type of service port, for example a Web Services Description Language port.

Application Server Environment

Application Server

The name of the application server. This field is not displayed for operations running on DataPower servers.

Computer System

Computer System Address

Node Name

Cell Name

Mediation type

This field is displayed if it is applicable to the operation.

DataPower Domain

This field is displayed if it is applicable to the operation.

Cluster name

This field is displayed if it is applicable to the operation.

For more information about situation events, see the *Situation event results workspace* section of the Tivoli Enterprise Portal online help.

The following information is displayed for a BPD aggregate or instance:

Status**Node type**

BPD Process Aggregate or BPD Operation Instance.

Number of instances

This field is displayed only for an aggregate.

Instance status

For an aggregate, this field shows the number of instances with abnormal status.

Number of open situation events**Process application name****Business process name****Namespace****Snapshot name**

The snapshot name of the process application or toolkit to which this BPD corresponds. If no snapshots exists, the value is empty.

Acronym name

This name identifies the project and its related library items.

Number started

The number of processes for this BPD that have been started in the selected time span. If you select real time, the selected time period is 5 minutes. For more information about selecting the time span, see “Selecting the time span for topology metrics” on page 120.

Number active

The number of processes for this BPD that are currently started and not yet completed. This field is displayed only if you select real time.

Average active

The average number of processes for this BPD that were started and not yet completed during the selected time period. This field is displayed only if you select a time span in the past.

Number completed

The number of processes for this BPD that have been completed in the selected time period.

Number failed

The number of processes for this BPD that have been completed with a failure result in the selected time period.

Min/Max/Avg Activities

The minimum, maximum, and average amount of activities within this BPD that were active at the same time within the selected time span.

Minimum Execution Time, Average Execution Time, Maximum Execution Time

The minimum, average, and maximum current execution time for BPDs that were active within the selected time span.

Minimum Completion Time, Average Completion Time, Maximum Completion Time

The minimum, average, and maximum total execution time for BPDs that have completed within the selected time span.

Minimum Failure Time, Average Failure Time, Maximum Failure Time

The minimum, average, and maximum time between the start and the triggering of failure for BPDs that have failed within the selected time span.

Frontend Service

Whether this BPD is a frontend service in the current process group. This field is displayed only if you are viewing the topology for a service group or process group. For more information about frontend services, see “Determining front-end services” on page 150.

For a BPD instance, application server information is also displayed:

Application Server Environment**Application Server****Computer System****Computer System Address****Node Name****Cell Name**

Important: The metrics for a BPD are aggregated only since the last run of the CleanupService. This service, by default, runs once every 24 hours.

The following information is displayed for an SCA (including BPEL process, Human task, and Mediation flow) component or operation aggregate or instance:

Status**Node type**

The type of SCA component or operation.

Number of instances

This field is displayed only for an aggregate.

Instance Status

This field is displayed only for an aggregate.

Number of open situation events

This field is displayed only for an aggregate.

Process application name

The name of the parent process application. This value is set if the application is installed via the Process Center. It is not set if the application is installed using WebSphere Integrated Solutions Console.

Snapshot name

The name of the snapshot of the parent process application. This value is set if the application is installed using the Process Center. It is not set if the application is installed using IBPM Integrated Solution Console.

Track name

The name of the track of the parent process application. This value is set if the application is installed using the Process Center. It is not set if the application is installed using IBPM Integrated Solution Console.

Application name

The name of the parent application.

Module name

The name of the parent module, which is a part of the parent application.

Component name

The name of the parent component, which is a part of the parent module. This name is only displayed in operation view, where the node represents an operation within a component.

Groups

Identifies one or more process groups to which the component or operation belongs.

For an SCA component instance or operation instance, application server information is also displayed:

Application Server Environment**Application Server****Computer System****Computer System Address****Node Name****Cell Name**

For several SCA component types, additional information is displayed.

The following additional information is displayed for a BPEL component or operation:

Kind Long-running or microflow.

Template ID**Namespace****Valid from****Monitored error data since**

The date and time when the data was reset by the CleanupService

Number of errors**Number of error types****Time of last error**

Blank if there were no errors since the information was reset.

The following additional information displays for a mediation flow component or operation:

Kind The kind of the mediation flow as modeled in the WebSphere Integration Designer.

Namespace

The following additional information displays for a Human Task component or operation:

Kind The kind of the Human Task as modeled in the WebSphere Integration Designer.

Template ID

Namespace**Valid from****JNDI Name Staff Plugin Provider**

The name of the staff plug-in provider that BPM uses to determine who can start or claim the Human Task that the node represents.

Workbasket Name

The name of the workbasket to which the Human Task that the node represents belongs. Workbaskets are an optional approach for assigning tasks to people.

For a Mediation subflow aggregate or instance node, the following information is displayed:

Status**Node type**

Mediation subflow.

Number of instances

For an aggregate only.

Subflow Name**Subflow Namespace**

For a Mediation subflow instance, application server information is also displayed:

Application Server Environment**Application Server****Computer System****Computer System Address****Node Name****Cell Name**

For a Database External System aggregate or instance node, the following information is displayed:

Status**Node type**

Database External System.

Number of instances

This field is displayed only for an aggregate.

JNDI Datasource

The JNDI name of the data source used to access the database from a mediation flow.

Table name

The table name within the database.

Search Column

The column name within the table.

Monitored error data since

The date and time when the data was reset by the CleanupService.

Number of errors

Number of error types**Time of last error**

Blank if there were no errors since the information was reset.

For a Registry External System aggregate or instance node, the following information is displayed:

Status**Node type**

Registry External System.

Number of instances

This field is displayed only for an aggregate.

Registry name**Monitored error data since**

The date and time when the data was reset by the CleanupService.

Number of errors**Number of error types****Time of last error**

Blank if there were no errors since the information was reset.

For an Adapter, the details window displays the node name and type, and information about the file or database that the adapter accesses.

Metrics and child resources

For mediation flows and subflows, BPEL business processes, and BPDs, in the lower part of the window, the **Metrics** tab displays the resources that are considered *children* of this resource. The definition of children depends on the resource type:

- For mediation flows and subflows, children are mediation flow primitives (some of them might call subflows).
- For BPEL business processes, children are BPEL activities. You can see all activities in a BPEL process in this table, irrespective of their place within the process hierarchy. The activity type is indicated by the same icons as in WebSphere Integration Developer. Structured activities are not included.
- For BPDs, children are activities and gateways.

For each of the child resources, the table displays metrics if they are available. The metrics are aggregated since the last run of the CleanupService. This service, by default, runs once every 24 hours.

To sort the data by any column, click the name of the column. You can also filter data in this table using standard Tivoli Enterprise Portal tools.

Important: Metrics from this table are not stored in the Tivoli Data Warehouse.

For mediation flows, subflows, and BPEL business processes, the table displays the following information.

Resource type

The type of the child resource. Activities and gateways can have different types.

Resource name

The name of the child resource.

Minimum response time

The smallest response time for a single call to this resource during the monitored time period.

Average response time

The average response time for all calls to this resource during the monitored time period.

Maximum response time

The largest response time for a single call to this resource during the monitored time period.

Invocation count

The number of calls to this resource during the monitored time period.

Fault count

The number of failed calls to this resource during the monitored time period.

For BPD components the table displays the following information.

Resource type

The type of the child resource (activity or gateway).

Name The name of the child resource.

Number Started

The number of activities or gateways of this type that were started during the monitored time period.

Currently Active

The number of activities or gateways of this type that are currently active.

Number Completed

The number of activities or gateways of this type that were completed during the monitored time period.

Minimum execution time

The minimum execution time for activities or gateways of this type that are currently active.

Average execution time

The average execution time for activities or gateways of this type that are currently active.

Maximum execution time

The maximum execution time for activities or gateways of this type that are currently active.

Minimum completion time

The minimum execution time for activities or gateways of this type that were completed in the monitored time period.

Average completion time

The average execution time for activities or gateways of this type that were completed in the monitored time period.

Maximum completion time

The maximum execution time for activities or gateways of this type that were completed in the monitored time period.

Important: In a cluster environment, a BPD activity might start on one instance (A), but complete on another instance (B). In this case, while it is active, it is counted as active on instance A; however, when it completes, it is counted as completed on instance B. It remains counted as started on instance A.

Errors

In the lower part of the window, the **Errors** tab is only available for BPEL components, BPEL operations, BPDs, and external systems (databases and registries).

It displays a list of all error types that occurred while the component was monitored. The error counters and last occurrence time stamp are displayed for errors that occurred since the last run of the CleanupService. This service, by default, runs once every 24 hours.

The error information is displayed as a table.

For BPEL components and BPEL operations, the table has the following columns:

Last Occurrence

The time stamp of the last occurrence of this error.

Error Count

The number of times this specific error occurred.

Fault Name

The name of the error.

Fault Namespace

The namespace of the error.

Error Message

The message of the error.

For BPDs, the table has the following columns:

Last Occurrence

The time stamp of the last occurrence of this error.

Count The number of times this specific error occurred.

Fault Name

The name of the error.

Fault Message

The message of the error.

For external systems, the table has the following columns:

Last Occurrence

The time stamp of the last occurrence of this error.

Error Count

The number of times this specific error occurred.

Error Message

The message of the error.

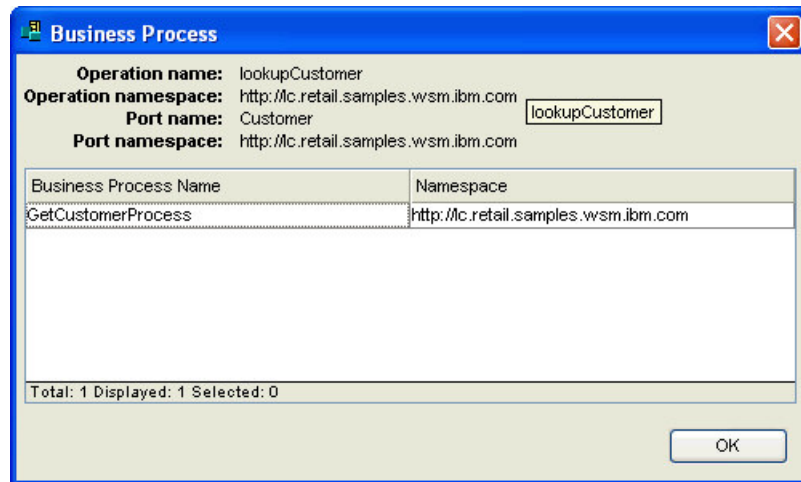
Viewing business processes

If you configured SOA Domain Management Server and Tivoli Common Object Repository components of ITCAM for SOA, and ran the Business Process Execution Language (BPEL) Discovery Library Adapter to process BPEL files that define business processes that are related to services in your environment, you can view a list of business processes that are related to a service port operation.

Use the Business Process window to display the business processes that are associated with an operation aggregate as determined by the configuration data in Tivoli Common Object Repository. This window is read-only.

To access the Business Process window, select an operation aggregate node in the Operation Flow view and from the menu click **Show Business Processes**. The Business Process window contains the following columns:

- Business Process Name
- Namespace




For example, you might have an operation that is named *lookupCustomer* and a business process that is associated with it is named *GetCustomerProcess*. The accompanying business process namespace might be, *http://lc.retail.samples.wsm.abc.com*.

If there are no business processes that are associated with the selected operation aggregate, a message is displayed with the KD4UI0015E error message.

Settings in Operational Flow workspaces

In each of the Operational Flow workspaces, you can modify several settings.

Selecting the time span for topology metrics

To specify the time frame for which summarized metrics are displayed in the operational flow views, click the  **Time Span** icon in the toolbar. The **Time Span** window is displayed.

Changes to the time frame affect the metrics that are displayed on the links and the time frame that is displayed on the status area. Changing the time frame also causes the view to be refreshed. The displayed status is always for the current time

frame, not the time frame that is specified for historical data.

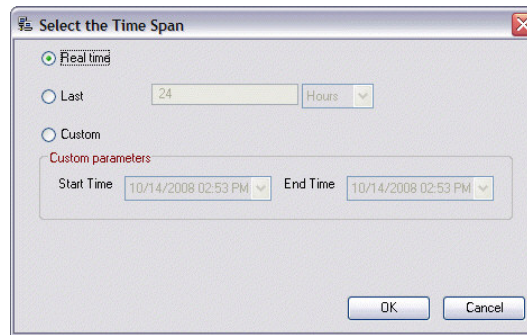
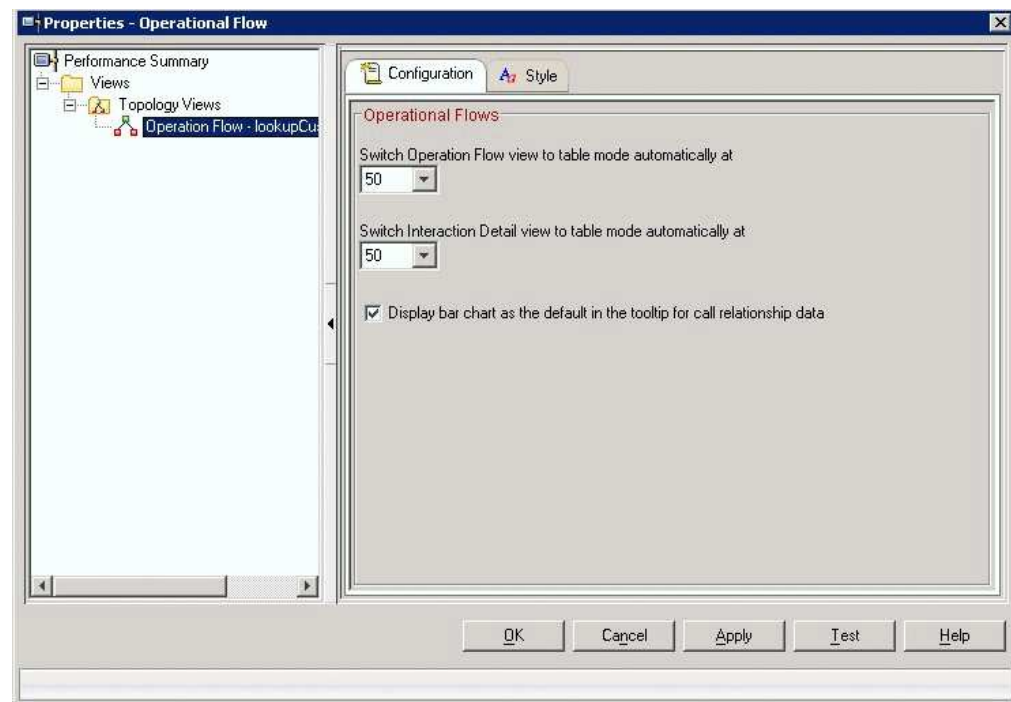


Figure 25. The Time Span Dialog

Viewing and setting properties

Use the Properties notebook for all of the Operational Flow workspaces to specify threshold values for the Operation Flow and Interaction Detail topology views or to change the appearance of your view.



Sometimes the number of objects in the topology views can make it difficult to navigate and pan around a specific view; it might be preferable to switch the view to table mode. The threshold values that you specify in the **Configuration** tab of this notebook determine when the view automatically switches from topology to table mode. The default threshold node setting is 50.

You can access this window using any of the following controls:

- The **Properties** button on the Tivoli Enterprise Portal toolbar.
- The **Edit ->Properties** menu selection on the Tivoli Enterprise Portal menu bar.

- The menu in the service-to-service topology view. Click anywhere in a view and select **Properties**.

The **Configuration** tab of the Properties notebook contains the following fields within the **Operational Flows** area:

- **Switch Operation Flow view to table mode automatically at**

This list contains the following node threshold values for the topology graph:

- Never
- 50
- 100
- 200
- Always

- **Switch Interaction Detail view to table mode automatically at**

This list contains the following node threshold values for the topology graph:

- Never
- 50
- 100
- 200
- Always

The **Display bar chart as the default in the tooltip for call relationship data** check box is selected to signify that the default page is the **Chart** tab in the Metrics notebook. Clear this check box to change the default setting.

The properties are stored on a per user basis for each workspace. If you change these properties, you change the workspace and are prompted to save a copy of the new workspace.

Click **OK** to save your new values and close the notebook.

Click **Apply** to apply your changes; the notebook remains open.

Click **Test** to preview your new values in the view.

Use the **Style** tab to change the color and other formatting elements of your view. For more information, see the online help for Tivoli Enterprise Portal.

Maintenance actions for Operational Flow workspaces

Sometimes, operations are removed from your environment or the IP addresses of servers hosting them change. In these cases, you might need to perform maintenance actions to ensure that information in Operational Flow workspaces remains up-to-date.

Deleting an operation instance from the service-to-service topology views

If an operation instance is no longer deployed or monitored, you can delete it from the Operational Flow workspaces.

However, when you delete an operation instance from the Operational Flow workspaces, you are deleting it only from the data that is used to populate the service-to-service topology views. The operation is not deleted from the data that is used to display the Performance Summary, Message Summary, and Faults

Summary workspaces. It is also not deleted from the data that is used to populate the workspaces for service registry integration (also called *static*) topology views in the Services Management workspace.

If the ITCAM for SOA data collector detects an operation instance after it was deleted from the Operational Flow workspace, it is displayed again in the service-to-service topology views the next time that the views are updated.

Important: SCA components and some related nodes are displayed based on static data (see “SCA component display” on page 85), ITCAM for SOA removes these nodes automatically as soon as the application is uninstalled on the server. You do not need to delete these nodes manually.

Deleting a BPD node from the service-to-service topology views

If a BPD instance is no longer used, you can delete it from the Operational Flow workspaces. To delete the node, select it in the Interaction Detail view, right-click it, and click **Delete**.

However, if the same BPD is run again, you must ensure that the Cleanup Service runs after the node is deleted and before the BPD is started. Otherwise, the BPD will not be displayed in the topology.

By default, the Cleanup Service runs every 24 hours.

Deleting unmanaged objects

The SOA Domain Management Server function of ITCAM for SOA provides several command-line interface (CLI) commands that you can use to delete unmanaged subnodes, clients, and operation instances from its database. The database contains the data that is displayed in the service-to-service topology views. These CLI commands have no parameters and are available as scripts that are in these directories on the computer system where your Tivoli Enterprise Portal Server is located:

- For Windows operating systems: `<ITM_Home>\CNPS\Products\KD4\bin`
- For Linux and AIX operating systems: `<ITM_Home>/<platform>/cq/Products/bin`

For information about these directory paths, see “Operating system-dependent variables and paths” on page xi).

Run the CLI commands when no activity is planned for the service-to-service topology workspaces. Many internal tables are updated by the CLI commands, resulting in longer times to process and display service-to-service topology workspaces and views.

Deleting unmanaged subnodes

To delete from the SOA Domain Management Server database operation instances, relationships, and metrics for data collector subnodes that are no longer being monitored by ITCAM for SOA, run the `deleteUnmanagedSubnodes` script.

Although you can remove an offline subnode from the Navigator Physical view and then run the `deleteUnmanagedSubnodes` script, this script does not stop ITCAM for SOA from monitoring the subnode if both the monitoring agent and the data collector are still deployed. Therefore, if the monitoring agent is started again, it is

possible for the subnode to be displayed again in the Navigator Physical view and for its operation instances to be displayed again in the service-to-service topology views.

If you have an application server environment that you want to stop monitoring, complete the following steps:

1. Disable the data collector that is monitoring the application server and uninstall the application server, if necessary.

If you want to stop monitoring a DataPower display group but you want to continue using the ITCAM for SOA data collector to monitor other DataPower appliances or display groups, disable data collection for that display group only.

For details about how to disable a data collector, see the *IBM Tivoli Composite Application Manager for SOA Installation Guide*.

2. If multiple application servers are on the computer system and you want to continue monitoring the other application servers, run the DeleteSubnode Take Action command to have the ITCAM for SOA monitoring agent remove the operation and relationship information for the subnode that is no longer being monitored. This Take Action command also causes the data collector subnode to change to the offline state. For details about this command, refer to “DeleteSubnode Take Action command” on page 236.

Otherwise, if no other application servers are monitored on the computer system, uninstall the ITCAM for SOA monitoring agent on that computer system. For details, see the *Installation Guide*.

3. On the Tivoli Enterprise Portal, right-click the offline data collector subnode that you are no longer monitoring, and select **Clear offline entry**.
4. From a command prompt, navigate to the directory where the CLI command is located for your operating system.
5. Run this command:
 - For Windows operating systems:
`deleteUnmanagedSubnodes.bat`
 - For Linux and AIX operating systems:
`deleteUnmanagedSubnodes.sh`
6. If you are using the ITCAM for SOA Discovery Library Adapter to discover service data for the static topology views, run the Discovery Library Adapter with the **refresh** option. This action creates a Discovery Library Adapter book that does not contain information about the service ports and operations associated with the deleted data collector subnode. After loading the new refreshed book into the Tivoli Common Object Repository using the bulk load program, the static topology views do not show service ports and operations for the deleted data collector node.

For each subnode that is not included in the list of managed systems in the Navigator Physical view, the operation instances, relationships, and metrics for the subnode are deleted from the SOA Domain Management Server database. The operation aggregate instance for deleted operation instances is automatically deleted if there are no other instances associated with the aggregate. Unmanaged clients and unmanaged operations that call and are called by a deleted managed operation are also removed. If the subnode being deleted represents a DataPower display group, the DataPower domains in the display group are checked to see if they are being monitored by another subnode. If a domain is being monitored by another subnode, its operation instances are not deleted.

Deleting unmanaged clients and unmanaged operation instances

You might have unmanaged clients and unmanaged operations in the service-to-service topology views that were created before the ITCAM for SOA monitoring agent was deployed, or that were created as the result of faults. For more information about unmanaged clients and operations, see “Considerations for unmanaged clients and operations.” To remove these unmanaged clients and operations from the service-to-service topology views and from the SOA Domain Management Server database, run the `deleteUnmanagedClientAndOperations` script.

To delete unmanaged clients and operation instances, complete these steps:

1. From a command prompt, navigate to the directory where the CLI command is located for your operating system.
2. Run this command:
 - For Windows operating systems:
`deleteUnmanagedClientAndOperations.bat`
 - For Linux and UNIX operating systems:
`deleteUnmanagedClientAndOperations.sh`

Considerations for unmanaged clients and operations

When you deploy the ITCAM for SOA monitoring agent to multiple computer systems, you might see unmanaged clients or unmanaged operations displayed in the service-to-service topology views. This can occur if some of your application servers are sending services traffic before the ITCAM for SOA monitoring agent and data collector is installed and configured.

You might also see unmanaged clients and operations if you have a mix of monitoring agents that include agents from ITCAM for SOA version 6.1 or version 6.0 in your environment while services traffic is running. After you install and configure the ITCAM for SOA monitoring agent and data collector on all application servers that you plan to monitor, you can use the delete function of the Operational Flow workspaces to delete individual unmanaged client and unmanaged operation instances. Alternatively, to remove all unmanaged clients and operations from the SOA Domain Management Server database where the data for service-to-service topology views is stored, run the `deleteUnmanagedClientAndOperations` script. For details about this command, see “Deleting unmanaged clients and unmanaged operation instances”).

After you run the `deleteUnmanagedClientAndOperations` script, any operations that are not monitored by ITCAM for SOA monitoring agents are displayed again in the service-to-service topology views as unmanaged clients or unmanaged operations if the following criteria are met:

- The operations send messages to another operation that is being monitored by ITCAM for SOA
- The operations are sent messages from another operation that is being monitored by ITCAM for SOA

When you deploy a new application and all operations in the application are being monitored by ITCAM for SOA, you might see unmanaged clients and unmanaged operations displayed in the service-to-service topology views. Within 5 - 10 minutes, the unmanaged clients and unmanaged operations become managed operations as the ITCAM for SOA monitoring agents and SOA Domain Management Server learn about the new operations and their call relationships.

Unmanaged operations might also be displayed in the service-to-service topology views when faults occur that prevent the message from reaching the target operation, for example, when the application server to where the target operation is deployed is stopped. These unmanaged operations can also be removed from the service-to-service topology views by using the delete operation instance function of the Operational Flow workspaces, or by running the `deleteUnmanagedClientAndOperations` script. However, the unmanaged operations are displayed in the service-to-service topology views again if the faults are detected again.

Updating IP addresses for operation instances

Operation instances might be displayed in Operational Flow workspaces for a monitored application server on a system where the static IP address or the DHCP IP address is changed as a result of a reconfiguration.

The flyover window for the operation instance includes IP address information. You can update this information in the SOA Domain Management Server database by running the `kd4UpdateIP` command-line interface (CLI) command script on your database server.

Obtaining the `kd4UpdateIP` scripts

The `kd4UpdateIP.bat` script updates IP addresses in the Managed System table of your SOA Domain Management Server database on Windows operating systems, and works with IBM DB2, Microsoft SQL Server 2005, Microsoft SQL Server 2008, and Oracle databases.

The `kd4UpdateIP.sh` script updates IP addresses in the Managed System table of your SOA Domain Management Server database on UNIX or Linux operating systems, and works with IBM DB2 and Oracle databases.

Depending on where your SOA Domain Management Server database is located, you can obtain the `kd4UpdateIP` scripts in several ways:

- If your SOA Domain Management Server database is local to the Tivoli Enterprise Portal Server, within your file system, navigate to the `<TEPS_Home>/Products/KD4/bin` directory and find the `kd4UpdateIP.bat` (for Windows operating systems). For Linux or UNIX operating systems, use the `kd4UpdateIP.sh` script. The directory path, `<TEPS_Home>` is where the Tivoli Enterprise Portal Server is installed:
 - For Windows systems: `<ITM_Home>\CNPS`
 - For Linux and AIX systems: `<ITM_Home>/<platform>/cq`
- If your SOA Domain Management Server database was created on a database server computer different from where the Tivoli Enterprise Portal Server is located, you or your database administrator might have already copied the `kd4RemoteDB.zip` (for Windows systems) or `kd4RemoteDB.tar.gz` (for UNIX or Linux systems) compressed file from the `<TEPS_Home>/Products/KD4/latest/db` directory on your Tivoli Enterprise Portal Server computer to an available directory on your remote SOA Domain Management Server computer. When these compressed files are unpacked, the `kd4UpdateIP` scripts are available, along with other database creation scripts.

If these files are not already copied to your database server, copy them now.

Running the `kd4UpdateIP` script

To run the `kd4UpdateIP` command on your database server, complete these steps:

- For supported Windows operating systems:

1. Log on to your database server with a user authorized to read and write to the SOA Domain Management Server database.
2. Open a command prompt. If you are using DB2 for the SOA Domain Management Server database, you must open a DB2 command-line prompt.
3. Navigate to the directory where the `kd4UpdateIP.bat` script is located.
4. Run the `kd4UpdateIP.bat` command, using this syntax:

```
kd4UpdateIP.bat -dbtype {DB2 | Oracle | MSSQL2005 | MSSQL2008} [-dbname <db>]
[-dbinst <mssql>] {-hostname <hostname> | -oldip <old_ip>} -newip
<new_ip> -sdmspw <sdms_pw>
```

The following parameters are specified for this command:

-dbtype

This is a required parameter, specifying the database type. Valid values are *DB2*, *Oracle*, *MSSQL2005*, or *MSSQL2008*.

-dbname

This is an optional parameter, specifying the name of the SOA Domain Management Server. If this parameter is not specified, the default name of *KD4SDMS* is used. If **-dbtype** is set to *Oracle*, the value of **-dbname** is the Oracle System Identifier of the Oracle database.

-dbinst

This is an optional parameter, specifying the database instance if the value of the **-dbtype** parameter is *MSSQL2005* or *MSSQL2008*.

-hostname

This is an optional parameter, specifying the host name of the computer system whose IP address has changed. If this parameter is not specified, you must identify the computer system using the **-old_ip** parameter.

-old_ip

This is an optional parameter, specifying the old IP address of the computer system whose IP address has changed. If this parameter is not specified, you must identify the computer system using the **-hostname** parameter.

-new_ip

This is a required parameter, specifying the new IP address of the computer system whose IP address has changed.

sdmspw

This is a required parameter if **-dbtype** is set to *Oracle*, specifying the password for the SOA Domain Management Server user account.

- For supported Linux and AIX operating systems:

1. Log on to your database server with a user that is authorized to read and write to the SOA Domain Management Server database.
2. Open a command prompt and source the DB2 profile.
3. Navigate to the directory where the `kd4UpdateIP.sh` script is located.
4. Run the `kd4UpdateIP.sh` command, using this syntax:

```
kd4UpdateIP.sh -dbtype {DB2 | Oracle} {-hostname <hostname> | -oldip <old_ip>}
-newip <new_ip> -sdmspw <sdms_pw>
```

The following parameters are specified for this command:

-dbtype

This is a required parameter, specifying the database type. Valid values are *DB2* or *Oracle*.

-dbname

This is an optional parameter, specifying the name of the SOA Domain Management Server. If this parameter is not specified, the default name of *KD4SDMS* is used. If **dbtype** is set to Oracle, the value of **-dbname** is the Oracle System Identifier of the Oracle database.

-hostname

This is an optional parameter, specifying the host name of the computer system whose IP address has changed. If this parameter is not specified, you must identify the computer system using the **-old_ip** parameter.

-old_ip

This is an optional parameter, specifying the old IP address of the computer system whose IP address has changed. If this parameter is not specified, you must identify the computer system using the **-hostname** parameter.

-new_ip

This is a required parameter, specifying the new IP address of the computer system whose IP address has changed.

sdmspw

This is a required parameter if **-dbtype** is set to Oracle, specifying the password for the SOA Domain Management Server user account.

After you run this command, the updated IP address is displayed in the flyover window for the operation instance in the Operational Flow workspaces.

Service-to-service topology for DataPower

Several considerations exist for displaying service-to-service topology data for the DataPower environment.

DataPower firmware levels and configuration

If you did not configure your DataPower appliance and its Web Services Proxies and Multi-Protocol Gateways in the manner described in the *IBM Tivoli Composite Application Manager for SOA Installation Guide*, you might see topology views in which the DataPower mediations are not displayed, operational flows are disjointed, or both. This happens because an unmanaged client and unmanaged operation are shown in place of a managed DataPower mediation.

If you configured monitoring of non-SOAP operations flowing through your DataPower appliance, you see those operations in the Operational Flow workspaces as DataPower mediation operations; however, they have a relationship to an unmanaged client and an unmanaged operation, and are disjointed from your other operational flows.

Topology views for multiple domains

In the Operational Flow workspace, an operation instance is displayed for each DataPower appliance and domain where the operation is observed. For example, suppose a DataPower appliance with host name *dp-server1* has two domains, one called *Production* and the other called *Test*. If message traffic for service port *Customer* and operation *getCustomer* is sent through both domains, the

service-to-service topology views display two operation instances for service port *Customer* and operation *getCustomer*, one for the *Production* domain, and another for the *Test* domain.

This view differs from how the Performance Summary, Message Summary, and Faults Summary workspaces might display aggregated information about these same operations. Depending on how you configured your ITCAM for SOA data collector, those workspaces show the set of operations that are observed on a single DataPower appliance, or for the set of domains mapped to a display group.

Continuing with this example, suppose that you did not configure a display group to monitor a subset of the domains on the DataPower appliance *dp-server1*. This causes a single node for *dp-server1* to be displayed in the Navigator Physical view. Selecting the Performance Summary, Message Summary, or Faults Summary workspace, metric data for service port *Customer* and operation *getCustomer* is displayed, showing the aggregated metrics for this service port and operation from both the *Production* domain and the *Test* domain.

Continuing further, suppose that you select the Performance Summary workspace under the *dp-server1* node in the Navigator Physical view, and locate the row in the Services Inventory table for service port *Customer* and operation *getCustomer*. If you link to the Operational Flow for Operation workspace, the topology workspace displays two *getCustomer* operation instances for DataPower appliance *dp-server1* in the Interaction Detail portion of the Operation Flow view, one for each *getCustomer* operation in the two domains, *Production* and *Test*.

Topology views for a domain assigned to multiple display groups

You can configure data collection for the DataPower environment and assign the same DataPower domain to multiple display groups. In this case, an operation for that domain is displayed in the Performance Summary, Message Summary, and Faults Summary workspaces for multiple nodes in the Navigator Physical view. However, the Operational Flow workspaces still display only a single operation instance.

For example, suppose that the *Production* domain on DataPower appliance *dp-server1* is assigned to two different display groups, *AllProductionDomains*, and *AllDomains*. This causes two nodes to be displayed in the Navigator Physical view, one for each display group. Each of the two nodes has its own set of Performance Summary, Message Summary, and Faults Summary workspaces.

If you select the Performance Summary, Message Summary, or Faults Summary workspace that are under the *AllProductionDomains* node, the resulting workspace displays information about the *Customer* service port and the *getCustomer* operation. Selecting these workspaces under the *AllDomains* node also displays information about the *Customer* service port and the *getCustomer* operation.

Regardless of which Performance Summary subnode you select under either the *AllProductionDomains* or *AllDomains* node, when you link to the Operational Flow for Operation workspace for the *Customer* service port and *getCustomer* operation for the *Production* domain on *dp-server1*, only one operation instance is displayed in the topology views.

Suppose that you want to link from the Operational Flow for Operation workspace back to the Performance Summary workspace. If you select the *getCustomer*

instance operation for the *Production* domain on *dp-server1* in the Operational Flow for Operation workspace, and select the link to the Performance Summary workspace, you must definitely select from one of two possible Performance Summary workspaces, either the workspace under the AllProductionDomains node or the workspace under the AllDomains node.

Service-to-service topology for WebSphere Message Broker

This section describes how service-to-service topology is used in the WebSphere Message Broker environment.

IBM WebSphere Message Broker provides a *user exit* structure in which applications can provide message processing extensions. This user exit structure is the interception mechanism for the Data Collector for WebSphere Message Broker.

Mapping WebSphere Message Broker concepts to the service model

The WebSphere Message Broker environment differs from a typical web application server environment. WebSphere Message Broker uses *message flows* to handle request and response messages for the monitored services. A single message flow can be mapped into multiple services.

WebSphere Message Broker separates message flows into different operating system processes. These operating system processes are called *execution groups*. Each execution group can contain one or more message flows, and a message broker can include one or more execution groups.

Generally, WebSphere Message Broker components can be mapped into the ITCAM for SOA services model as shown in Table 26:

Table 26. Mapping WebSphere Message Broker concepts to the ITCAM for SOA services model

Concept in WebSphere Message Broker	ITCAM for SOA services model
Message Broker	Application server node (for example, <i>tivu02Node</i>)
Execution Group	Application server, such as <i>server1</i> or <i>server2</i> in WebSphere Application Server.
Message Flow	Service port (for enabling flows to be monitored)

The Data Collector for WebSphere Message Broker determines the service port name and namespace and the operation name and namespace from various information that is available to the user exit, depending on the transport method (HTTP, JMS, or WebSphere MQ) and the type of message (SOAP or binary). Table 27 on page 131 displays how message broker message elements are mapped to the service port and operation names and namespaces.

Table 27. Mapping WebSphere Message Broker message elements to Service Port and Operation

Message type	Service port namespace	Service port name	Operation namespace	Operation name
SOAP over HTTP	Extracted from WebSphere Message Broker HTTPRequest Node property <i>Web Service URL</i> . The portion of the URI after the <i>hostname:port</i> and before the question mark character (?) is used.	Message flow name	Namespace of first child element of <Body> in SOAP message	First child element of <Body> in the SOAP message. This is correct for RPC/literal. For doc/literal, it is the message name.
Non-SOAP over HTTP	Extracted from WebSphere Message Broker HTTPRequest Node property <i>Web Service URL</i> . The portion of the URI after the <i>hostname:port</i> and before the question mark character (?) is used.	Message flow name	MessageSetName: MessageTypeName. Use <i>UNKNOWN</i> for the field that is unavailable. Delimiter braces {} and () in the namespace are replaced with underscore characters (_).	The name of the input/output node
SOAP over WebSphere MQ	Use the value of the <user> <targetService> folder in the MQRFH2 header in the WebSphere MQ message if it is available. Otherwise, use WebSphere MQ Queue Manager name.	Message flow name	Namespace of first child element of <Body> in SOAP message	First child element of <Body> in SOAP message. This is the correct operation name for RPC/literal. For doc/literal, it is the message name.
Non-SOAP over WebSphere MQ	WebSphere MQ Queue Manager name	Message flow name	MessageSetName: MessageTypeName. Use <i>UNKNOWN</i> for the field that is unavailable. Delimiter braces {} and () in the namespace are replaced with underscore characters (_).	Use <i>NodeName: QueueName</i> . <i>QueueName</i> is the destination queue name for output nodes, or the source queue name for input nodes.
SOAP over JMS	JMS <i>targetService</i> property from the message if it is available. Otherwise, use the WebSphere MQ Queue Manager name.	Message flow name	Namespace of the first child element of <Body> in the SOAP message.	First child element of <Body> in the SOAP message. This is the correct operation name for RPC/literal. For doc/literal, it is the message name.
Non-SOAP over JMS	JMS <i>connectionFactory</i> name from the message	Message flow name	MessageSetName: MessageTypeName. Use <i>UNKNOWN</i> for the field that is unavailable. Delimiter braces {} and () in the namespace are replaced with underscore characters (_).	Use <i>NodeName: QueueName</i> . <i>QueueName</i> is the destination queue name for output nodes, or the source queue name for input nodes.

Table 27. Mapping WebSphere Message Broker message elements to Service Port and Operation (continued)

Message type	Service port namespace	Service port name	Operation namespace	Operation name
SOAP Messages through SOAP nodes	The node property, <i>targetNamespace</i> . It originally comes from the Web Services Description Language file used for the SOAP node.	The node property, <i>selectedPort</i> . It originally comes from the Web Services Description Language file used for the SOAP node	The node property, <i>targetNamespace</i> . It originally comes from the Web Services Description Language file used for the SOAP node.	For a SOAPRequest or SOAPAsyncRequest node, it is the node property, <i>selectedOperation</i> , which originally comes from the Web Services Description Language file used for the SOAP node. For a SOAPInput node, it is the first child element of <Body> in SOAP message.
Messages through Collector nodes	Queue Manager name	Message Flow name	Use the constant string, <i>CollectorNode</i>	The name of the collector node.

Monitoring data is displayed in the existing Tivoli Enterprise Portal workspaces and views similar to other data collectors. Message content logging is supported in the same manner as other data collectors, but message rejection is not supported because WebSphere Message Broker provides its own capabilities for rejecting messages.

Supported transport protocols

WebSphere Message Broker supports services that are started over HyperText Transfer Protocol (HTTP), Java Message Service (JMS), and Message Queue (MQ) transport protocols, and supports various message formats (SOAP messages and non-SOAP messages, and unstructured binary messages). For this version of the product, only the following subset of these transport protocol and message format combinations are supported:

- “SOAP messages over HTTP”
- “Non-SOAP messages over HTTP” on page 133
- “SOAP messages over JMS” on page 133
- “Non-SOAP messages over JMS” on page 134
- “SOAP messages over WebSphere MQ” on page 134
- “Non-SOAP messages over WebSphere MQ” on page 135

The HTTP Transport protocol

This section describes the support for the HyperText Transfer Protocol (HTTP) using SOAP and non-SOAP message formats.

For more information about limitations of the HTTP protocol, see “Limitations of the HTTP Transport protocol” on page 145.

SOAP messages over HTTP: SOAP messages over the HTTP transport protocol are supported by the HTTPInput, HTTPReply, and HTTPRequest nodes in WebSphere Message Broker. Using these nodes, HTTP message flows can act either as a web services provider or as a web services client.

- Using HTTP message flows as a web services provider:

Using the HTTP message flow as a web service provider, the HTTPInput node receives the request from the network, and the HTTPReply node returns the response. The HTTPInput node and HTTPReply node can be in different flows. Each HTTP request picked up by WebSphere Message Broker is uniquely identified, and this identifier is passed along with the message through the request and response message flow.

When both the request message flow instance and the response message flow instance are in same execution group, both request and response flows are monitored. When each message flow instance is in a different execution group, the requester identity cannot be maintained across two separate execution groups. Therefore, only the request flow is monitored.

The ITCAM for SOA data collector in the WebSphere Message Broker environment is activated when the message broker is started or reloaded. The data collector then collects metric information from the message broker and the message flows, and writes the information into the metric log file for later display in Tivoli Enterprise Portal workspaces and views.

- Using HTTP message flows as a web services client:

Using the HTTP message flow as a web service client, the message flow is used to call an existing web service. The HTTPRequest node sends the request to the existing service, waits for the response, and then passes it on the next node in the flow.

Non-SOAP messages over HTTP: The HTTP transport can also support non-SOAP message formats, such as general XML messages or traditional binary messages. These messages are handled the same way as SOAP messages over HTTP, in terms of correlating the request and response flows. When you configure the monitored message logging level to *Full* using the AddMntrCntrl_610 or UpdMntrCntrl_610 Take Action commands, the complete message content (including the header and body of the message) is logged using Base64 encoding.

The JMS Transport protocol

This section describes the support for the Java Message Service transport protocol using SOAP and non-SOAP message formats.

For additional information about limitations of the JMS protocol, see “Limitations of the JMS Transport protocol” on page 146.

SOAP messages over JMS: SOAP messages over the JMS transport are supported by the JMSInput, JMSOutput, and JMSReply nodes in WebSphere Message Broker. Using these nodes, JMS message flows can act as a web services provider.

- Using JMS message flows as a web services provider:

Using the JMS message flow as a web service provider, the JMSInput node receives the request from the network, and the JMSOutput or JMSReply node returns the response. The JMSInput node and JMSOutput or JMSReply node can be in different flows.

Each JMS request that is detected by WebSphere Message Broker is uniquely identified. This identifier is passed with the message through the request and response message flow.

When both the request message flow instance and the response message flow instance are in same execution group, both request and response flows are monitored. When each message flow instance is in a different execution group, the requester identity cannot be maintained across two separate execution groups. Therefore, only the request flow is monitored.

The ITCAM for SOA data collector in the WebSphere Message Broker environment is loaded when the message broker is started or reloaded. The data collector then collects metric information from the message broker and the message flows, and writes the information into the metric log file for later display in Tivoli Enterprise Portal workspaces and views.

For ITCAM for SOA version 7.2 Fix Pack 1, the SOAP over JMS protocol that is certified for use in WebSphere Application Server version 5.1 and version 6.0 (and later) is supported.

An implementation of the WebSphere version of SOAP over JMS can be monitored by ITCAM for SOA if it has these characteristics:

- The JMSMessageID field is used to uniquely identify a message.
- When you monitor JMS services by requester identity, the JMSXUserID field in the JMS transport header must contain the identity of the requester.

Important: For this release, only the user ID form of requester identity is supported. The IP address form of requester identity is not supported. To set the type of requester identity to monitor using the **SetReqIDTypeUserInfo** take action command, see “SetReqIDTypeUserInfo Take Action command” on page 248.

- When a client expects a response, the JMSReplyTo field is set as a queue where the responding message is stored. This message is identified as a two-way message invocation. If the JMSReplyTo field is not set, this message is considered a one-way request.
- In the response message flow, the JMSCorrelationID field is used to store the value of the JMSMessageID from the request message flow.
- The message content complies with the SOAP specification.

Non-SOAP messages over JMS: The JMS transport also supports non-SOAP message formats. These messages are handled in the same way as SOAP messages over JMS, in terms of correlating the request and response flows. When you configure the monitored message logging level to *Full* using the AddMntrCntrl_610 or UpdMntrCntrl_610 Take Action commands, the complete message content (including the header and body of the message) is logged using Base64 encoding.

The WebSphere MQ Transport protocol

This section describes the support for the Message Queue transport protocol using SOAP and non-SOAP message formats.

For more information about limitations of the WebSphere MQ transport protocol, see “Limitations of the WebSphere MQ Transport protocol” on page 146.

SOAP messages over WebSphere MQ: The ITCAM for SOA data collector monitors message flows in the WebSphere MQ transport that conform to the WebSphere MQ version 6.0 SOAP transport protocol. The MQInput, MQOutput, and MQReply nodes are used. Any implementation that satisfies these basic criteria can be monitored:

1. The MQMD.MsgId field is used to uniquely identify a request message flow.
2. When monitoring WebSphere MQ services by requester identity, in the MQMD message header, the UserIdentifier field must contain the identity of the requester.

Restriction: For this release, only the user ID form of requester identity is supported. The IP address form of requester identity is not supported. To set the type of requester identity to monitor using the **SetReqIDTypeUserInfo** take action command, see “SetReqIDTypeUserInfo Take Action command” on page 248.

3. When a reply is required, the MQMD.ReplyToQ and MQMD.ReplyToQMgr fields indicate the target for the reply message. MQMD.ReplyToQ specifies the queue name for the reply. Optionally, MQMD.ReplyToQMgr specifies the queue manager name for the reply. If the queue manager name is not specified, the queue manager name for the message broker is used.
4. The response message uses MQMD.CorrelId to contain the value of MQMD.MsgId from the corresponding request message flow.
5. The MQRFH2 message header follows the MQMD message header. The MQRFH2.usr.endpointURL field contains the Web Services Description Language (WSDL) URL for the service endpoint.
6. The message content is in jms_byte type format and conforms to the SOAP standard.

Non-SOAP messages over WebSphere MQ: The WebSphere MQ transport also supports non-SOAP message formats, such as general XML messages or traditional binary messages. These messages are handled the same way as SOAP messages over WebSphere MQ, in terms of correlating the request and response flows. When you configure the monitored message logging level to *Full* using the AddMntrCntrl_610 or UpdMntrCntrl_610 Take Action commands, the complete message content (including the header and body of the message) is logged using Base64 encoding.

The MQInput, MQGet, MQOutput and MQReply nodes are monitored. The correlation between request flow and reply flow is supported for these common protocols:

- Correlating by Correlation ID
- Correlating by Message ID

Correlating by these protocols conforms to these criteria:

- The MQMD.MsgId field is used to uniquely identify a request message.
- When the MQMD.ReplyToQ field and, optionally, the MQMD.ReplyToQMgr field, is specified in the MQMD message header, the response message is expected in the queue specified by the queue manager. If the client is on same queue manager, then the MQMD.ReplyToQMgr field is not needed. The response message is put in the queue manager for the message broker.
- In the response message, the value of the MQMD.MsgId field from the request message is stored in different fields, depending on the correlation protocol:
 - When correlating by correlation ID, the value is stored in the MQMD.CorrelId field.
 - When correlating by message ID, the value is stored in the MQMD.MsgID field.

SOAP node support

WebSphere Message Broker version 6.1.0.2 and later includes support for new SOAP nodes that enhance the support for SOAP over HTTP. This support uses SOAPInput, SOAPReply, SOAPRequest, SOAPAsyncRequest, and SOAPAsyncResponse nodes.

The SOAPInput and SOAPReply nodes are analogous to the HTTPInput and HTTPReply nodes, and are used in message flows that implement a web service.

The SOAPRequest node is analogous to the HTTPRequest node, and is used in a message flow to call a web service provider synchronously.

The SOAPAsyncRequest and SOAPAsyncResponse nodes are used to construct a message flow (or a pair of flows) which calls a web service asynchronously. Calling a web service asynchronously means that the SOAPAsyncRequest node sends a web service request. However, the request does not block the message flow by waiting for the associated web service response to be received. The web service response is received at the SOAPAsyncResponse node, which is in a separate message flow. SOAP nodes are supported only in the HTTP transport protocol in WebSphere Message Broker version 6.1.0.2 and later. The IBM Tivoli Composite Application Manager for SOA data collector for WebSphere Message Broker supports messages flows with SOAP nodes from both a web services client and a web services provider. For more information about the limitations of SOAP node support, see “Limitations of support for SOAP nodes” on page 148.

Collector node support

WebSphere Message Broker version 6.1.0.2 includes support for a new collector node. This node is used to create a message collection from one or more sources based on configurable criteria. The WebSphere Message Broker data collector considers a collector node as an input node that processes only one-way non-SOAP messages.

Message flow to topology mapping rules

The data collector for WebSphere Message Broker supports the following input nodes:

- HTTPInput
- JMSInput
- MQGet
- MQInput
- SOAPInput
- SOAPAsyncResponse
- Collector

Input nodes that receive request messages are referred to as *request input nodes*, and input nodes that receive response messages are referred to as *response input nodes*.

The data collector supports the following output nodes:

- HTTPReply
- HTTPRequest
- JMSOutput
- JMSReply
- MQOutput
- MQReply
- SOAPAsyncRequest
- SOAPReply
- SOAPRequest

Output nodes that send request messages are referred to as *request output nodes*.
Output nodes that send response messages are referred to as *response output nodes*.

Mapping rules for mediations

The following rules govern the mapping of message flows with mediations to the topology that is displayed in Tivoli Enterprise Portal Operation Flow workspaces and views:

- **Rule M1:** A message flow that starts with a request input node is represented by a mediation object in the topology view. If such a flow sends request messages to a backend server, this is referred to as a *request flow*.
- **Rule M2:** A message flow that starts with a response input node is not represented by a separate mediation object in the topology view, because this message flow and the corresponding request flow are displayed in the topology view as a single mediation. This type of flow is referred to as a *response flow*.
- **Rule M3:** A collector node is represented by a mediation object in the topology view.
- **Rule M4:** A message flow that starts with multiple request input nodes is displayed in the topology view with multiple mediation objects.
- **Rule M5:** A message flow that starts with an unsupported input node is not displayed in the topology view as a mediation object. However, a supported request output node in such a message flow is displayed as a separate mediation object if the request output node sends two-way messages to a backend server, or sends a one-way SOAP message to a managed backend server.

Mapping rules for relationships

The following rules govern the mapping of message broker mediation relationships to the topology that is displayed in Tivoli Enterprise Portal Operation Flow workspaces and views.

For SOAP messages:

- **Rule R1:** If a request message flow receives an inbound SOAP message, an inbound relationship to the message broker mediation object is displayed in the topology view. If the service requester is not managed, an unmanaged object is displayed to the left of the mediation object in the topology view.
- **Rule R2:** If a request message flow sends a two-way outbound SOAP message, an outbound relationship from the message broker mediation to the service provider object is displayed in the topology view. If the service provider is not managed, an unmanaged object is displayed to the right of the mediation object in the topology view.
- **Rule R3:** If a request message flow sends a one-way outbound SOAP message to a managed server, an outbound relationship from the message broker mediation object to the managed server object is displayed in the topology view.
- **Rule R4:** If a request message flow sends a one-way outbound SOAP message to an unmanaged server, no relationship or server node is displayed in the topology view.

For non-SOAP messages:

- **Rule R5:** If a request message flow receives an inbound non-SOAP message, an unmanaged client object is displayed to the left of the mediation object in the topology view. The operation name and the service port name of the unmanaged client are the same as the names for the request input mediation object. An unmanaged object is displayed to the left of the mediation object in the topology view.

- **Rule R6:** If a request message flow sends an outbound two-way non-SOAP message, an outbound relationship from the message broker mediation is displayed in the topology view. An unmanaged server object is displayed to the right of the mediation object in the topology view. The operation name and the service port name of the unmanaged server are derived from the attributes of the request output node.
- **Rule R7:** No relationship is displayed for outbound one-way non-SOAP request messages.

For Collector nodes:

- **Rule R8:** If a message is propagated to a Collector node, a one-way relationship from the mediation object representing the input node from where the message originates, to the mediation representing the Collector node, is displayed in the topology view.
- **Rule R9:** If an output node that is placed in the message flow following a Collector node sends a request message, an outbound relationship from the Collector node mediation object is displayed in the topology view. An unmanaged node is displayed to the right of the mediation object if the server is unmanaged, or if the message is one-way non-SOAP. Otherwise, the mediation object that represents the managed server is displayed to the right of the Collector node mediation object in the topology view.

The following additional mapping rule is for MGet nodes:

- **Rule R10:** The MQGet node is not treated as a request input node. No mediation or relationship is displayed for a MQGet node.

Sample applications and resulting topology

This section illustrates several sample applications and the resulting topology display, following the applicable message flow to topology mapping rules.

Sample 1: Web service host

The web service host application is a sample application described in the WebSphere Message Broker version 6.1 documentation. For more information, see the WebSphere Message Broker documentation.

The message flows for this application are shown in Figure 26.

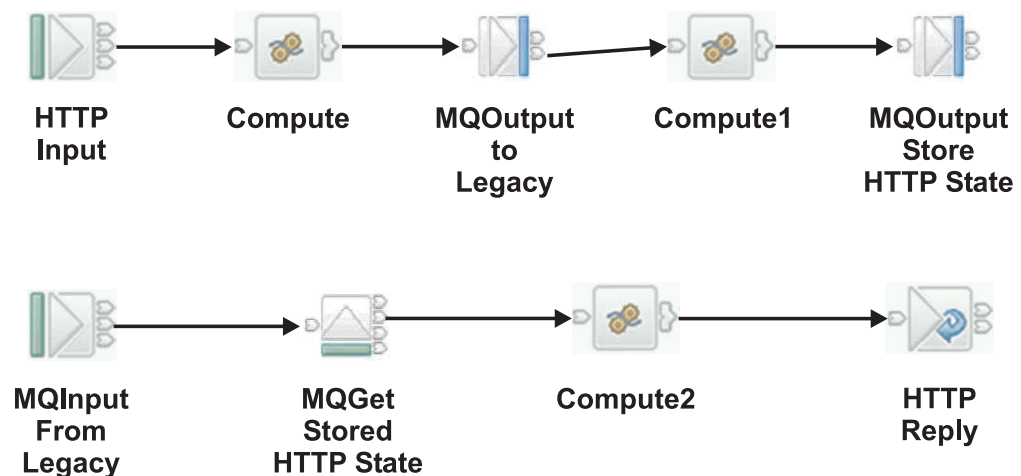


Figure 26. The message flows for the web service host sample application

This sample is a SOAP flow with HTTP nodes. The application displays two message flows; the first flow representing the request flow, and the second flow representing the response flow.

The request flow begins with a request input node, *HTTP Input*. According to mapping rule M1, this message flow is displayed in the service-to-service topology as the mediation object, *IA81CONFIN*, shown in the center of Figure 27. The label that is used for an object in the topology is the operation name for the node.

The input node for the response flow receives only response messages from the legacy application. Therefore, according to mapping rule M2, no separate mediation object is displayed in the topology for this second flow. The response flow is considered to be part of the mediation object, *IA81CONFIN*.

In the request flow, the single request input node, *HTTP Input*, receives SOAP requests from an unmanaged client. Because of mapping rule R1, this inbound SOAP message results in a corresponding inbound relationship to the message broker mediation object displayed in the topology. This relationship originates from the unmanaged object, *IA81CONFIN*, shown in Figure 27

In the request flow, you see two request output nodes, *MQOutput to legacy* and *MQOutput store HTTP state*. The *MQOutput to legacy* request output node sends a two-way non-SOAP request message to the legacy application. Therefore, according to mapping rule R6, the topology displays a corresponding relationship from the mediation object to an unmanaged server object, *MQOut...ST1_OUT1*.

The *MQOutput store HTTP state* request output node sends one-way non-SOAP messages to a queue. Therefore, according to mapping rule R7, no corresponding relationship or server object is displayed in the topology.

Due to mapping rule R10, the input node *MQGet stored HTTP state* has no corresponding mediation node or relationship displayed in the topology because the *MQGet* node is not treated as a request input node.

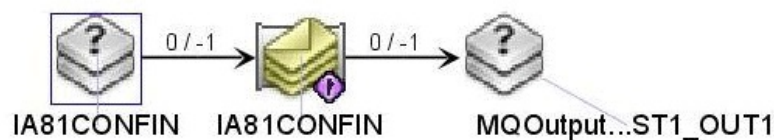


Figure 27. The corresponding service-to-service topology representation of the Web Service Host sample application

Sample 2: Address book

The address book sample consists of three message flows, shown in Figure 28 on page 140:

- The first flow is the *request flow*. This flow receives a non-SOAP request message from a client. The storeMQMD node sends MQMD as a one-way message to a queue. Finally, a non-SOAP request message is sent to the backend flow at MQOutput.
- The second flow represents the *backend flow*. This flow receives the non-SOAP request message from the request flow. A SOAP request is sent to a managed WebSphere Application Server, at the node *HTTP Request*. The flow then sends a reply message to the response flow, at the MQReply node.

- The third flow is the *response flow*. This flow receives the reply message from the backend flow. It also retrieves MQMD from the queue at the MQGet node, reStoreMQMD. Finally, a response message is sent to the client at the MQReply node.

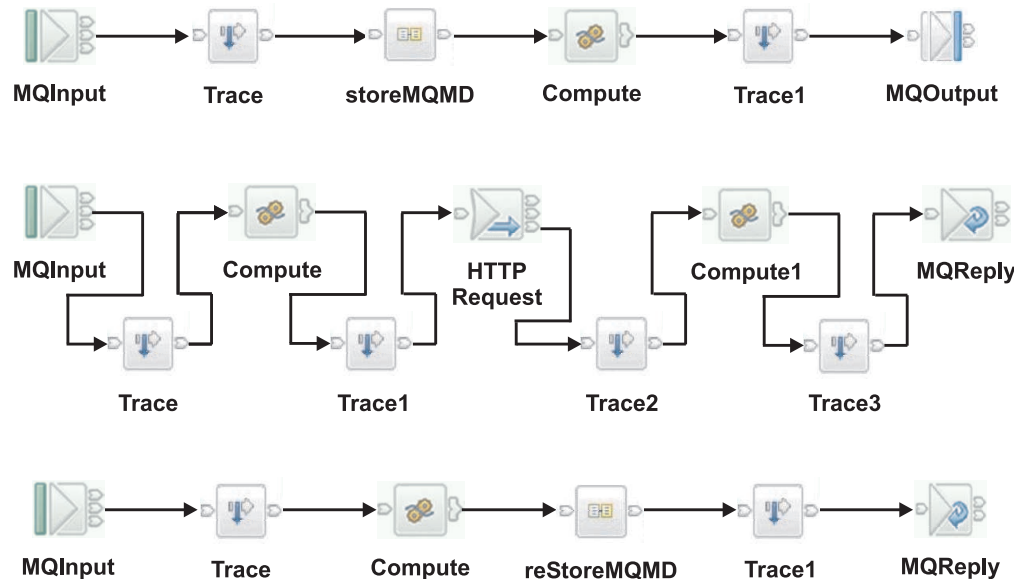


Figure 28. The message flows for the Address book sample application

This sample is a SOAP flow with MQ nodes. The request flow contains a request input node, *MQInput*. Therefore, according to mapping rule M1, the topology view includes a corresponding mediation object, *MQInput:AB_REQ_IN*, shown in the first flow in Figure 29 on page 141.

The backend flow also contains a request input node, *MQInput*. Therefore, the topology view also includes a corresponding mediation object, *MQInput:AB_SRV_REQ*, shown in the third flow in Figure 29 on page 141.

In the reply flow, the input node *MQInput* receives only reply messages. Therefore, according to mapping rule M2, no corresponding mediation object exists in the topology view, because it is considered to be part of the mediation object, *MQInput:AB_REQ_IN*.

The request flow receives non-SOAP request messages from a client. Therefore, according to mapping rule R5, the first flow in the topology view displays a relationship from the unmanaged object, *MQInput:AB_REQ_IN*, to the mediation object, *MQInput:AB_REQ_IN*.

In the request flow, the *MQOutput* node, *storeMQMD*, sends one-way non-SOAP messages to a queue. In this case, mapping rule R7 results in no corresponding relationship or server object displayed in the topology view.

The last node in the request flow, *MQOutput*, sends two-way non-SOAP request messages to the backend flow. Therefore, according to mapping rule R6, a relationship is displayed in the topology view from the mediation *MQInput:AB_REQ_IN* object to an unmanaged server object, *MQOutput:AB_SRV_REQ*.

The backend flow receives non-SOAP request messages from the request flow. Therefore, according to mapping rule R5, a relationship is shown in the third flow in the topology view from the unmanaged object, *MQInput:AB_SRV_REQ*, to the mediation object, *MQInput:AB_SRV_REQ*.

The HTTPRequest node, HTTP Request, sends two-way SOAP messages to a managed WebSphere Application Server. Therefore, the third flow in the topology view includes a relationship from the mediation node, *MQInput:AB_SRV_REQ*, to the managed server node, *findAddress*.

In the reply flow, no request input node or request output node exists. Therefore, in the reply flow topology, no relationship is displayed.

The MQGet node, *reStoreMQMD*, is not treated as a request input node. Therefore, according to mapping rule R10, no relationship is displayed in the topology.

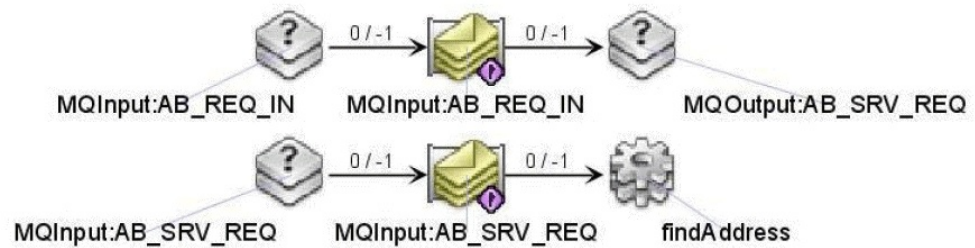


Figure 29. The corresponding service-to-service topology representation of the Address book sample application

Sample 3: Aggregation

The Aggregation sample application is a sample application for WebSphere Message Broker version 6.1. To understand how these flows work, refer to your WebSphere Message Broker documentation. Like the Address Book example, this sample application has three flows: a request flow, a response flow, and a backend flow. In the request (or *fan-out*) flow, multiple request messages are sent concurrently to backend applications using aggregation control. In the response (or *fan-in*) flow, multiple response messages are received and combined to build a reply for the client.

The request flow, shown in Figure 30 on page 142, is also referred to as a *fan-out flow*, because of the multiple request output nodes in the flow.

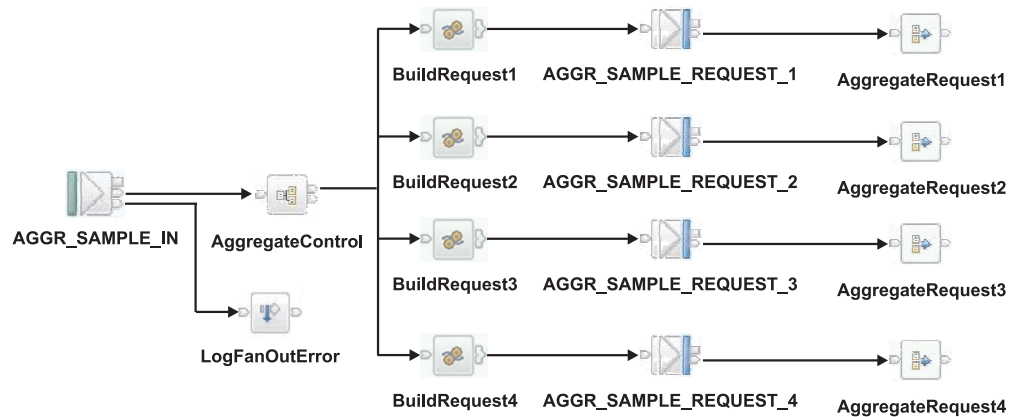


Figure 30. The request flow (also called the fan-out flow) for the Aggregation sample application

The response flow, shown in Figure 31, is also referred to as a *fan-in flow*.

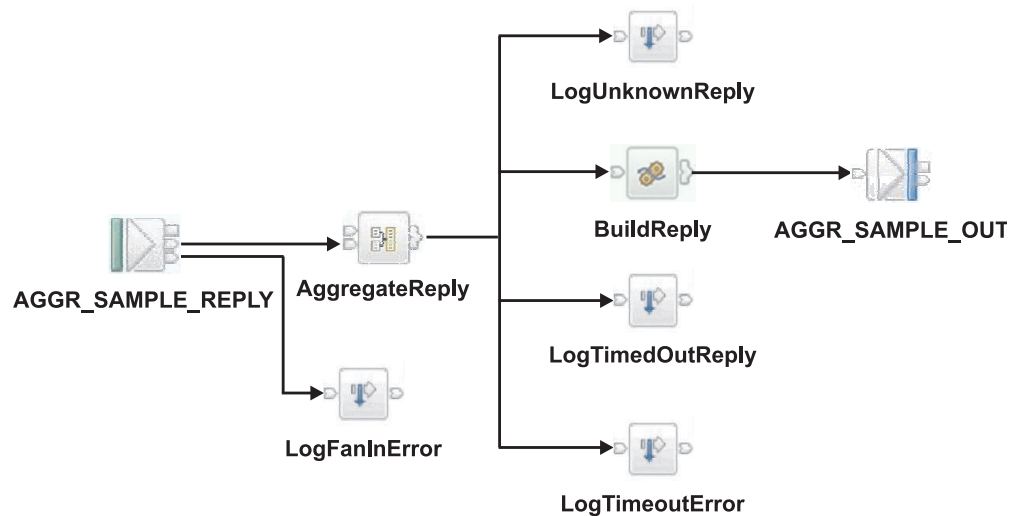


Figure 31. The response flow (also called the fan-in flow) for the Aggregation sample application

The backend flow, shown in Figure 32



Figure 32. The backend flow for the Aggregation sample application

The fan-out flow (Figure 30) contains a request input node, *AGGR_SAMPLE_IN*. Therefore, it has a corresponding mediation object, *AGGR_SAM...AMPLE_IN*, in the topology view, shown in Figure 30.

The backend flow (Figure 32) also contains a request input node, *AGGR_SAMPLE_REQUEST*. Therefore, it has a corresponding mediation object, *AGGR_SAM...REQUEST*, in the topology view, shown in Figure 32.

In the fan-in flow (Figure 31 on page 142), the input node *AGGR_SAMPLE_REPLY* receives only reply messages. Therefore, no separate mediation object is displayed in the topology for the fan-in flow. It is considered to be part of the mediation object *AGGR_SAM...AMPLE_IN*.

Figure 33 displays the corresponding topology display for these Aggregation sample application flows.

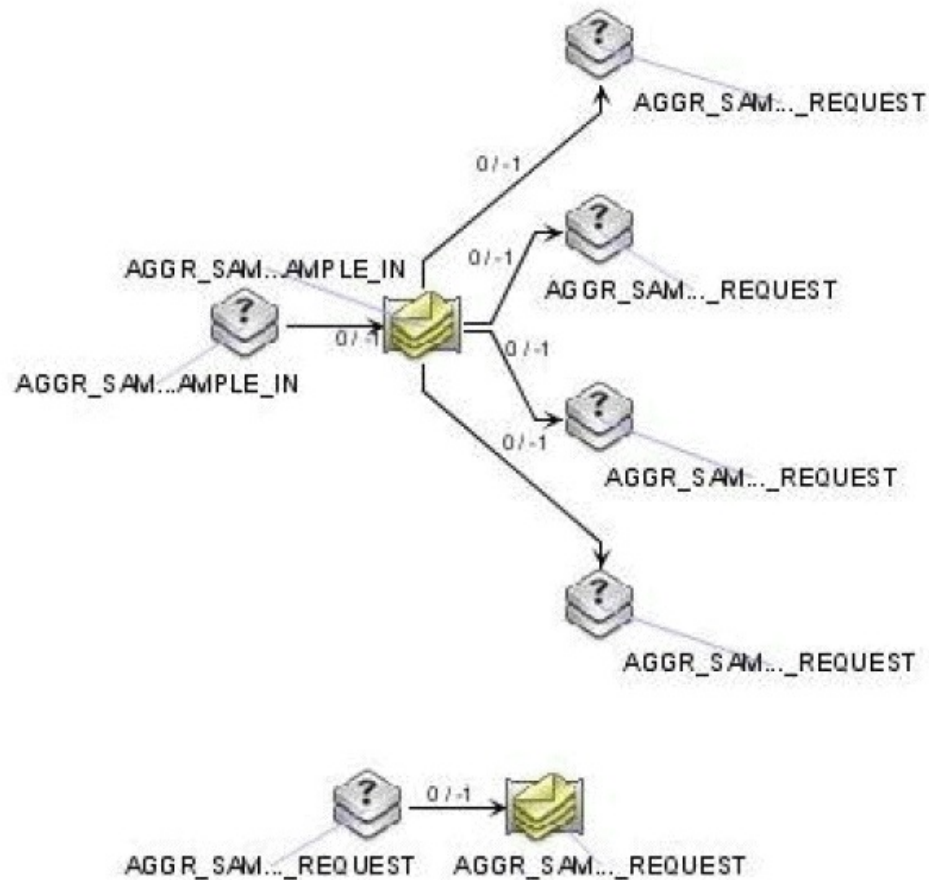


Figure 33. The corresponding service-to-service topology representation of the Aggregation sample application

In the fan-out flow, the input node *AGGR_SAMPLE_IN* receives non-SOAP messages from a client. Therefore, a relationship exists from the unmanaged object, *AGGR_SAM...AMPLE_IN*, to the mediation object, *AGGR_SAM...AMPLE_IN* in Figure 33.

Four request output nodes are contained in the fan-out flow, each sending non-SOAP two-way messages to the backend flow. Therefore, four corresponding relationships exist from the mediation object, *AGGR_SAM...AMPLE_IN*, to the four unmanaged nodes (all named *AGGR_SAM...REQUEST* in Figure 33).

The input node *AGGR_SAMPLE_REQUEST* in the backend flow receives non-SOAP messages from the fan-out flow. Therefore, a relationship exists from the unmanaged object, *AGGR_SAM...REQUEST* to the mediation object, *AGGR_SAM...REQUEST* in Figure 33.

The reply flow does not have a request input node or a request output node. Therefore, no relationship is displayed in the topology for the reply flow.

Sample 4: Automatic timeout

This example is from the Timeout sample application, which is provided with WebSphere Message Broker version 6.1. To understand how this application works, refer to your WebSphere Message Broker version 6.1 documentation.

The automatic timeout flow is shown in Figure 34.

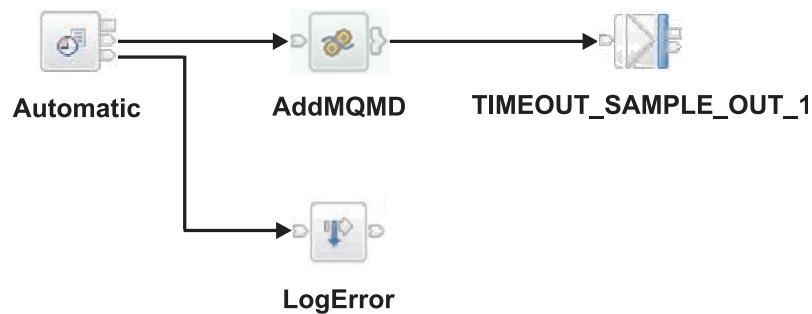


Figure 34. The message flow for the Automatic timeout sample application

The input node, Automatic, is a TimeoutNotification node and is not monitored by the ITCAM for SOA data collector in the WebSphere Message Broker environment. The request output node, TIMEOUT_SAMPLE_OUT_1, sends one-way non-SOAP messages to a queue. Therefore, no mediation object or relationship is displayed in the topology view for this flow.

Sample 5: Collector node

This Collector node example is a sample application, provided with WebSphere Message Broker version 6.1. To understand how this application works, refer to your WebSphere Message Broker version 6.1 documentation.

The Collector node flow is shown in Figure 35.

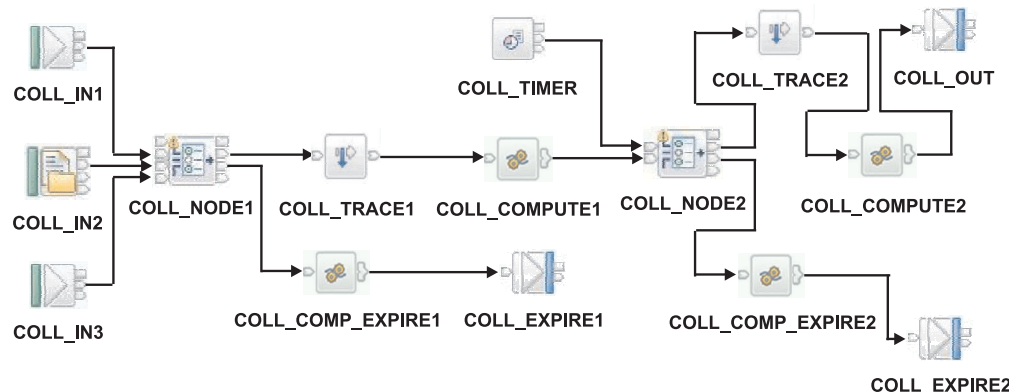


Figure 35. The message flow for the Collector node sample application

According to mapping rule M3, the Collector Nodes, COLL_NODE1 and COLL_NODE2, are both represented by corresponding mediation objects in the topology view in Figure 36 on page 145.

Similarly, the input nodes, COLL_IN1 and COLL_IN3, are also represented by their own corresponding mediation objects in the topology view.

The fileInput node, COLL_IN2, and the timer node, COLL_TIMER, do not have corresponding mediation objects in the topology because they are not supported by the ITCAM for SOA data collector in the WebSphere Message Broker environment.

According to mapping rule R8, a relationship in the topology view exists from mediation object COLL_IN1 to mediation object COLL_NODE1 because messages are propagated from COLL_IN1 to COLL_NODE1.

Similarly, a relationship in the topology view exists from mediation object COLL_IN3 to mediation object COLL_NODE1 because messages are propagated from COLL_IN3 to COLL_NODE1.

A relationship exists from mediation object COLL_NODE1 to mediation object COLL_NODE2 because messages are propagated from COLL_NODE1 to COLL_NODE2.

According to mapping rule R7, no mediation or relationship is displayed for the output nodes COLL_OUT, COLL_EXPIRE1, and COLL_EXPIRE2, because all of these output nodes send one-way non-SOAP messages.

Figure 36 displays the corresponding topology display for this Collector node sample application flow.

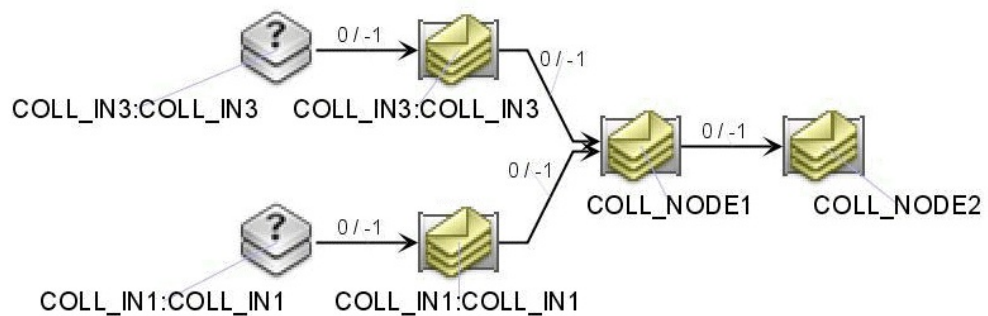


Figure 36. The corresponding service-to-service topology representation of the Collector node sample application

Limitations

This section describes additional considerations and limitations to remember when you monitor message flows in the WebSphere Message Broker environment.

Limitations of the HTTP Transport protocol

Unless otherwise specified, the following limitations apply to both SOAP and Non-SOAP message types:

1. If you use two different execution groups, one for the HTTP request flow and one for the HTTP response flow, the correlation between message flow instances cannot be maintained. In this case, only the request message flow instance is monitored.
2. The ability to monitor services by the remote host name or IP address form of Requester Identity is supported only for WebSphere Message Broker version 6.1.0.2 or later.

3. For a server-side message flow, if the correlation information between the request and response expires, the response flow fails to find the context for the request flow. In this case, the response message is ignored, and the message metric and content information for the response is not written to the log files.

Limitations of the JMS Transport protocol

Unless otherwise specified, these limitations apply to both SOAP and Non-SOAP message types over the JMS transport:

1. The ITCAM for SOA data collector for the WebSphere Message Broker environment depends on the correlation between the request message flow and the response message flow. Using the JMS transport, if the request and response message flows are deployed in separate execution groups, the correlation between the two execution groups cannot be maintained, and only metric data for the request message flow is logged.
2. Even when the request message flow and response message flow are in the same execution group, if the context information between the message flows times out or expires, the data for the response message flow is not written to the metric log.
3. To determine whether the message flow is a one-way or two-way request, the JMS transport relies on the JMSCorrelationID. If the messaging applications are setting the JMSCorrelationID for reasons other than to correlate between a request and response message flow, this identification is incorrect.
4. Because no remote host name and IP address is available in the WebSphere Message Broker JMS transport, the ability to monitor services by using the IP address form of the requester identity is not supported for the JMS transport protocol. You can only monitor by requester identity using the user ID form.

Limitations of the WebSphere MQ Transport protocol

Unless otherwise specified, the following limitations apply to both SOAP and Non-SOAP message types:

1. The ITCAM for SOA data collector for the WebSphere Message Broker environment depends on the correlation between the request message flow and the response message flow. Using the WebSphere MQ transport, if the request and response message flows are deployed in separate execution groups, the correlation between the two execution groups cannot be maintained.
 - If the data collector is unable to correlate an MQOutput or MQReply message with a previous MQInput or MQGet message, the data collector records the uncorrelated message as a *client request* event with no response time. This message is displayed in Tivoli Enterprise Portal as a requester entry in the Services Inventory table view.
 - If the data collector is unable to correlate an MQInput or MQGet message with a previous MQOutput or MQReply message, the collector records the uncorrelated message as a *server enter* event with no response time. This message is displayed in Tivoli Enterprise Portal as a provider entry in the Services Inventory table view.
2. For a one-way message flow, a default value is always assigned to the MQMD.ReplyToQ field. This value causes the message to be temporarily identified as two way. The condition is cleared after it expires.
3. For an XML message (non-SOAP) over WebSphere MQ, two entries are written to the metric log:

```
Provider INPUTNODENAME:QNAME <other data>
Provider OUTPUTNODENAME:QNAME <other data>
```

If you configure monitoring based on the second entry, the WebSphere Message Broker data collector does not record the message content, because the monitor control criteria does not match the `<input_node_name>:queue name`.

4. Because no host name and IP address is available in the WebSphere Message Broker WebSphere MQ transport, the ability to monitor services by using the IP address form of the requester identity is not supported for the WebSphere MQ transport protocol. You can only monitor by requester identity by using the user ID form.

Common Limitations of Transports

Be aware of these common limitations when using the various transports:

1. Some Non-SOAP messages (messages that contain a SOAP header, but that are not well-formed) are validated as SOAP messages. For example:

```
<soap:Envelope asdf lasd;jflsadjf;l asdjf;l asdjf  
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
adfasdlkfj;l asdjf ...
```

This message might be viewed as an attack. Upon receiving this message, an exception is thrown and transactions roll back.
2. The ITCAM for SOA data collector for WebSphere Message Broker does not support external correlation for non-SOAP messages in this release. This limitation affects the topology view in the following ways:
 - If the message from the client to the message broker is non-SOAP, instead of seeing a relationship from the client to the message broker mediation, you see a relationship from the client to an unmanaged server, and a relationship from an unmanaged client to the message broker mediation.
 - If the message from the message broker to the server is non-SOAP, instead of seeing a relationship from the message broker mediation to the server, you see a relationship from an unmanaged client to the server, and a relationship from the message broker mediation to an unmanaged server.
3. The ITCAM for SOA data collector for WebSphere Message Broker supports external correlation only for SOAP messages with a message domain MRM, XMLNSC, XMLNS, or SOAP. For SOAP messages with other message domains such as BLOB, external correlation of the SOAP message is not supported. In these cases, the service-to-service topology view displays a call relationship from an unmanaged client to the mediation for the message flow, and a call relationship from the mediation to an unmanaged server.
4. One-way non-SOAP requests are not displayed in the topology view.
5. If an MQGet node is recorded as a *Server Enter* event, the Metrics Record Control flag is turned off for the event. You cannot link from the associated row in Services Inventory table view to a corresponding node in the service-to-service topology view.
6. Between Server Enter and Server Leave, Message Flow might roll back because of business logic or exceptions of WebSphere Message Broker. Rollback causes one SOAP fault record to be written into the contents log and one corresponding record for Server Leave with the SOAP fault string to be written into the metrics log. Because the message itself cannot be accessed, the value written into the content log is an empty string.
7. If internal WebSphere Message Broker errors exist between the *Client Request* and *Client Response* events, a timeout occurs and the message flow is rolled back, and a *Client Response* event with a SOAP fault string is recorded into the metric log and the content log. Because the message itself cannot be accessed, the value written to the content log is an empty string.

8. Rollback can also cause *Server Leave* and *Client Response* events to be ignored. When rollback occurs, the message flow does not continue. Output nodes are never reached, and for the HTTP transport protocol the propagation of the HTTPRequest output node is never reached. In this case the corresponding content and metric log file is not created.
9. The ITCAM for SOA data collector cannot correlate a response message with a request message if either of the following conditions occur:
 - The request and response message flows are deployed in separate execution groups.
 - The response fails to be returned before the timeout limitation expires.

In this situation the response flow is displayed in the topology view along with the request flow. Typically, when correlation works correctly, only the request flow is displayed in the topology view.

Limitations of support for SOAP nodes

The following limitations apply to ITCAM for SOA support for SOAP nodes.

1. If you use two different execution groups, one for the SOAP request message flow and one for the SOAP response message flow, the correlation between message flow instances cannot be maintained. In this case, only the request message flow instance is monitored.
2. For a server-side message flow, if the correlation information between the request and the response expires, the response flow fails to find the context for the request flow. In this case, the response message is ignored, and the message metric and content information for the response is not written to the log files.
3. External correlation for SOAP messages without SOAP Envelopes is not supported. SOAPRequest nodes, SOAPAsyncRequest nodes, and SOAPReply nodes accept messages without SOAP Envelopes as input messages. The data collector for WebSphere Message Broker cannot include correlation support for these messages. For these nodes, there is no relationship displayed in the topology view from the mediation to the downstream mediation or the application server. To avoid this problem, you can add a SOAPEnvelope node immediately before the SOAPRequest, SOAPAsyncRequest, or SOAPReply nodes, and select the **Create new envelope** option for the SOAPEnvelope node to add a SOAP envelope for the SOAP message.
4. The data collector for WebSphere Message Broker cannot correlate a SOAPAsyncResponse node with a corresponding SOAPAsyncRequest node without WS-Addressing headers. No *Client Response* event record is logged for such SOAPAsyncResponse nodes. To avoid this problem, select the **Place WS-Addressing headers into LocalEnvironment** option for the SOAPAsyncResponse node, and redeploy the flow.
5. Because of a known limitation of Message Broker version 6.1.0.2, the data collector for WebSphere Message Broker logs *Client Response* event records for one-way messages sent by SOAPRequest nodes, and they are included in the Services Inventory table. In the topology view, a relationship from the mediation representing the requester flow to an unmanaged server is displayed.

Chapter 8. Workspace for monitoring service health

A *service* is a set of service operations that performs a specific task, sending requests to, or acting on requests from, other operations. An *operation instance* is the deployment of a uniquely identified service operation to an application server.

ITCAM for SOA groups the operation instances that have common service port names and namespaces, operation names and namespaces, and mediation types into *operation aggregates*. Operation instances and operation aggregates interact with each other, and the resulting *call relationships* are derived by ITCAM for SOA, forming the basis for the service-to-service topology that is displayed in the various operational flow workspaces and views. For more information about topology, see Chapter 7, “Workspaces for service-to-service topology,” on page 71.

While you view the operation aggregates in your service-to-service topology, you might think of a number of these operation aggregates or the flows in which they participate as collectively representing an application or a business process. You can think of this collection of operation aggregates as a *group*.

Using ITCAM for SOA, you can select service flows, parts of service flows, or specific groups of operation aggregates in your service-to-service topology views, and assign them to a uniquely identified group. You can create multiple groups, and assign operation aggregates to more than one group at the same time. After you create one or more groups, you can display them in a high-level summary workspace. Using this Group Summary workspace in graphical or table mode, you can see at a glance the overall *health* of your groups, expressed in terms of overall message volume, service unavailability, performance, and status.

You can also include a Group Summary view in any Tivoli Enterprise Portal workspace.

From the Group Summary workspace or view, you can display an operation flow view, showing the service-to-service topology for the selected group.

Service groups and process groups

A *service group* is a set of related service operations that collectively can represent or encompass a business function or application in your enterprise. The group can consist of a service flow, a subset of a flow, or any collection of operation aggregates that represent something meaningful to you in your monitored environment.

A *process group* is a Business Process Management (BPM)-specific group that aggregates elements that are part of a BPM solution, such as applications, modules, components, or operations. A process group allows the monitoring of interactions between the components of the solution. This monitoring provides BPM-specific context information.

Important: Process groups are a new feature of ITCAM for SOA version 7.2. If you upgrade from a previous version of ITCAM for SOA, service groups that contain BPM operations in installed applications are automatically changed to process groups.

The Group Summary view contains an array of group objects, each with its own indicators of service health. Figure 37 displays an example of a typical service group, pointing out the meaning of the various parts of the display.

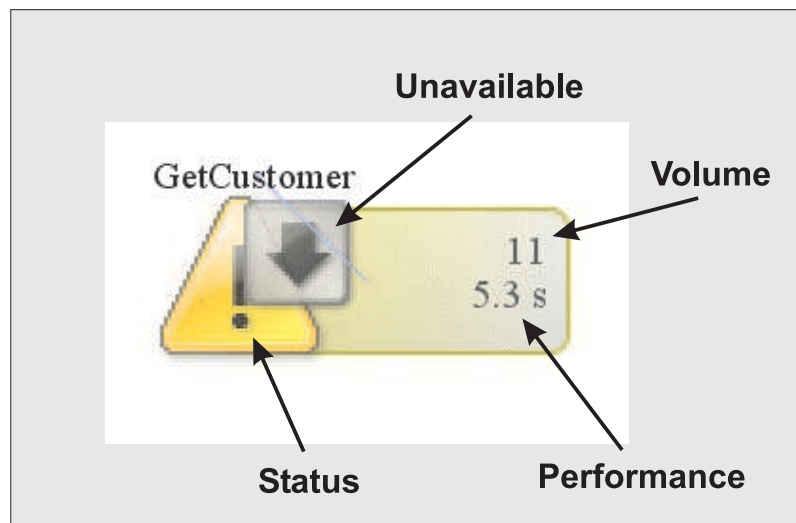


Figure 37. Sample Service Group object

Figure 38 displays an example of a typical process group with blue icon, pointing out the meaning of the various parts of the display.

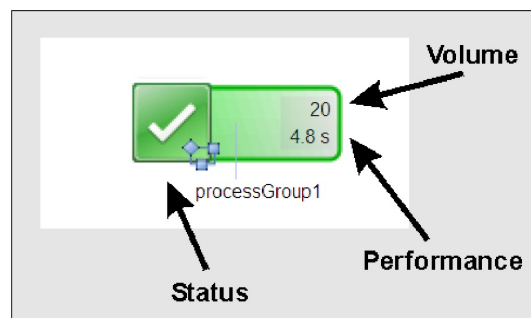


Figure 38. Sample Process Group object

Determining front-end services

The indicators of overall service health are determined relative to the *front-end* services that are associated with the group.

Think of a front-end service for a group as an entry point to the flows that make up the group. You do not define a front-end service, but it is based on the intersection of the message flows within the group. You can recognize a front-end service by examining the operation aggregates that are assigned to the group in the Operation Flow view. An operation aggregate is considered a front-end service if the following conditions are met:

- The operation aggregate is a member of the group.
- The operation aggregate is either called by an operation aggregate that is not a member of the group, or it has no callers.

Health indicators of a group

The group object that is displayed in the Group Summary view conveys the following health indicators:

Status Summarizes the functional health of the group. This status is calculated by using the set of situations that are open for the operation aggregates that are members of the group and the relationships of those operation aggregates. The color and shape of the status indicator provides a quick indication of the overall status for the group. This status is calculated based on a set of rules and algorithms that are described in detail in Appendix B, “Determining status for operation instances, operation aggregates, and groups,” on page 331.

Volume

Summarizes the message traffic (expressed in message counts) for operation aggregates within the group. It is a count of the number of messages that are observed at the *Provider Enter* point on the front-end services of the group. The format of the message count is expressed in as few characters as possible, for example, 12k (thousands), or 13.2m (millions). The value is fully expressed in the flyover window and the details window. If a group has multiple front-end services, the volume is the sum of the message counts for each front-end service.

For a group that includes Business Process Definitions (BPDs), the number of processes that are currently active for each BPD is added to the volume, instead of a message count for the BPD.

Performance

Summarizes performance in terms of the average response time metrics that are reported for a front-end service in the group. When the group has multiple front-end services, this metric is the average of the response times that are observed across all of the front-end services in the group, at the *Provider Leave* point on the front-end services of the group. The format of the response time is expressed in as few characters as possible, for example, 8.6s (seconds), or 15.7ms (milliseconds). The value is fully expressed in the flyover window and the details window.

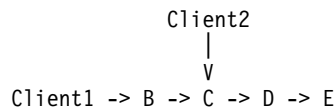
Important: For operations involving user response, such as BPDs, the average response time can be long (sometimes hours or days).

Unavailability

Summarizes the overall unavailability of the group. The unavailability of a group is based on situations that are open for the front-end services of the group and whether those situations are identified as unavailability situations. If at least one front-end service has an open unavailability situation, then the group is considered to be unavailable. For more information about creating unavailability situations and including them in the determination of unavailability for groups and for operation aggregates and operation instances, see “Configuring for unavailability” on page 180.

Define meaningful groups: Health indicators are only as meaningful as the group definition. When you create groups and associate them with operation aggregates and service flows, make sure that the groups are selected in a correct way, so that the health information reflects the impact on your tasks.

Groups with multiple front-end services: Be careful when defining groups with multiple front-end services. For example, consider the following type of flow:



If you create a group that includes operation aggregates B, C, and D, both B and C are considered front-end services. Because the volume is the sum of the message counts of each front-end service, the messages going through operation C are counted twice (once for the front-end service C and also for the front-end service B, because its message count includes messages flowing from operation B to operation C). To avoid this double counting, do not include operation B in the group, because operation C is the real entry point to the flow.

The Group Summary workspace

After you define one or more groups for your monitored environment, you can display them in the Group Summary workspace. This workspace is accessed from the Services Management node in the Navigator ITCAM for SOA view. For more information about the ITCAM for SOA navigator view, see “The Navigator ITCAM for SOA view” on page 26.

If ITCAM Agent for WebSphere Applications is installed on the monitored server, you can also access this workspace by using links from the Business Process Manager Summary, Applications, and Application Health Summary workspaces. Right click any application and select **Selected Application - Group Summary** to view the Group Summary workspace. In this case, the workspace displays only the process groups that include any components or operations that are a part of the selected application.

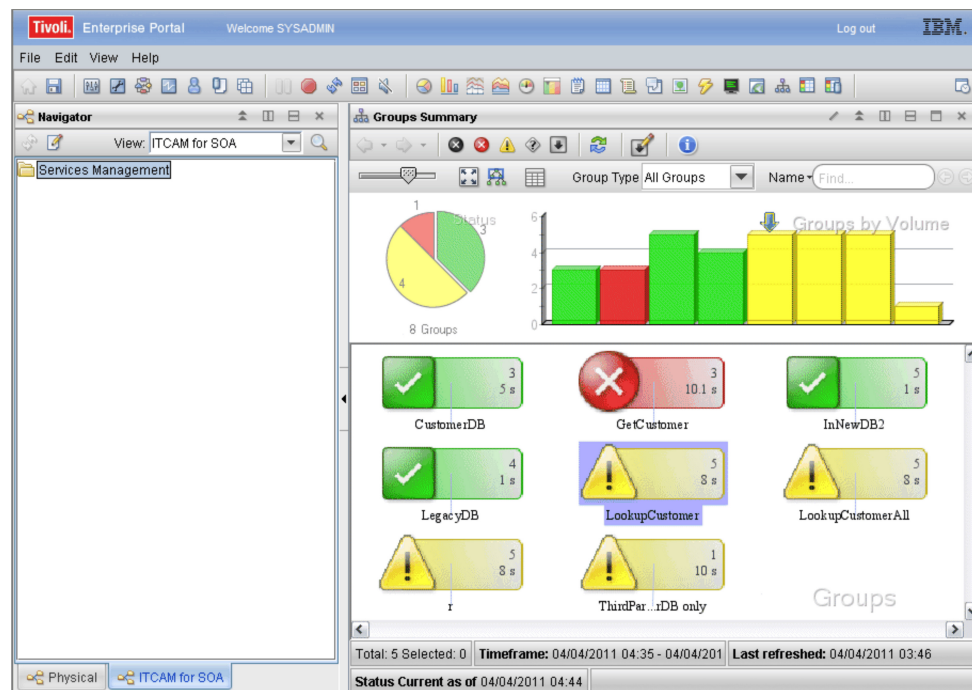


Figure 39. The Group Summary workspace

You can filter groups by group type with the **Group Type** list. To switch between the filtered views, select **All Groups**, **Service Group**, or **Process Group** from the list, as shown in Figure 39 on page 152.

If you display the Group Summary workspace before defining any groups, the Group Summary view is empty. To create one or more groups, you can open the Groups dialog by either of the following methods:

- If the Group Summary view is empty, click **Manage Groups** at the top of the view.
- Right-click anywhere in the Group Summary view and select the Manage Groups option.
- In the ITCAM for SOA Navigator view, right-click the Services Management node and select the Operational Flows workspace link to display the service-to-service topology view. In this view, highlight one or more operation aggregates, right-click and select Manage Groups to open the Groups dialog.

The Group Summary view

The Group Summary view provides a high-level picture of the overall health of your monitored services. It helps you to quickly identify trouble spots in your environment, and is useful for managing large numbers of services, service flows, and operation aggregates.

Figure 39 on page 152 displays an example of the Group Summary view that is displayed in the Group Summary workspace. You can also include the Group Summary view in any Tivoli Enterprise Portal workspace.

The Group Summary view has three main areas:

- An array of defined service and process groups.
- A vertical bar chart at the top, that can be configured to display either group message volume or performance for each service or process group.
- A status summary pie chart showing the number of groups defined and the number of groups at each level of status. The chart also displays the number of unavailable groups.

The view also contains a status area at the bottom that displays the following information:

- The total number of groups and the number of selected groups.
- The timeframe for current metrics (you cannot select a historical timeframe). The current timeframe is calculated in the same way as the current timeframe for the Operational Flow workspaces. It is shown in the timezone of your Tivoli Enterprise Portal client.
- The last time that the view was refreshed. The time is shown in the time zone of your Tivoli Enterprise Portal client.
- The time when the status was last calculated. By default, status is recalculated every 2.5 minutes.

The groups array

In the lower portion of the view, an array of *group* objects is displayed. The group objects are sorted alphabetically by group name, similar to the example shown in Figure 40 on page 154.

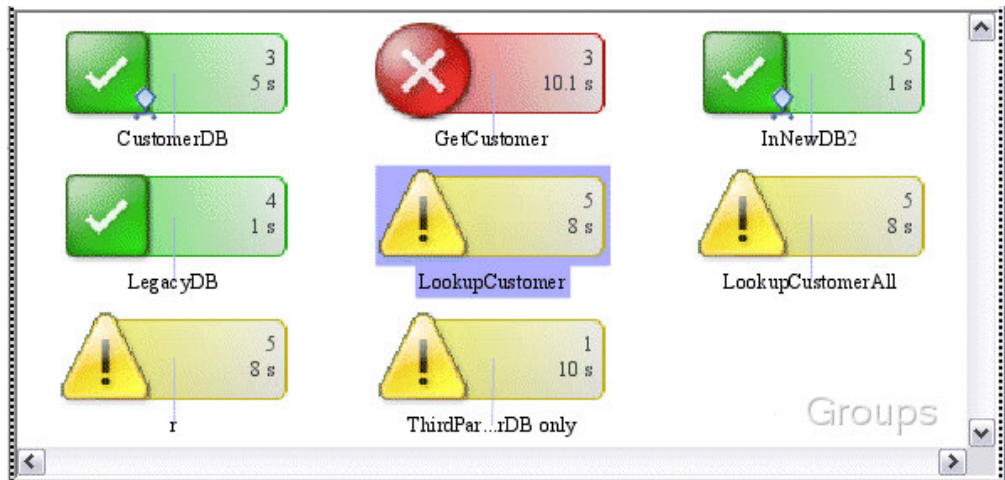


Figure 40. Groups displayed in an array in the Group Summary view

This part of the view helps you to see the overall health and status of your groups. You can select a group and display additional information about the group, manage your defined groups, and display the associated operation aggregates in a more detailed group topology view.

If any group is unavailable, an additional down arrow icon is displayed for the group. For details, see “Displaying unavailability” on page 181.

Table mode: This portion of the view can also be displayed in table form. The table format is useful for managing large numbers of groups. The table format, similar to the example shown in Figure 41, allows for multi-column sorting and filtering, and presents the data in a more accessible format. By default, the table is sorted by the Status column, then by the Availability column, and then by the Name column. As with the other topology workspaces, you can switch between table and topology views by using the icons in the action bar. The table view includes the following columns:

- Name
- Description
- Aggregates (the number of operation aggregates in the group)
- Status
- Availability (counts the number of unavailability situations)
- Performance (Response time, seconds)
- Volume (message counts)
- Group Type (service or process groups)

You can sort and filter the table by any of the columns.

Name	Description	Aggregates	Status	Availability	Performance	Volume	Group Type
<no filter>	<no filter>	<no filter>	<no filter>	<no filter>	<no filter>	<no filter>	<no filter>
Customer		1			No Value		All Service Groups
Faults		1			No Value	No Value	All Service Groups
JUNIT		2			No Value		All Process Grou...
medFaultsApp		2			No Value		All Process Grou...
PG		0			No Value	No Value	All Process Grou...
Scenario1		12			No Value	No Value	All Process Grou...

Figure 41. Sample Group Summary table view

Configuring the table mode threshold: This graphical view automatically switches to table format when the number of groups exceeds the threshold settings defined in the properties for the workspace. To configure this threshold, see “View properties” on page 157.

Sorting table columns for groups with no value: If you want to sort by those groups that have no value (a blank cell) in the table column, you can sort on the appropriate column in the table. Sorting moves all of the rows in ascending or descending order.

The groups bar chart

Across the top of the view is the *groups* bar chart, similar to the example shown in Figure 42. The groups bar chart consists of a three-dimensional vertical bar chart, with one bar representing each defined group. Each semi-transparent vertical bar is color-coded by the overall status of its associated group. The height of each bar represents either the overall message volume for the group (the default) or the average response time. (You can select which of these health aspects is displayed. For more information, see “View properties” on page 157.). Bars representing groups with values of -1 or with no value are plotted on the chart with a height of 0.

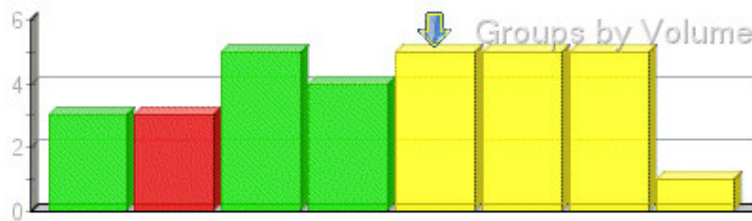


Figure 42. Sample groups chart

The entire set of groups is always displayed in the chart, and groups are listed left to right in alphabetic order by name. The groups chart is unaffected by the use of status filter controls. If you hold the cursor over a bar in the chart, the flyover window for that group is displayed. To select the bar and its associated group in the Group Summary view, click a bar in the chart (the selected bar is indicated by the down arrow located above the top of the bar). To display the Operation Flow view for the selected group, double-click a bar in the groups chart.

The status summary pie chart

Beside the groups vertical bar chart is a *status summary* pie chart. The chart displays the number of groups that are at each status level (*Fatal*, *Critical*, *Warning*, *Unknown*, or *Normal*). Use the mouse cursor to view the tooltip for each status level.

The total number of groups is also displayed. The number of unavailable groups (not shown in this example) is also displayed if there is at least one group determined to be unavailable.

View actions

The action bar at the top of the Group Summary view includes the following functions (not all of these functions are available in table mode):

Topology navigation

After you double-click a group to display its associated operation aggregates in the Operation Flow for Service Group view or Operation

Flow for Process Group view, you can use the forward and back arrows to navigate between the Group Summary view and the operation flow view.

Refresh

The standard Tivoli Enterprise Portal function for refreshing the view.

InfoTips

Additional explanatory information is displayed about the objects in the Group Summary view. Additional help information for each group is also enhanced with additional information when the hover help is displayed.

Zoom Used to enlarge or shrink the group objects in the view.

Fit to View

Used to fit the group objects in the available view space.

Overview

Provides a high-level look at the group topology. This function is useful if you zoom in on the group topology.

Table mode

Use this control to toggle the group display between topology and table formats. When in table mode, another control is displayed to switch to topology mode.

Status filters

Use this set of buttons to filter the display of groups in graph mode. Groups matching the selected status level (*Fatal*, *Critical*, *Warning*, or *Unknown*) are shown normally, and groups that do not match the selected combination of status levels are displayed as inactive. For example, if you click the *Fatal* filter control button, all groups that do not have a status of *Fatal* are dimmed in the display. Clicking the *Fatal* filter control again restores all groups in the display. You can activate more than one filter at a time in any combination to filter out the status levels that are not of interest.

In the same way, you can filter the display of groups in graph mode to display only those groups matching the selected unavailability state.

These filter buttons do not apply when groups are displayed in table mode. To filter status levels as wanted, use the usual column filtering functions that are provided in the table.

Configure Unavailability

Use this button to associate one or more defined situations with the unavailability status of the group. When a situation that is associated with unavailability is triggered for a front-end service of the group, the group is considered to be unavailable.

For more information about configuring for unavailability, see “Configuring for unavailability” on page 180. For more information about the metrics that you can include in situations for determining the unavailability of your groups, see “Measuring service unavailability” on page 214.

Search

Use this search function to locate groups in the view by several different criteria. To select the wanted search criteria from the selection list, click the down arrow, and type the search criteria to match. All groups that match the search are highlighted in the view. You can search the groups using any of the following criteria:

Name Type the name of the group. All groups with matching names are highlighted.

Description

Type the text to search for in the description of each group.

Performance

You can select to locate all groups with performance values greater than or equal to, or less than or equal to, a specified performance threshold (in milliseconds). Be sure to enter a whole number, such as *8600* (milliseconds), instead of the short form, such as *8.6s*.

Volume

You can select to locate all groups with message volume values greater than or equal to, or less than or equal to, a specified volume threshold. Be sure to enter a whole number, such as *3500*, instead of the short form, such as *3.5k*.

You can also specify the special case values of *-1* and *No Value* for Performance and Volume searches using the search function.

View properties

You can configure several properties that apply to both the Group Summary view and to the Operation Flow for Service Group view. To configure the view properties, you can right-click in the view and select **Properties**, or select **Edit -> Properties** from the menu bar, or press **Ctrl+R**.

In the navigation tree of the Properties page, select the **Group Summary** node to display and modify the view properties in the **Configuration** tab.

You can configure the following properties under the **Group Summary** area:

Switch Group Summary to table mode automatically at

By default, the Group Summary view is configured to switch to table mode if the number of groups exceeds 50 in the display. You can configure this setting to *Never*, *50*, *100*, *200*, or *Always*.

Plot performance for groups in the Group Summary (instead of volume)

By default, the height of the vertical bars in the groups chart in the Group Summary view displays message volume for each group. To display average response time performance instead of message volume in the groups chart, select this check box. The check box is cleared by default.

Allow positioning of groups in the Group Summary

To reposition the group objects in the Group Summary view by dragging them with your cursor, select this check box. The check box is cleared by default. The repositioning of the group objects is not remembered across refreshes of the view.

You can configure the following properties under the **Operational Flows** area:

Switch Operation Flow view to table mode automatically at

By default, the Operation Flow portion of the Group Topology view is configured to switch to table mode if the number of operation aggregates exceeds 50 in the display. You can configure this setting to *Never*, *50*, *100*, *200*, or *Always*.

Switch Interaction Detail view to table mode automatically at

By default, the Interaction Detail portion of the Group Topology view is


configured to switch to table mode if the number of operation instances exceeds 50 in the display. You can configure this setting to *Never*, *50*, *100*, *200*, or *Always*.

Display bar chart as the default in the tooltip for call relationship data

You can display metric data about the call relationship between two objects in the Group Topology view by either holding your cursor over the line drawn between two operation attributes (in the Operation Flow portion) or two operation instances (in the Interaction Detail portion), or right-clicking the line between two objects and selecting **Show Metrics**. The resulting display shows metric data for the call relationship in both chart and table formats, available on two separate tabs. Selecting this check box displays the **Chart** tab by default. Clearing this check box displays the **Table** tab by default. If needed, you can still manually select either tab to display the data in chart or table form, .

Including the Group Summary view in a workspace

You can include the Group Summary view in any Tivoli Enterprise Portal workspace by completing the following procedure:

1. Select the workspace where you want to create the view, such as an Operational Flows workspace.
2. If you want the view to occupy a new space, in the action bar of the Navigator view in the workspace, create a new empty view in the workspace and select the **Split vertically** or **Split Horizontally** icon.
3. From the action bar of the workspace, select the Topology icon  and drag it into the empty view space.
4. The Select Topology Source window offers you several views to include in your workspace. From the available choices, select **Group Summary** and click **OK**. The Group Summary view is added to the workspace.

Displaying group details

You can move your mouse pointer over a group and display a flyover window about the group, or you can right-click on the group. From the context list, select **Show Details**. Figure 43 on page 159 displays an example of the service group details that are displayed.

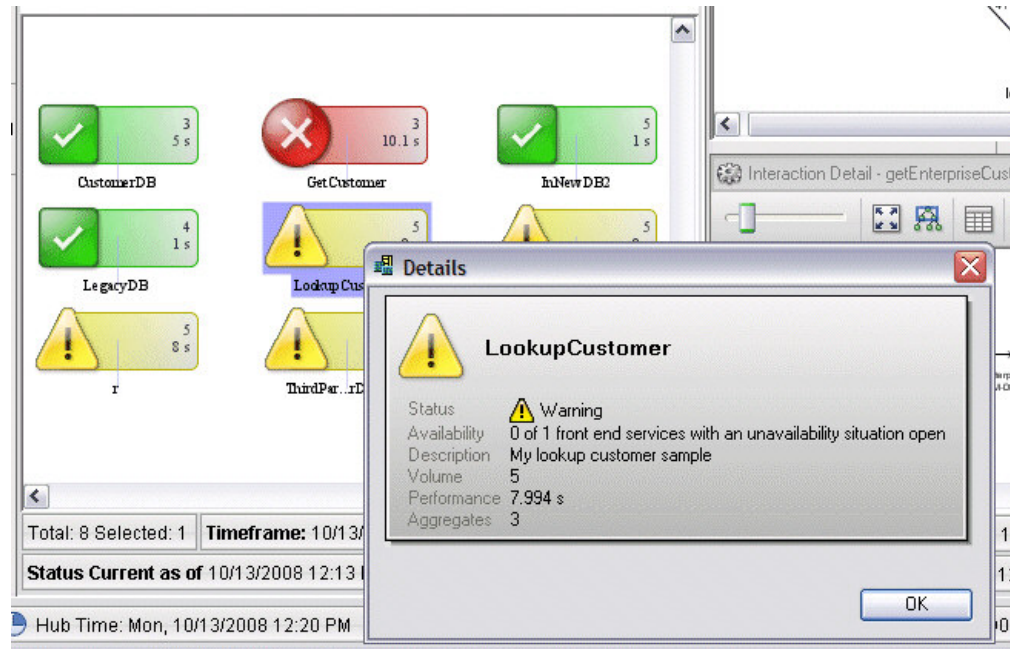


Figure 43. Sample Service Group with additional details displayed by selecting Show Details

The group details displays the health of your group in more detail, including overall status, average response time, message volume, and the unavailability status. The text description of the group is also displayed, and the number of operation aggregates that make up the group is also displayed.

Changing the situation impact on status of group

Any situation that you create in ITCAM for SOA has a default impact on availability and status of a group. You can also customize the impact of situations on a particular service group or process group.

To set the custom impact of situations, select one or several groups in the Group Summary view. Then right-click a selected group and select **Configure Situations Settings**.

The **Situation Setting for Groups** window opens.

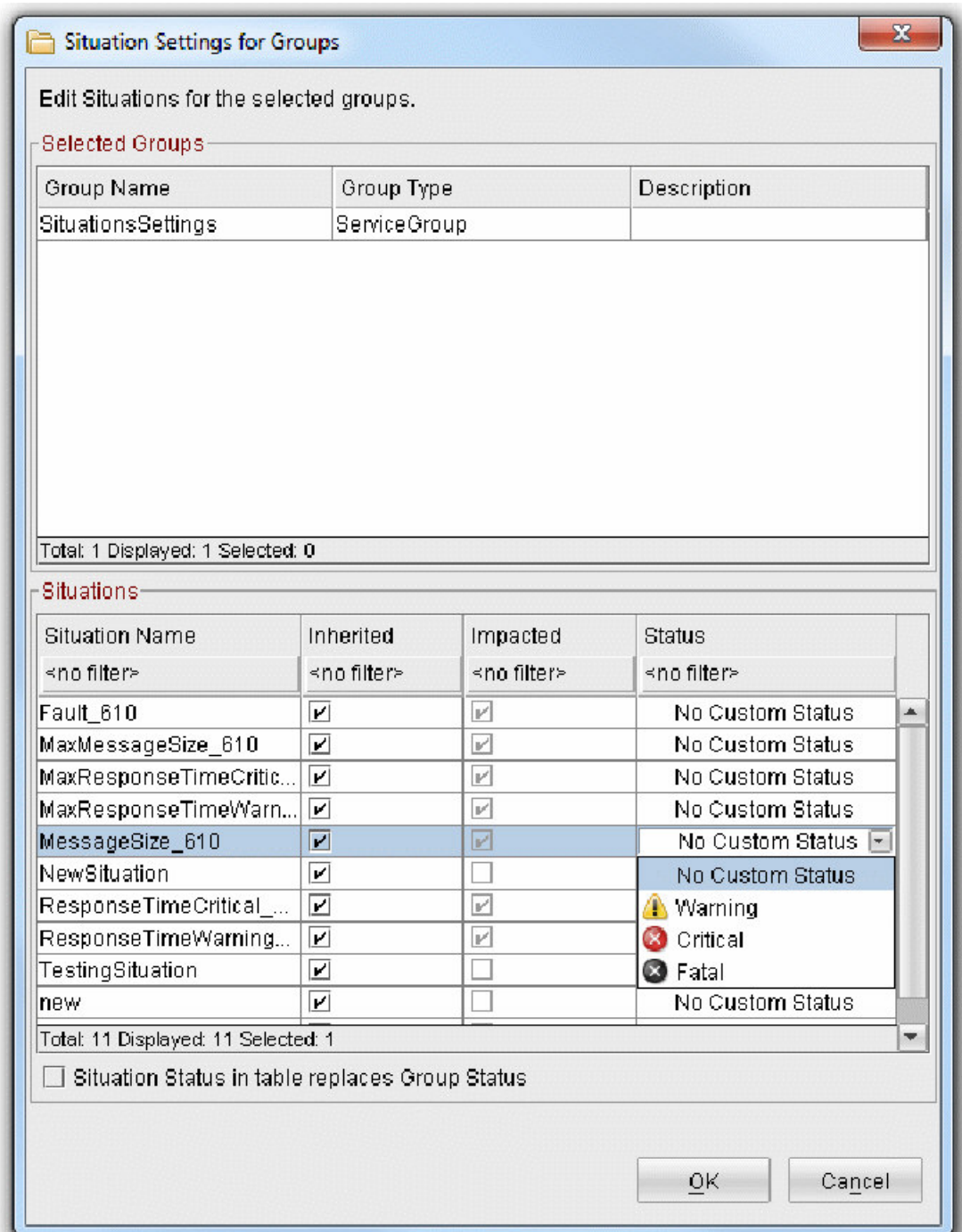


Figure 44. The Situation Setting for Groups window.

In the **Selected groups** table, you can see which groups are affected by the situation settings. This table is for information purposes only. To select different groups, close the window, select the groups, and open the window again.

The **Situations** table lists all the situations that are currently defined in the ITCAM for SOA agent. You can set the impact of every situation on the group availability and status:

- Select **Inherited** to use the global unavailability impact that is set for the situation. If the situation is configured as setting a group to Unavailable, this

setting applies. For more information about configuring situations for unavailability, see “Configuring for unavailability” on page 180.

- Select **Impacted** if the situation makes the group unavailable. If both **Inherited** and **Impacted** are cleared, the situation has no effect on group availability.
If **Inherited** is selected, this check box is disabled. In this case, the check box is displayed as selected if the situation is globally configured as setting a group to Unavailable.
- In the **Status** field, you can select the group status that the situation sets when it is triggered. If **No custom status** is selected, the global status configured for the situation is set.

Select the **Situation status in table replaces group status** check box to ensure that the status settings completely define group status, overriding the global status. If the check box is not selected, ITCAM for SOA uses the most severe of the following two status values: the status that is configured for the situation globally and the status that is determined by the table.

For example, a triggered situation might determine a Warning status according to the table, while it determines a Critical status globally. If **Situation status in table replaces group status** is selected, the group status is Warning; otherwise, it is Critical.

If you selected multiple groups, different settings might exist for these groups. In this case, the affected check box display a gray box, and the affected status fields display **Mixed**. If you change any setting, it is set to the new value for all the selected groups. This change becomes permanent when you click **OK**. If you do not change a gray or mixed setting and click **OK**, the settings remain unchanged for each of the groups.

Tip: If the **Situation status in table replaces group status** check box displays a gray box, after you select or clear it, you cannot return a gray box. For other check boxes, if the original value was gray, you can toggle between deselected, cleared, and gray values. If a gray value is set, the previous settings for these groups are not changed.

To save the changes, click **OK**.

Displaying the topology view for a group

From the Group Summary view, you can display the service-to-service topology view (also called the Operation Flow for Service Group or Operation Flow for Process Group view) for the operation aggregates that are associated with a group. You can either double-click the group object (or the row in the table, if in tabular mode) or right-click the group (or table row) and select **Show Service Group Topology** or **Show Process Group Topology** from the displayed menu.

The Operation Flow for Service Group or Operation Flow for Process Group view is displayed in the same view space as the Group Summary view. You can return to the Group Summary view at any time by clicking the **Back** (left arrow) icon at the top of the view.

The Operation Flow views show all of the flows that pass through operation aggregates that are members of the selected group. Aggregates that are members of the group are highlighted in the topology view. Aggregates that are not members of the group but which are part of the displayed flow are included in the topology view without being highlighted.

Similar to the Operation Flow for Application Server view, the Operation Flow view displays only the operational flows in which the operations from the selected group participates.

On the first visit, the upper part of the view is maximized (the Interaction Detail portion of the view is empty) until you select an aggregate to display its flows in more detail in the Interaction Detail portion of the view. You can move the cursor over any of the operation aggregate nodes and links to see flyover windows with additional information.

Front-end services are identified by a rounded square superimposed on the operation aggregate. The flyover window for the operation aggregate also identifies it as a front-end service. When the topology is displayed in table mode, the Front-End Service column indicates for each operation aggregate if it is a front-end service.

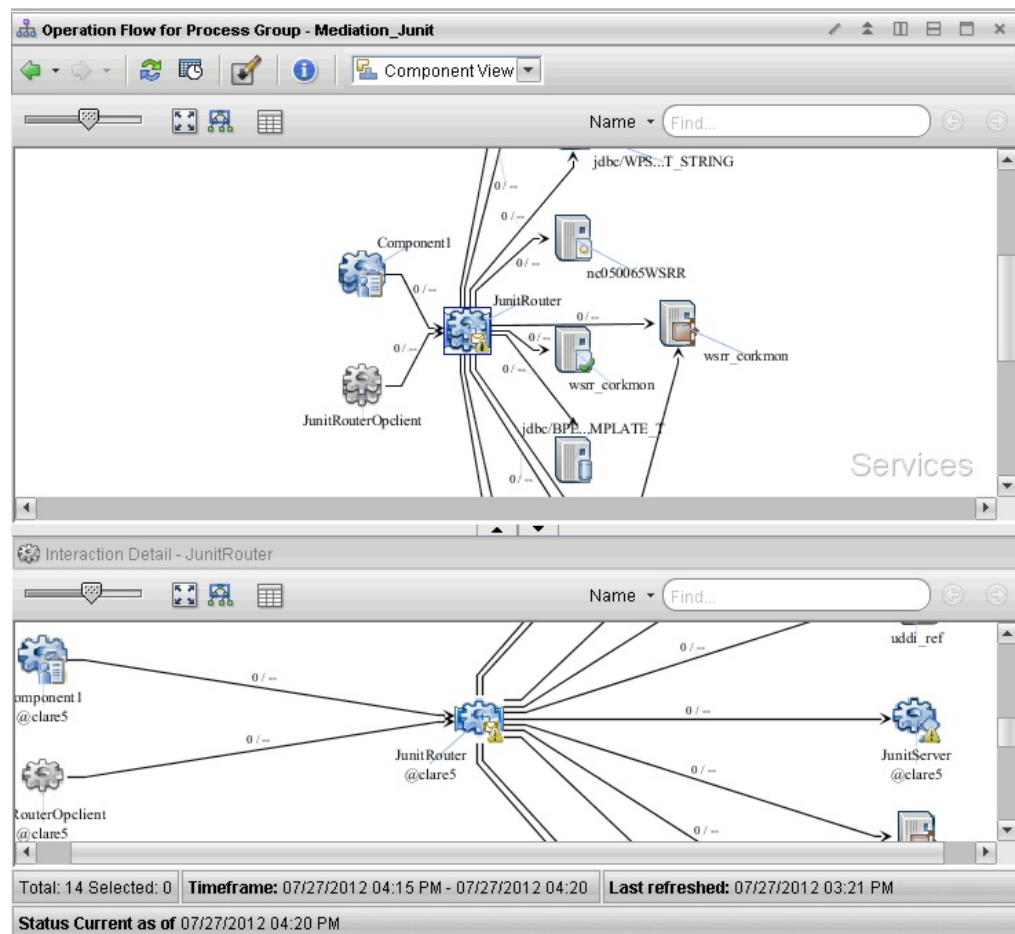


Figure 45. Sample Operation Flow for Process Group view

In this Operation Flow for Service Group or Operation Flow for Process Group view, you can use all of the actions and dialogs that are available with other operational flow workspaces:

- This view uses the same view toolbar, topology toolbar, and status area format.
- You can switch between topology and table display modes.
- You can link to situations from the flyover window for an operation instance with an open situation in the Interaction Detail portion of the view.

- You can display the metrics for the call relationships between operation aggregates and between operation instances.
- You can use the same search function, including additional support for locating groups by name.
- You can display historical data for the metrics. By default, the most recently completed monitoring interval is displayed, but you can use the Select Time Span dialog to specify a historical period.
- You can right-click on objects and select similar options from the displayed list, including the new option for **Manage Groups** (which is available only for operation aggregate objects).
- The flyover window for operation aggregates is the same, and includes the list of groups for which the operation aggregate is a member.
- You can continue to use the topology toolbar functions, including forward and back navigation arrows.
- You can select one or more objects in the topology view, then right-click and zoom in on the selected objects.

Specifically for the Operation Flow views and the Group Summary view, you can also select predefined situations from a list and associate them with the calculation of unavailability status when they occur on front-end services within a group.

Displaying unavailability indicators for operation aggregates and operation instances is also unique to the Operation Flow views.

When you double-click an operation aggregate in the Operation Flow view or right-click an operation aggregate in the view and select Show Interaction Detail, the *Interaction Detail* pane displays that portion of the topology in finer detail.

The Interaction Detail view displays only a subset of the operational flow. For the selected operation aggregate, the Interaction Detail view displays these items:

- Its operation instances
- The set of operation instances that call the operation
- The set of operation instances that it calls.

You can also display groups in the Group Summary view in table format by selecting the **View as Table** icon in the action bar at the top of the view. To switch back, select the **View as Topology** icon.

For more information about service-to-service topology, see Chapter 7, “Workspaces for service-to-service topology,” on page 71. For more information about operational flow topology views, see “Operational Flows workspace” on page 92.

Managing groups

You can create and configure groups from the Group Summary view or from any Operational Flow workspace.

Select one or more operation aggregates, right-click, and select **Manage Groups**. The Groups window opens. Use this window to create, edit, delete, and configure groups as required. The Manage Groups selection is not available if you select only unmanaged operations and unmanaged clients.

To display the Groups window from the Group Summary view, right-click a group and select **Manage Groups**.

You can also right-click anywhere in the empty space of the Group Summary view, in the Operation Flow for Service Group view, or Operation Flow for Process Group view and select **Manage Groups**.

The Groups window

The Groups window is where you create, edit and delete groups, display the operation aggregates that are already assigned to a group, and add available operation aggregates or remove them as required. To display the Groups window from the Group Summary view or from any operation flow view, select the Manage Groups option.

Figure 56 on page 175 shows an example of the Groups window. Within the Groups window is, the Groups table lists, in alphabetical order by name, all known service and process groups, and any text description for each group, if available. You can sort the name column in reverse order if wanted, and you can scroll the list to view all of the groups.

Displaying group details

To see the details of a group, select it in the Groups table. You can select only one group at a time.

The Group Detail section includes two lists of operation aggregate names:

- Operations for Group contains the names of operation aggregates that are already part of the selected group.
- Available Operations contains the names of all other known operation aggregates that you can select to add to the selected group.

Restriction: These objects cannot be members of a group and are not available for inclusion in the Available Operations list:

- Unmanaged clients
- Managed clients
- Unmanaged operations

Between these two lists are the **Add** and **Remove** buttons. Use these buttons to move operation aggregates between the two lists, and to control which operation aggregates are to be part of the selected group.

Figure 56 on page 175 displays an example, with the selected service group highlighted in the Service Groups list. The name of the selected service group is displayed again in the Group Name field in the Group Detail section, along with the full text description, if it is available. The operation aggregates that are already part of this service group are listed in the Operations for Group list. All other available operation aggregates are shown in the Available Operations list.

When you select the **Manage Groups** option from an Operational Flow view, any operation aggregates that you selected in that view are preselected in the appropriate list (either Available Operations or Operations for Group) when the Groups page is displayed. You can select additional operation aggregates in the Available Operations list to add to the group.

Creating a group

Initially, the Groups table is empty. You must create one or more groups and add them to this list. To create a group, complete these steps:

1. To the right of the Groups table, click **Create**.

The Create Group window is displayed, as shown in Figure 46.



Figure 46. The Create Group dialog

2. In the **Group Name** field, type a unique name for the group. It can be up to 64 characters in length. If the name exists, an error message is displayed, and you must type another name.
3. Optional: in the **Description** field, type a text description for the group. It can be up to 128 characters in length.
4. Select **Process Group** or **Service Group** from the **Group Type** list.
5. Click **OK** to save the group name, group type, and optional description. The new group name is added to the list of groups in the Groups table in the correct sorted location. Click **Cancel** to close the Create Group window without creating the group.

Editing an existing group

After creating a group, you can change the name of the group, change the group type, or modify the text description for the group. To edit a new group, complete these steps:

1. In the Groups table, select the group that you want to edit, for example, *LookupCustomerAll*.
2. Next to the Groups table, click **Edit**.

The Edit Group window is displayed as shown in Figure 47.

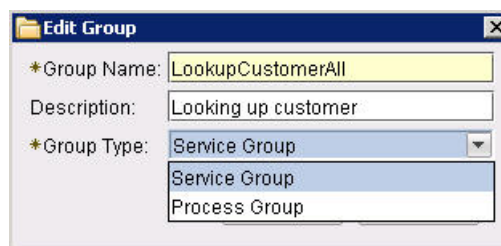


Figure 47. The Edit Group dialog

3. Modify the name of the group by typing over the name in the Group Name field. It must be a unique name, and it can be up to 64 characters in length. If the name exists, an error message is displayed, and you must type another name.
4. In the Description field, optionally type a text description or type over an existing description for the group. The text string can be up to 128 characters long.
5. You can change the group type by selecting Service Group or Process Group from the Group Type drop-down menu.

6. To save the group name, group type, and optional description, click **OK**. Alternatively, to close the Edit Group dialog without modifying the group name, group type, or description, click **Cancel**.

The modified name of the group is displayed in the list of groups in the Group Detail table, in the correct sorted location.

Adding operation aggregates to a service group

To add one or more operation aggregates to a service group, click **Add** to move operation aggregates from the Available Operations list into the Operations for Group list. To add operation aggregates to a service group, complete the following steps:

1. In the **Groups** table, select the service group that you want. The group name, type, and description are displayed in the **Group Detail** table.
2. If you previously selected a different group and changed it, you are prompted to save or discard those changes before continuing.
3. After you select the service group, scroll through the list of operation aggregates in the **Available Operations** list and select one or more names to be added to the service group. Optionally, to obtain more detail about a selected operation aggregate, you can display hover help information.
4. To move the selected operation aggregates into the **Operations for Group** list, click **Add**.
5. To remove operation aggregates from the service group, select one or more names in the **Operations for Group** list and click **Remove**. The names are moved out of the service group and into the **Available Operations** list.
6. Continue selecting operation aggregates as required. To manage the operation aggregates that are assigned to the service group, use **Add** and **Remove**.
7. When you are finished, to save your changes, click **OK**. To close the Groups window without modifying the service group, click **Cancel**.

Removing operation aggregates from a service group

You remove one or more operation aggregates from a service group by clicking **Remove** to move operation aggregates from the Operations for Group list into the Available Operations list. To remove operation aggregates from a service group, complete these steps:

1. In the groups table, select the group that you want.
2. If you previously selected a different group and changed it, you are prompted to save or discard those changes before continuing.
3. You select the service group, scroll through the list of assigned operation aggregates in the Operations for Group list and select one or more names. Optionally, to obtain more detail about a selected operation aggregate, you can display hover help information.
4. To move the selected operation aggregates into the Available Operations list, click **Remove**.
5. To add operation aggregates to the service group, select one or more names in the Available Operations list and clicking **Add** to move them into the Operations for Group list.
6. Continue selecting operation aggregates as needed and use **Add** and **Remove** to manage the operation aggregates that are assigned to the service group.
7. When you are finished, to save your changes, click **OK**, close the Groups window without modifying the service group, click **Cancel**.

Also, if you delete all of the operation instances from an operation aggregate, that operation aggregate is automatically removed from all groups to which it was assigned.

Defining what operations to monitor in a process group

To add one or more operations to a process group, click **Add** to move operations from the Available Operations tree into the Operations for Group tree. To add operations to a process group, complete these steps:

1. In the **Groups** table, select the process group. The group name is displayed in the **Group Detail** table name.
2. If you previously selected a different group and changed it, you must save or discard those changes before continuing.
3. After selecting the process group, view the list of operations in the Available Operations tree.

If business process management (BPM)-specific data is available for an operation, it is displayed as a child of a server/cluster, application, module, and component.

Server A software program or a computer that provides services to other software programs or other computers.

Cluster

A group of application servers that collaborate for the purposes of workload balancing and failover.

Application

One or more computer programs or software components that provide a function in direct support of a specific business process or processes.

Module

A module is a WebSphere Business Integration project that is used for development, version management, organizing resources, and deploying to WebSphere Process Server.

Component

Components are the parts of the module that are the actual services. On the inside, they are implemented using implementation types, which include business processes, state machines, human tasks, and others. The WebSphere Integration Developer tools generate implementations that the assembly editor can use for components. The structure of the component has the following parts:

- An implementation
- Optionally, one or more interfaces
- Optionally, one or more partner references

Business Process Definition (BPD)

A model of a business process workflow, consisting of individual activities. The activities can depend on user input. BPDs are created and edited in the Process Designer, an Eclipse-based integrated development environment. BPDs are in the hierarchy under Applications, independent of Modules and Components. After you start monitoring an application server for the first time, any BPD on the server is displayed in the hierarchy only after the BPD is called.

If no BPM-specific data is available for an operation, it is displayed as a child of a virtual parent application node, named SOA 7.1 Operations. For example, if you have the current version of Tivoli Enterprise Portal and an older version of Data Collector (version 7.1.1 or earlier), the data collector does not provide

BPM-specific data about the application structure. In this case, the monitored operations are displayed as children of the virtual application, SOA 7.1 Operations.

A hierarchical tree structure, as shown in Figure 48, displays the servers/clusters, applications, modules, and components that a particular operation belongs to.

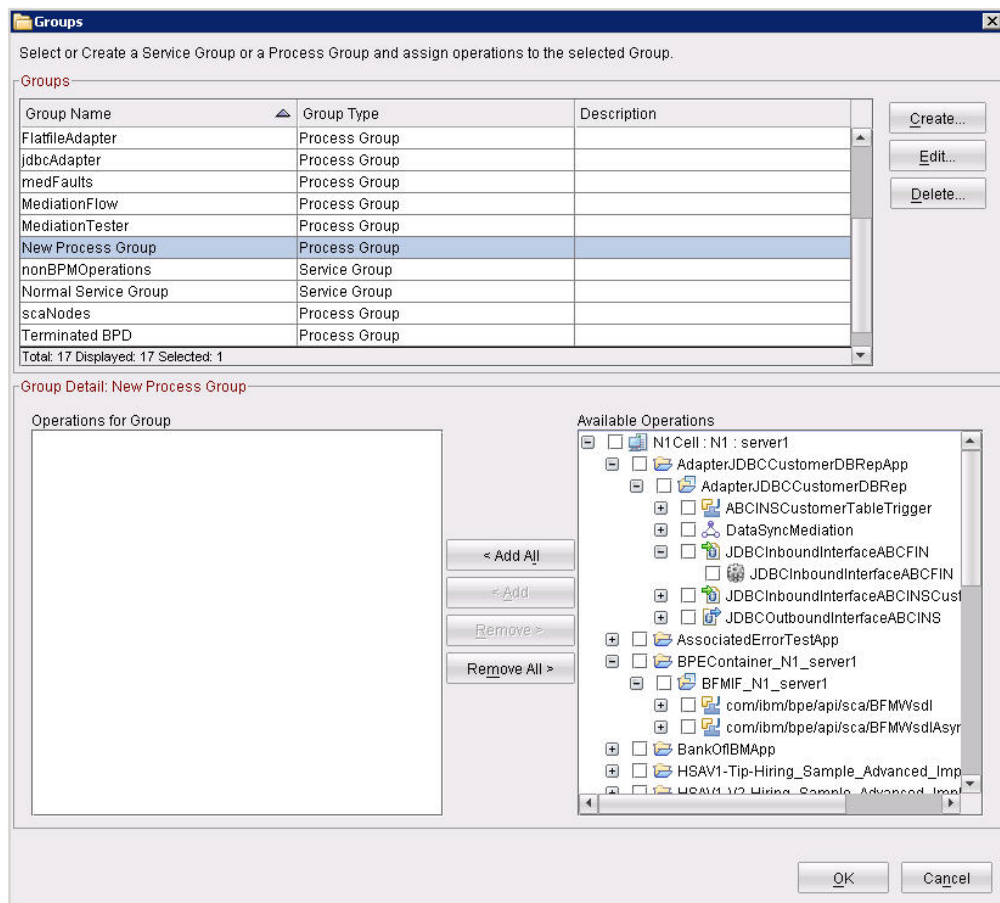


Figure 48. The Available Operations tree for process groups

Select one or more nodes that are to be added to the process group, for example, the ITCAMTest01 module node.

Standard operation icons differentiate each element in the hierarchical tree.

Table 28 shows the most common icons.

Table 28. Tree icons












Icon	Description
	Server/cluster on which the application is deployed.
	Application

Table 28. Tree icons (continued)

Icon	Description
	Module
	Untyped component (the default component icon shown if the component type is unknown)
	Process component
	Human task component
	State machine component
	Java component
	Mediation flow component
	Rule group component
	BPD component

- To move the selected nodes into the Operations for Group tree, click **Add**. The added nodes are disabled in the Available Operations tree and are displayed in the Operations for Group tree, as shown in Figure 49 on page 170.

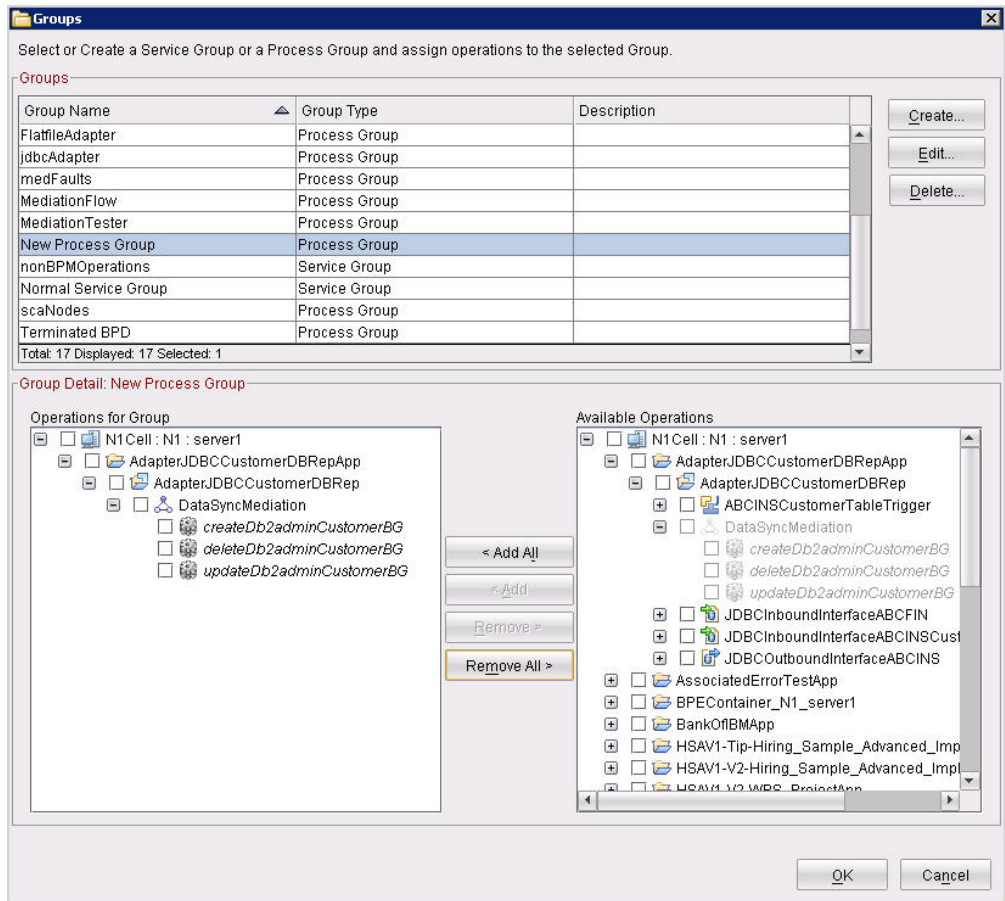


Figure 49. Nodes added to the Operations for Group tree

Monitoring now and in the future

Operation selection applies to currently known elements and elements that will be monitored in the future.

Adding multiple operations by using elements

You can monitor multiple operations by adding one or more parent elements from the Available Operations tree. All child elements are indirectly added to the process group. In this example, selecting the ITCAMTest01 module indirectly selects the child elements: Process01, operation1, operation2, and operation3. A gray square indicates that a parent node contains one or more selected child elements. Indirectly selected nodes are indicated by italicized text, as shown in Figure 49. This method is not available for service groups, where you can only select single operations.

Clusters

A cluster is a group of servers. You can monitor the cluster group as a whole. However, you cannot select an element for only one cluster member. For example, if a user selects a module in a cluster, then all child elements on all cluster members are indirectly selected for the process group.

- To remove operations from the process group, select one or more nodes in the **Operations for Group** tree and click **Remove**. The nodes are moved out of the process group and into the Available Operations tree.

6. Continue selecting nodes as needed. To manage the operations that are assigned to the process group, use **Add** and **Remove**.
7. When you are finished, to save your changes, click **OK**. To close the Groups window without modifying the process group, click **Cancel**.

Removing operations from a process group

You can remove one or more operations from a process group by clicking **Remove** to move operations from the Operations for Group tree into the Available Operations tree. To remove operations from a process group, complete these steps:

1. Select the wanted process group in the Groups table. The group name is displayed in the Group Detail table name.
2. If you previously selected a different group and changed it, you must save or discard those changes before continuing.
3. After selecting the wanted process group, view the tree of assigned operations in the Operations for Group tree and select one or more nodes, as shown in Figure 50.

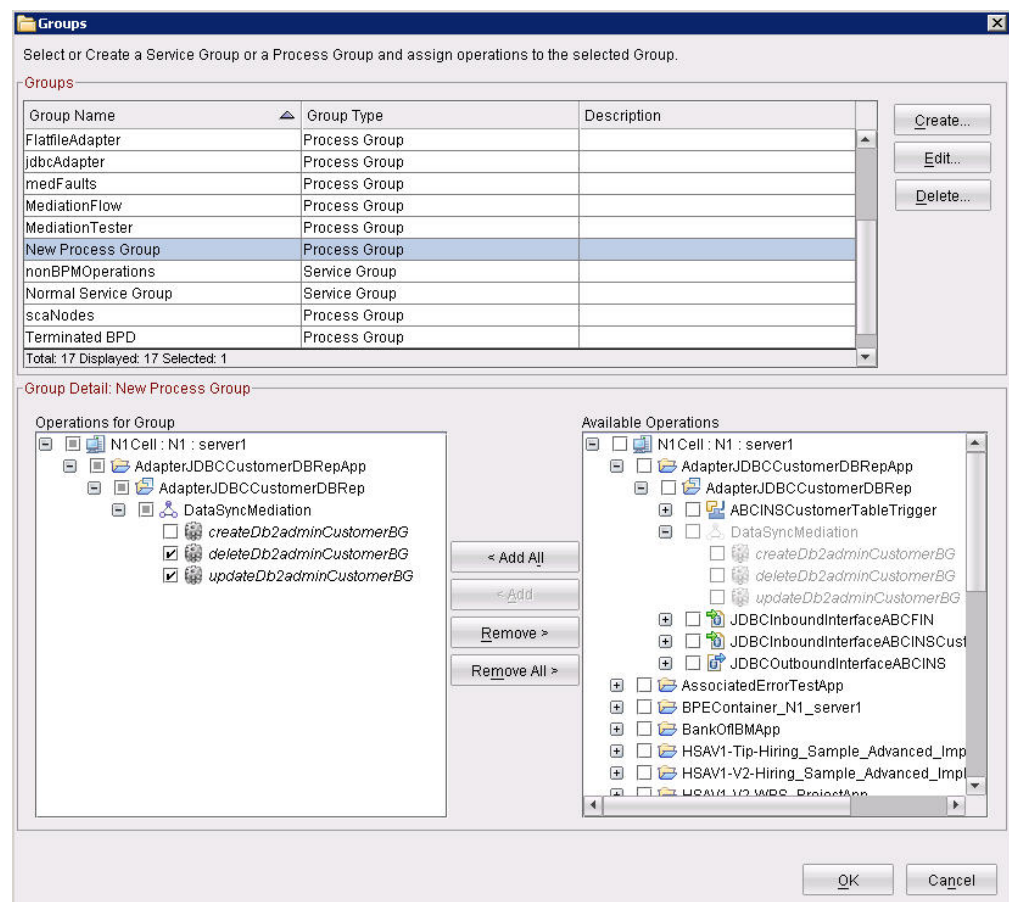


Figure 50. Nodes selected for removal from the process group

4. To move the selected operations from the Operations for Group tree, click **Remove**. The removed nodes are enabled in the Available Operations tree. Removing nodes from a process group changes the way other elements are selected. In this example, the ITCAMTest01 module was added to the process group. This selection indirectly added the Process01, operation1, operation2, and operation3 child elements to the process group. When the operation2

element, which is currently indirectly selected, is removed from the process group, operation1 and operation3 are now directly selected. The status of parent elements also changes. For example, ITCAMTest01 and Process01 are no longer directly selected.

5. You can also add operations to the process group by selecting one or more nodes in the Available Operations tree and clicking **Add** to move them into the Operations for Group tree.
6. Continue selecting nodes as needed and use **Add** and **Remove** to manage the operations that are assigned to the process group.
7. When you are finished, to save your changes, click **OK**. Alternatively, to close the Groups window without modifying the process group, click **Cancel**.

Also, if you delete all of the operation instances from an operation aggregate, that operation aggregate is automatically removed from all groups to which it was assigned.

Deleting a group

To delete a group, complete these steps:

1. In the Groups table, select the group that you want to delete, for example, *Account Processing*.
2. To the right of the Groups table, click **Delete**.
3. You are prompted to confirm that you want to delete the selected group.
4. To delete the group, click **Yes**. Alternatively, to cancel the request, click **No**.

If you clicked **Yes**, the group is removed from the Groups table, and the table is refreshed in sorted order.

An example: Creating a service group

Figure 51 displays a simple service-to-service topology view of a sample application that looks up a customer name in a database. You can select one or more of these operation aggregates and assign them to a service group, and then track their overall health as a group.

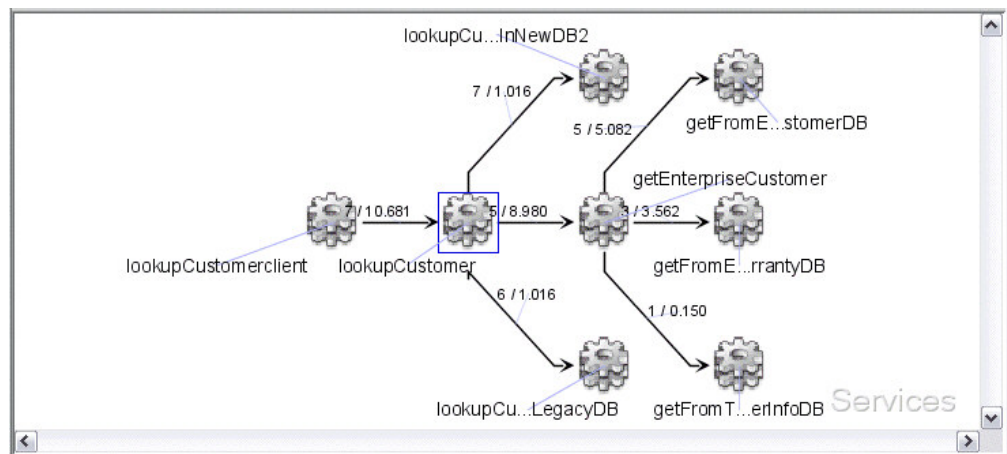


Figure 51. Sample lookupCustomer application topology view

To assign one or more of these operation aggregates to a service group, complete these steps:

1. In the service-to-service topology view, select the operation aggregates that you want to include in the service group.

2. Right-click and select **Manage Groups**, as shown in Figure 52.

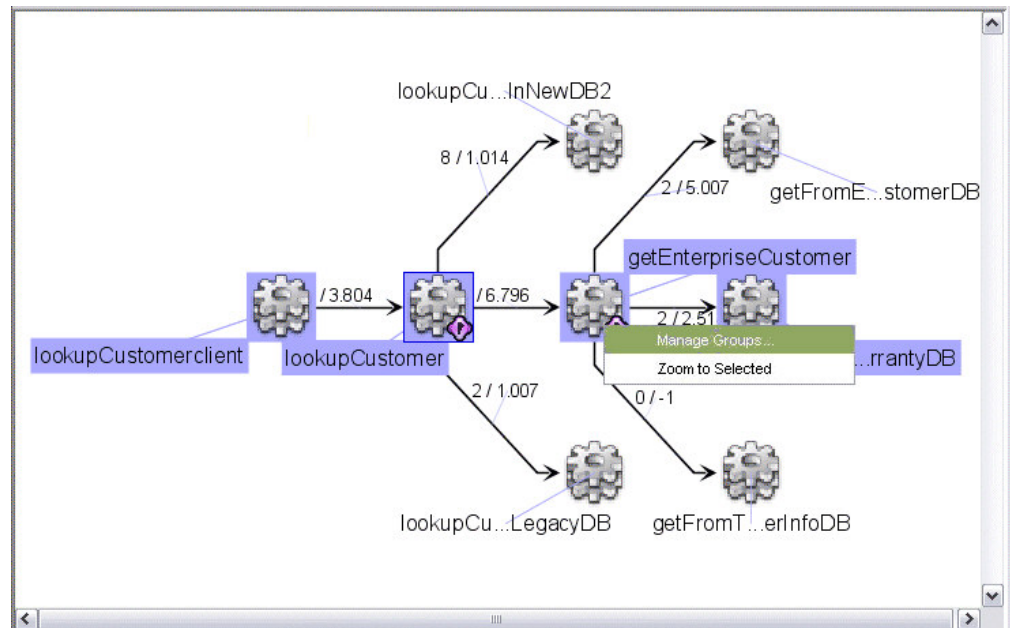


Figure 52. Selecting several operation aggregates to include in a service group

3. The Groups window is displayed, showing any service or process groups that were already created. For this example, because no groups were previously created, the Groups window is empty, as shown in Figure 53. To create a service or process group, in the Groups window, click **Create**.

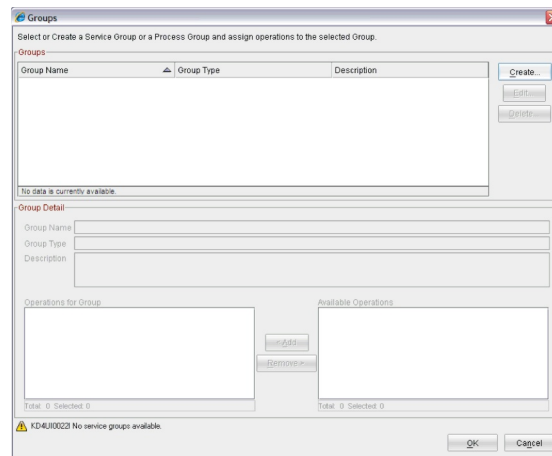


Figure 53. Creating a service group in the Groups window

4. In the Create Group dialog, type a name for the new group. Then, select the group type (service or process), and type an optional text description, similar to the example in Figure 54 on page 174. To continue, click **OK**.



Figure 54. Naming and describing your new service group

5. You are returned to the Groups window, where your new service group, *MyServiceGroup*, is now included in the Groups area at the top of the window. Selecting this group in the Groups area causes the details (name, type, and description) of the service group to be displayed in the Group Detail area in the center of the window. In the lower part of the window, seen in Figure 55, the Available Operations list displays all of the operation aggregates that are available for inclusion in the group. The operation aggregates that you previously selected in the service-to-service topology views are also highlighted in this list, and are preselected to be added to the group.

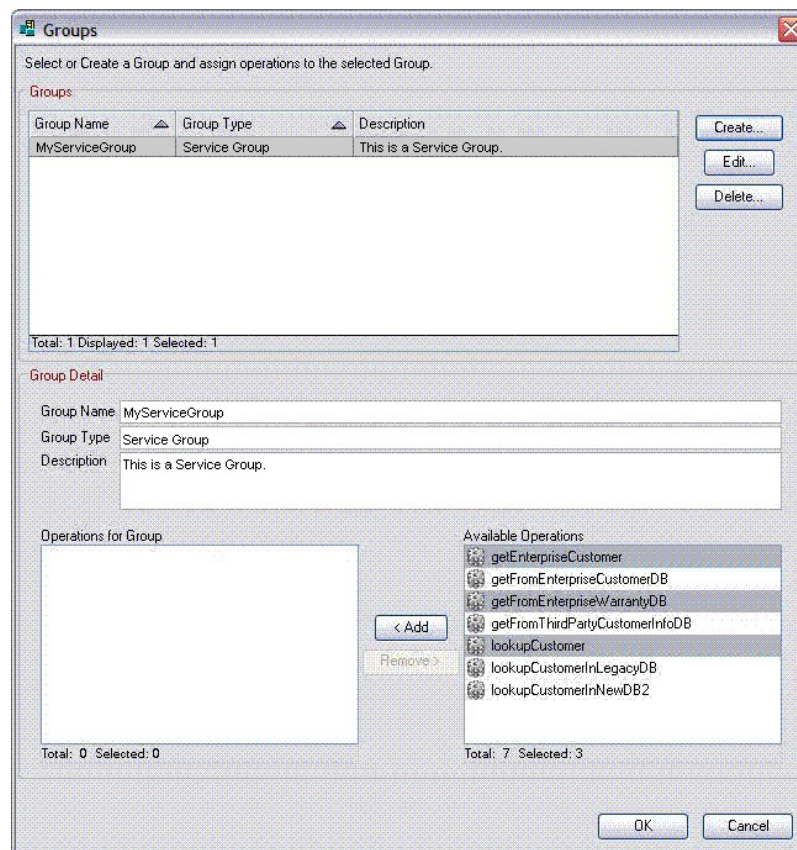


Figure 55. Highlighted operation aggregates are preselected for you in the Available Operations list

6. To move the selected operation aggregates from the Available Operations list into the Operations for Group list, click **Add**, as shown in Figure 56 on page 175.

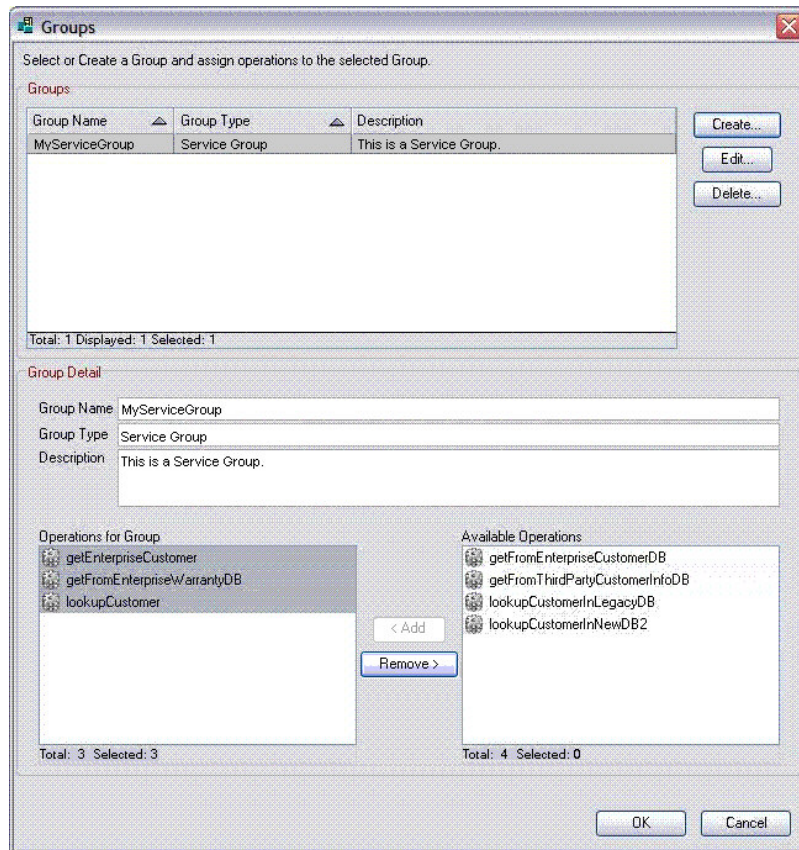


Figure 56. Adding selected operation aggregates to the service group

7. You now have two choices: Continue to select additional operation aggregates from the Available Operations list and add them to the service group, or remove operation aggregates from the service group by selecting them in the Operations for Group list and clicking **Remove** to move them back to the Available Operations list. When you finish assigning operation aggregates to the service group, to close the Groups window, click **OK**.

When you refresh the Operational Flow workspace, you can move the cursor over one of the operation aggregates in the view and display the updated flyover window, which now includes the identity of the service group to which the operation aggregate is assigned, as shown in Figure 57 on page 176.

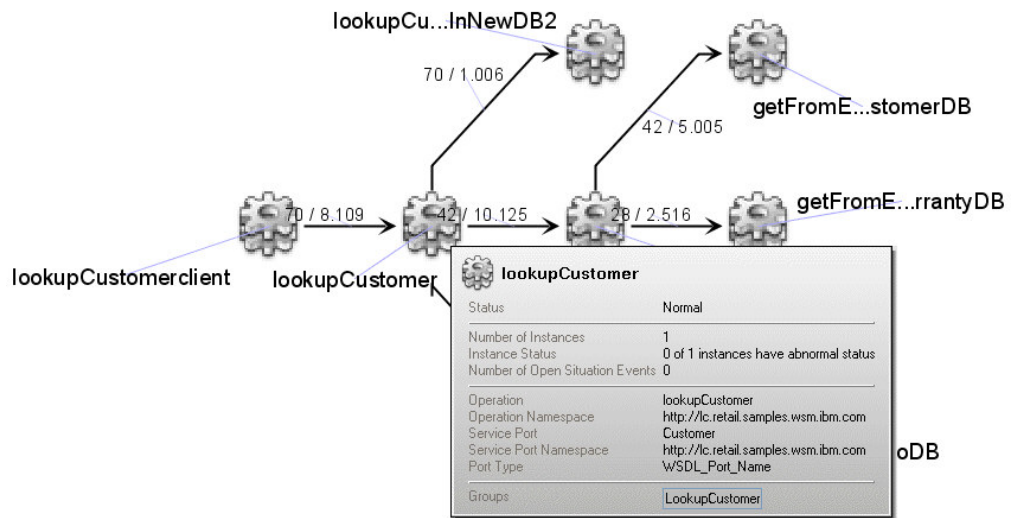


Figure 57. Displaying service group assignments in the flyover window for a selected operation aggregate

If you want to create a second service group and include all of the operation aggregates, including the ones that are already assigned to the first service group, repeating this procedure, you can highlight one or more operation aggregates in the view. Right-click and select **Manage Groups**. Then, create a second service group name. To add to the new group, select the operation aggregates from the available list. Figure 58 on page 177 displays the result of creating a service group, *LookupCustomerAll*, which is now added to the list of service groups in the Group area.

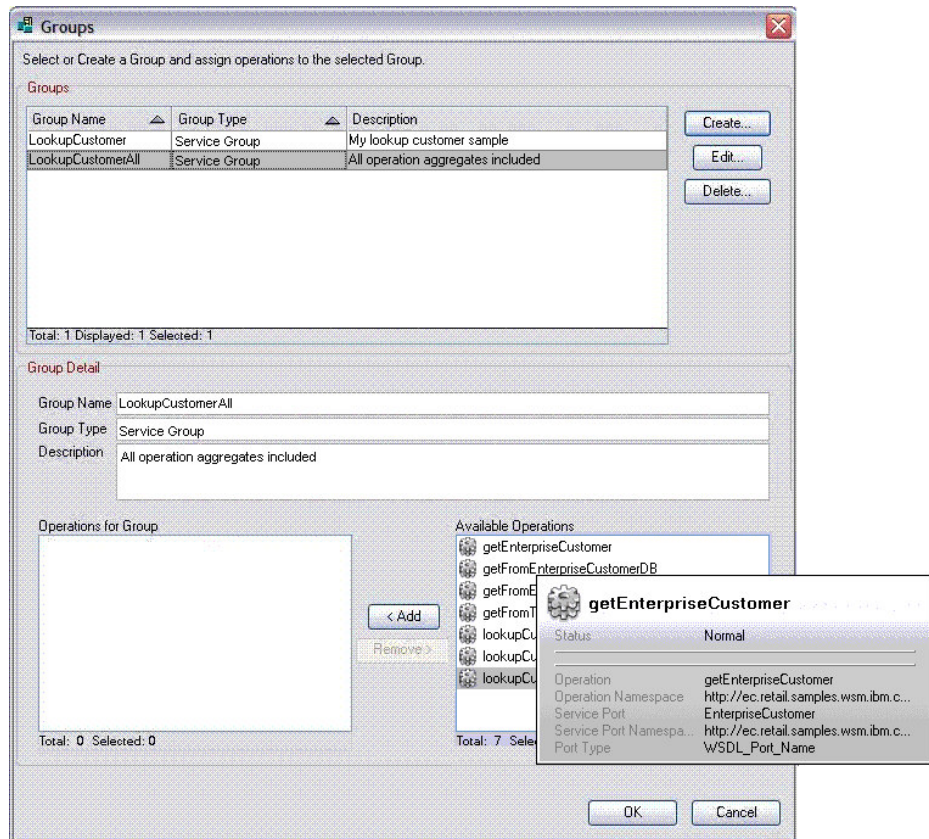


Figure 58. Adding a second service group

In Figure 58, note how you can use flyover windows for operation aggregates to provides you with more details about the operation aggregates and groups. Note in this example that the one operation aggregate that you selected in the topology view is already highlighted.

To add an operation aggregate to the service group, you can select each one in turn and click **Add**. Alternatively, you can highlight all of the operation aggregates at the same time and add them in one step. After adding all of the operation aggregates to the new service group, to close the Groups window, click **OK**.

When you refresh the Operation Flow - All Flows view, you can display the updated flyover windows for operation aggregates that were added to both service groups, and see both service group names (in our example, *LookupCustomer* and *LookupCustomerAll*) displayed in the help information, as shown in Figure 59 on page 178.

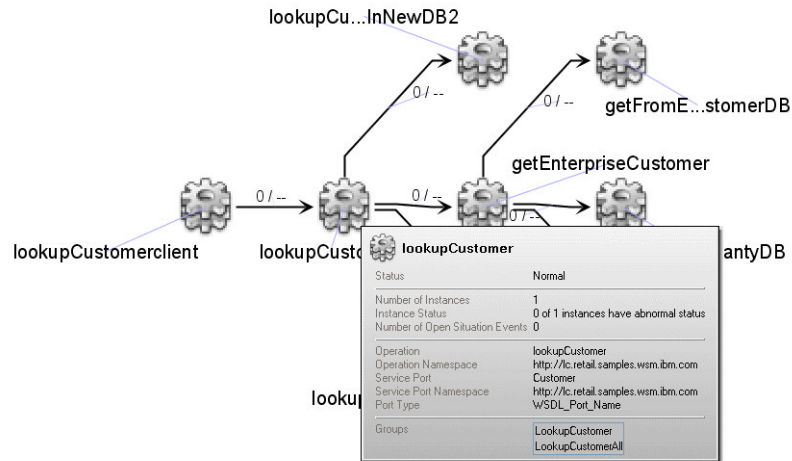


Figure 59. Operation aggregates assigned to multiple service groups

These examples give you an overview of how you can create service groups that represent applications or business processes, and then monitor their overall health in a high-level summary view, either in its own default workspace or in a view that you can include in any Tivoli Enterprise Portal workspace. The sections that follow provide additional information about how to manage your service groups.

Displaying groups in the Group Summary workspace: You can display defined groups in the Group Summary workspace. At the top of the Navigator Physical view in the Tivoli Enterprise Portal application window, locate the **View:** list. From the **View:** list, select the **ITCAM for SOA Navigator** view. The Group Summary workspace is displayed in the same way as the example that is shown in Figure 60.

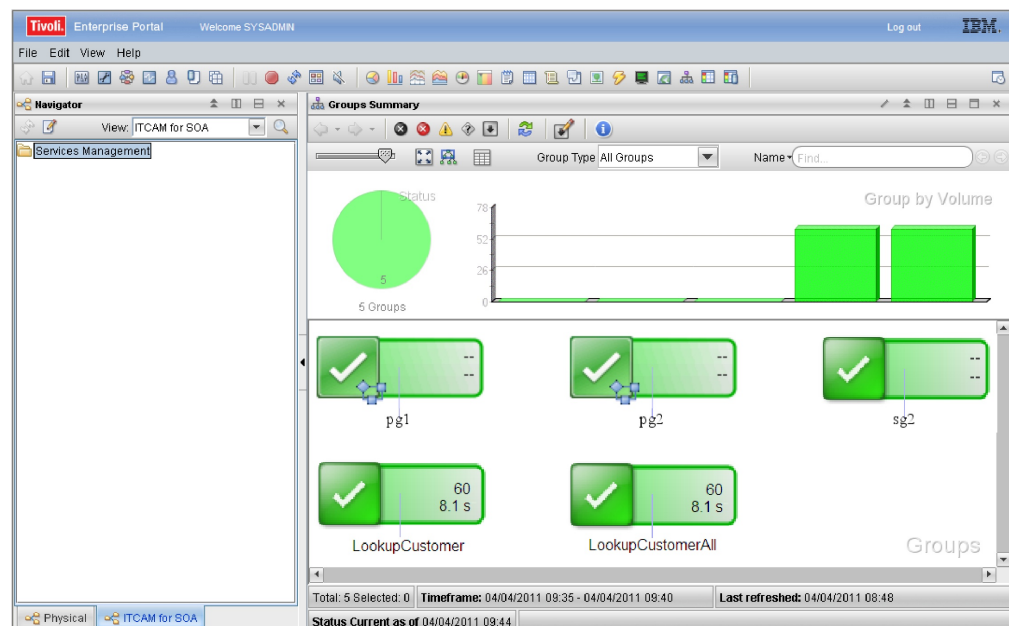


Figure 60. The Group Summary workspace displays the groups

If the **ITCAM for SOA Navigator** view is not available for selection in the **View:** list, you might have to configure your user ID to access this view. For information

about configuring your user ID to access the **ITCAM for SOA** Navigator view, see “Accessing the Navigator ITCAM for SOA view” on page 26.

If the **ITCAM for SOA** tab is displayed at the bottom of the Navigator view, you can also click that tab to switch to the customized Navigator ITCAM for SOA view. To switch back to the Navigator Physical view, click the **Physical** tab.

Each group that you defined is shown in the Group Summary view by name. The overall health of the group is represented by the corresponding status graphic, and each group displays summary metric statistics on overall performance (response time) and message volume. Other indicators show you at a glance the calculated group status and unavailability.

From this view you can configure unavailability situations, and you can add additional groups or modify groups.

Along the top of the Group Summary view above the groups, a vertical bar chart displays another summary view of the overall health of each group. Each group is represented by a vertical bar in the chart. Each bar in this vertical bar chart displays the volume of observed messages for its associated group during the monitoring interval. You can configure this chart to display performance instead of volume by using the View Properties function (for more information, see “View properties” on page 157).

To the left of the vertical bar chart is a pie chart that shows a summary of the status of all groups.

From the Group Summary view, you can display the flyover window by holding the cursor over a group in the view, as shown in Figure 61.

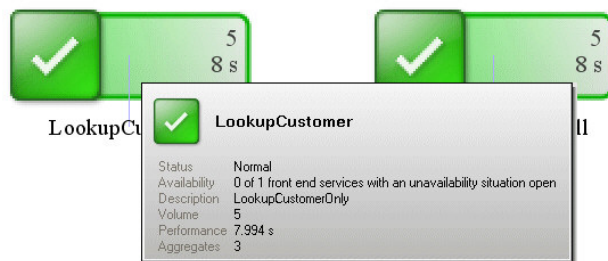


Figure 61. Displaying the flyover window for a group in the Group Summary view

From the Group Summary view, you can also display the service-to-service topology detail associated with the group, by either double-clicking the group or right-clicking the group to highlight it, and then selecting Show Service Group Topology for service groups or Show Process Group Topology for process groups. For more information about these views, see “Displaying the topology view for a group” on page 161

Configuring for unavailability

The unavailability of a group is based on situations that are associated with the front-end services of the group. These customized situations typically use the service unavailability attributes and metrics that are provided with this version of ITCAM for SOA.

Familiarize yourself with the special attributes that you can include in situations for determining unavailability. For information and examples about including these attributes in provider-side situations for determining the unavailability of operation aggregates and operation instances, see “Measuring service unavailability” on page 214 (The term *provider-side* refers to the service type of *Provider* for a situation defined using attributes from the Services Inventory and Services Inventory for Requester ID tables).

The unavailability attributes that are derived from the metric data determine whether to trigger the situation, based on your customized thresholds. To be included in the determination of whether the group is unavailable, these customized situations must be triggered for a front-end service of the group.

Configuring the list of unavailability situations

After you create one or more customized situations based on these unavailability attributes, to identify these situations as unavailability situations for the group, in the toolbar of either the Group Summary view or the Operation Flow views, click **Configure Unavailability**. Figure 62 displays an example of clicking **Configure Unavailability** from the Group Summary view.

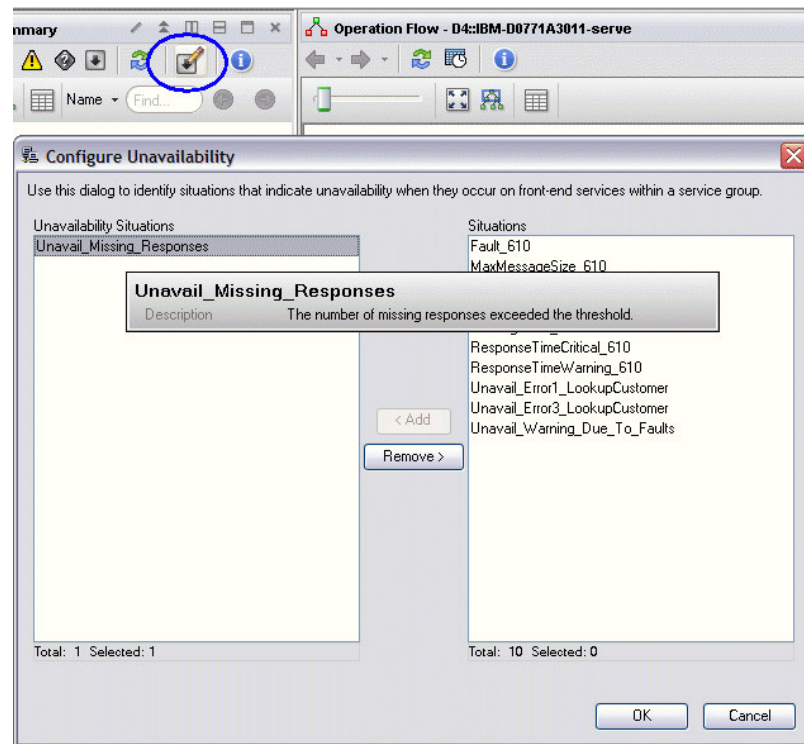


Figure 62. Identifying a situation as an unavailability situation.

In the Configure Unavailability window, the list of predefined situations that you can select as an unavailability situation is shown in the Situations list. You can

select one or more situations from this list. Then, to move them into the Unavailability Situations list, click **Add**. In the same way, to remove situations from the Unavailability Situations list, click **Remove**. Figure 62 on page 180 displays the user-defined situation, *Unavail_Missing_Responses* after it is selected and added to the list of unavailability situations. Optionally, you can display the text description of the situation in the hover help as shown in this example.

Renaming an unavailability situation: If you configure a situation for unavailability and then change the name of the situation, when you open the Configure Unavailability dialog, you might see the situation displayed in the Situations list (as if it was not configured for unavailability) with the new name. If you do not close the dialog, this condition might remain in effect for up to 2.5 minutes, until the next time the SOA Domain Management Server polls its information and updates the situation name. Similarly, if the situation was triggered, it retains the previous situation name for up to 2.5 minutes, until after the next poll of the SOA Domain Management Server.

Important: You can also customize the impact of situations on the status and unavailability of a particular group. See “Changing the situation impact on status of group” on page 159.

Displaying unavailability

A service or process group is displayed as unavailable when at least one unavailability situation is triggered for a front-end service of the group. A special indicator (a gray square with a downward pointing arrow) is displayed on the group, indicating that it is unavailable. Figure 63 on page 182 displays a customized Operational Flow workspace that includes the Group Summary view and has one service group, *LookupCustomerAll*, defined. The *Unavail_Missing_Responses* situation was triggered against the front-end service, *LookupCustomer*, as shown in the Operation Flow topology view. The optional hover help information for the group is also displayed, indicating the unavailability state of the front-end service, and the name of the unavailability situation that is open.

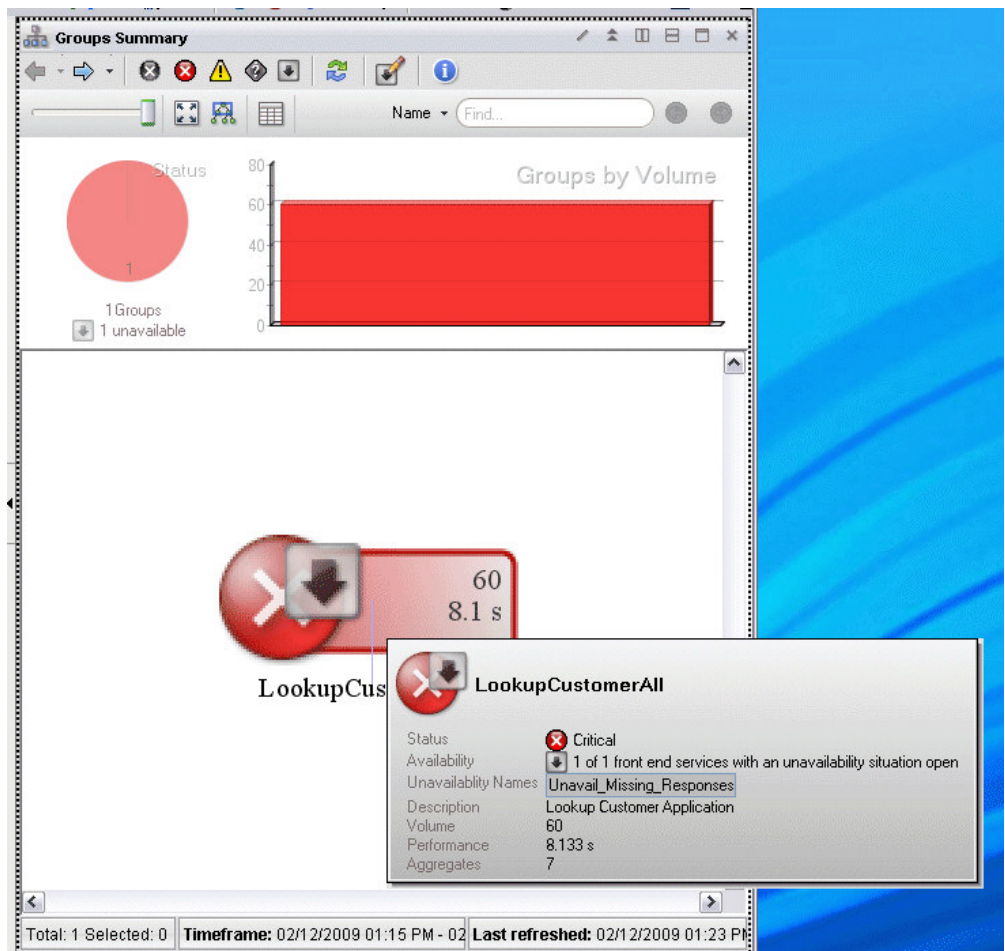


Figure 63. Displaying a group as unavailable.

The Operation Flow for Service Group view for this service group can also be displayed, as shown in Figure 64 on page 183. The *LookupCustomer* operation aggregate also displays the unavailability indicator, and when you display the operation instance information in the Interaction Detail portion of the view, the specific operation instance that is associated with the open situation also displays the unavailability indicator.

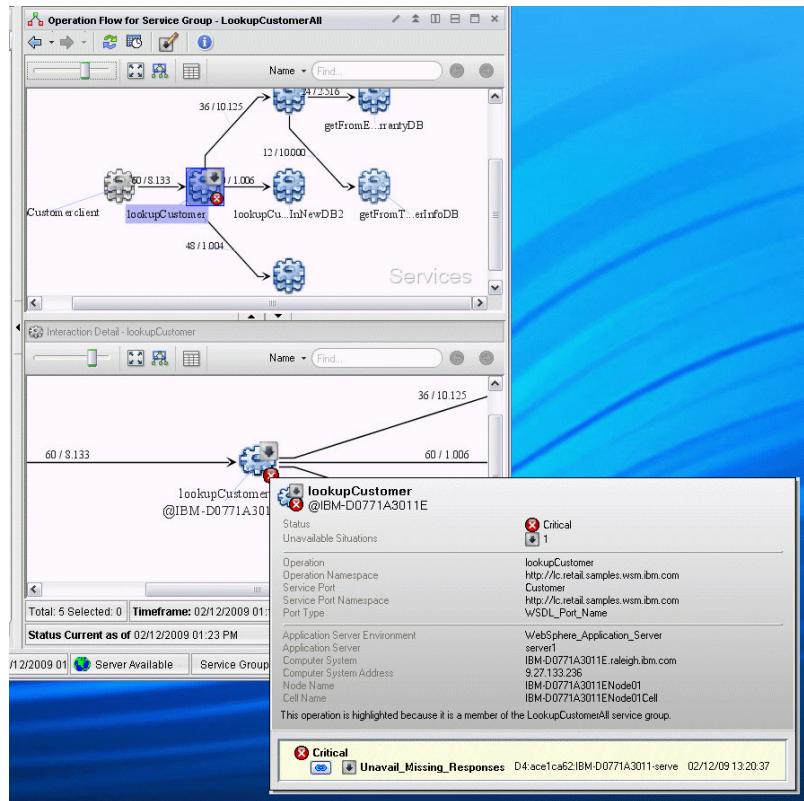


Figure 64. Displaying the unavailability state in the service group topology view.

You can also display unavailability in the table mode display of the Operation Flow for Service Group view. An example is shown in Figure 65.

Resource	Name	Operation	Operation Namespace	Service	Service Port	Status	Availability	Frontend Service	Notes
<no filter>	<no filter>	<no filter>	<no filter>	<no filter>	<no filter>	<no filter>	<no filter>	<no filter>	<no filter>
lookupCustomerClient	lookupCustomerClient	lookup...	http://...	Custo...	http://...				
lookupCustomerInLegacyDB	lookupCustomerInLegacyDB	lookup...	http://...	Legacy	http://...				
lookupCustomerInNewDB2	lookupCustomerInNewDB2	lookup...	http://...	NewD...	http://...				
getEnterpriseCustomer	getEnterpriseCustomer	getEnt...	http://...	Enterp...	http://...				
getFromEnterpriseCustomerDB	getFromEnterpriseCustomerDB	getFro...	http://...	Enterp...	http://...				
getFromEnterpriseWarrantyDB	getFromEnterpriseWarrantyDB	getFro...	http://...	Enterp...	http://...				
lookupCustomer	lookupCustomer	lookup...	http://...	Custo...	http://...	Critical	Unavailable	Frontend Service	
getFromThirdPartyCustomerInfoDB	getFromThirdPartyCustomerInfoDB	getFro...	http://...	ThirdP...	http://...				
calls									
calls									
calls									
calls									
calls									
calls									
calls									
Total: 15 Displayed: 15 Selected: 0									

Figure 65. Displaying the unavailability state in table mode.

Offline behavior: The unavailability indicator is not displayed for a group, or for its associated operation aggregates, or operation instances if both of the following conditions are met:

- An unavailability situation is open for its front-end service operation instance, but the subnode for the operation instance went offline.
- The situation is not open for any other operation instance of that front-end service.

Chapter 9. Creating custom workspaces and links

In addition to the predefined workspaces and links that are provided with ITCAM for SOA, you might want to create your own customized workspaces and links.

For details on creating workspaces and links, see the help information and documentation for Tivoli Enterprise Portal. The following additional information applies to ITCAM for SOA workspaces and links.

Creating your own workspaces: You cannot directly modify or delete the predefined workspaces that are provided with this product. However, you can make new workspaces by editing a predefined workspace and saving the changes using a different workspace name.

Workspace Links are not copied: Workspace links are associated with a specific view on a specific workspace. If you copy a workspace to customize it and create a workspace, you must re-create any workspace links that you want to use on the new workspace, using the link definitions from the original workspace as a model.

Link symbols for the Operation Flow view

When linking to the Operation Flow view, you must provide context using a set of link symbols that the view recognizes by name. In order to pass in the contextual information that is needed to populate the topology view, your workspace link definition must include these symbols, along with appropriate values. The link symbols that are supported by ITCAM for SOA are defined in Table 29.

Table 29. Link symbols for Operation Flow views

Link symbol name	Description
ServicePortName	The Service Port Name is used when identifying the service port and operation pair whose operational flow is displayed in the topology view. This link symbol is used only when the target of the link is the Operational Flow for Operation workspace. If this link symbol is specified, then AppServerEnvironment, ServicePortNamespace, OperationName, and OperationNamespace must also be specified.
ServicePortNamespace	The Service Port Namespace is used when identifying the service port and operation pair whose operational flow is displayed in the topology view. This link symbol is used only when the target of the link is the Operational Flow for Operation workspace. If this link symbol is specified, then AppServerEnvironment, ServicePortName, OperationName, and OperationNamespace must also be specified.
OperationName	The Operation Name is used when identifying the service port and operation pair whose operational flow is displayed in the topology view. This link symbol is used only when the target of the link is the Operational Flow for Operation workspace. If this link symbol is specified, then AppServerEnvironment, ServicePortName, ServicePortNamespace, and OperationNamespace must also be specified.

Table 29. Link symbols for Operation Flow views (continued)

Link symbol name	Description
OperationNamespace	The Operation Namespace is used when identifying the service port and operation pair whose operational flow is displayed in the topology view. This link symbol is used only when the target of the link is the Operational Flow for Operation workspace. If this link symbol is specified, then AppServerEnvironment, ServicePortName, ServicePortNamespace, and OperationName must also be specified.
AppServerEnvironment	The Application Server Environment value is used in the Services Inventory table to identify the type of application server runtime environment that is being monitored by the ITCAM for SOA data collector that observed a service port and operation pair. This value is used to derive the mediation type information. The mediation type information is required if the service port and operation pair is a mediation operation (for example, an SCA mediation, WebSphere Message Broker mediation, or DataPower mediation), but the Application Server Environment value is sufficient to determine the mediation type information when required. This symbol is used only when the target of the link is the Operational Flow for Operation workspace. If this link symbol is specified, then ServicePortName, ServicePortNamespace, OperationName, and OperationNamespace must also be specified.
OpFlowContext	<p>This value is used only when the target of the workspace link is either the Operational Flow for Operation workspace or the Operational Flow for Application Server workspace. If you leave this workspace and then later return to the workspace using the Tivoli Enterprise Portal console history <i>back</i> button, this link symbol value ensures that the state of the topology view for that particular linked context is restored.</p> <p>When the target of the workspace link is the Operational Flow for Operation workspace, using the CALL function in the Link Expression Editor, specify the value for this link symbol:</p> <pre>CALL("com.ibm.management.soa.tepui.utils.SCPLinkUtil", "getOpFlowContext", null)</pre> <p>When the target of the workspace link is the Operational Flow for Application Server workspace, specify the value for this link symbol using this format:</p> <pre>CALL("com.ibm.management.soa.tepui.utils.SCPLinkUtil", "getAppOpFlowContext", null)</pre>

When these link symbol names are displayed in the Link Expression Editor, they are surrounded by \$ characters (for example, \$OperationName\$).

Important: Ensure that link symbol names are spelled correctly. If your link definition does not work, return to this link definition and verify that the link symbols that you added are all spelled correctly with correct values and expressions.

Building your own views

ITCAM for SOA provides data through a number of tables that you can use to build your own views.

For example, if you want to see a historical view of a particular web service operation response time, you can create a plot chart using the same performance summary query, but using a filter for the specific operation of interest. To see the individual metric data collected by the monitoring agent, create a view based on the Services Metric table data (for more information, see “Services Message Metric_610 attributes” on page 282).

The Services Message Metric attributes contains a significant amount of data. Create the query such that a limited number of rows are returned, with only the fields that you need. For example, return only those rows for a specific combination of service port name and operation name, with a response time greater than a particular value.

Linking to a new workspace containing an Operational Flow for Operation topology view

This section describes an example of how you might create a dynamic workspace link to a new workspace that contains an Operational Flow for Operation topology view. A similar procedure can be used for an Operational Flow for Application Server topology view.

This new workspace must meet the following criteria:

- The new workspace must be defined as a secondary workspace on the Performance Summary node of the Navigator Physical view.
- The new workspace must be defined as a target of a workspace link.
- To populate the topology view with data, the workspace link definition to the new workspace must include a specific set of link symbols that are supported by ITCAM for SOA, that provide the context information for the service port and operation pair. This context information is used in the query that populates the topology view. This procedure illustrates how these link symbols are specified in the link definition.

Creating the workspace

The new workspace must be defined as a secondary workspace on the Performance Summary node of the Navigator Physical view. You can create this workspace by completing these steps:

1. From a Tivoli Enterprise Portal desktop client or browser client, go to the desired Services Management Agent Environment node in the Navigator Physical view, and select the Performance Summary node. The Performance Summary workspace is displayed.
2. Optionally, delete one or more views from this workspace.
3. To create an empty view in the workspace, in the upper right corner of the Navigator Physical view, select either the **Split horizontally** or **Split vertically** icons.

Figure 66 on page 188 displays an example of a new workspace created by modifying the Performance Summary workspace, first removing the Average Response Time by Operation view and then splitting the Navigator Physical view vertically to create a new empty view in the workspace.

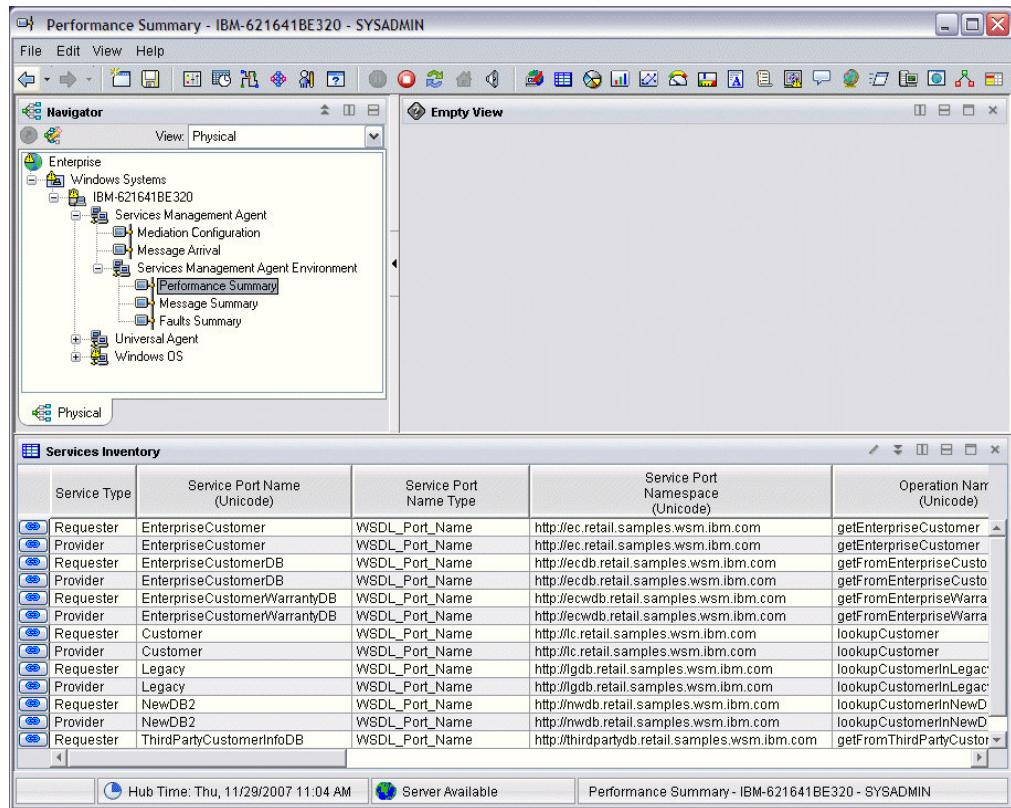


Figure 66. Creating a view in the new workspace

The new workspace is not yet saved or defined as the target of a workspace link. If you attempt to drag an Operational Flow topology view into this new empty view space, using the Topology icon in the workspace toolbar, a generic Topology view is added to the view space. However, data cannot be populated in the view because the workspace must first be defined as the target of a workspace link, and the link definition must include the context for the query using the supported link symbols. Figure 67 on page 189 shows the message that is displayed in this case, reminding you of these requirements before the topology view can be populated with data.

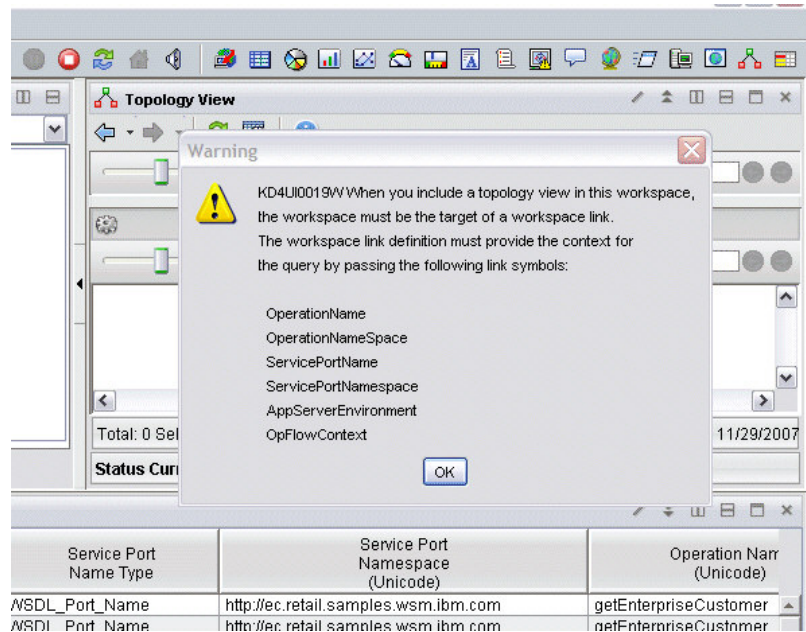


Figure 67. Warning message displayed when adding a topology view before defining the workspace as the target of a workspace link

4. Save the new workspace (select **File** → **Save Workspace As** in the menu bar at the top of the new workspace) to a unique name. For this example, the workspace is saved as *New Topology Workspace*. Under **Workspace Options**, select the check box for **Only selectable as the target of a Workspace Link**.

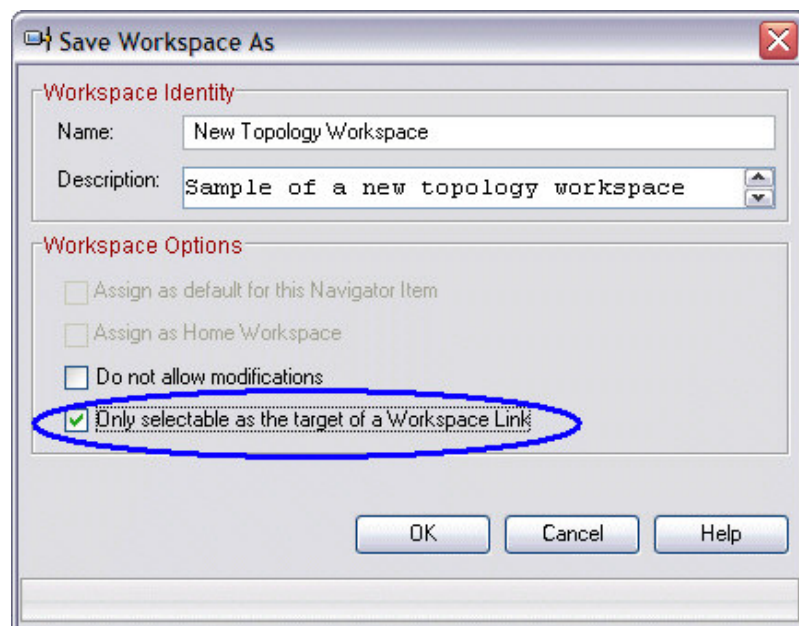


Figure 68. Saving the new workspace and defining it as the target of a workspace link

Defining the dynamic workspace link

After creating and saving the new workspace and defining it as the target of a workspace link, use the Workspace Link Wizard to define the link and add the

required link symbols that define the context for the query to the SOA Domain Management Server to populate the topology view with data.

Define the workspace link for your new workspace by completing these steps:

1. In the Tivoli Enterprise Portal, navigate to the Performance Summary workspace under the monitored Services Management Agent Environment node in the Navigator Physical view.
2. In the Services Inventory table view, select the workspace link icon and select **Link Wizard**, or right-click on a row in the table and select **Link To -> Link Wizard**. The **Workspace Link Wizard - Welcome** page is displayed.
3. Select the radio button to **Create a new link**.
4. Click **Next**. The **Workspace Link Wizard - Link Name** page is displayed.
5. Type the name and a description for the workspace link. This is the name that is displayed in the **Link To** menu when you want to select this workspace link. For purposes of this example, the name is specified as *My New Workspace Link*.
6. Click **Next**. The **Workspace Link Wizard - Link Type** page is displayed.
7. Select the radio button for the **Dynamic** link type.
8. Click **Next**. The **Workspace Link Wizard - Target Workspace** page is displayed. Verify that the Navigator View field at the top of the page points to the *Physical* view.
9. To display the Performance Summary node link that is associated with your newly created secondary workspace, expand the Navigator node tree on the left.
10. To highlight it, click the Performance Summary node. The workspace list on the right displays all of the available target workspaces that you can link to from this Performance Summary node, including the new workspace, *New Topology Workspace*.
11. Select *New Topology Workspace*.

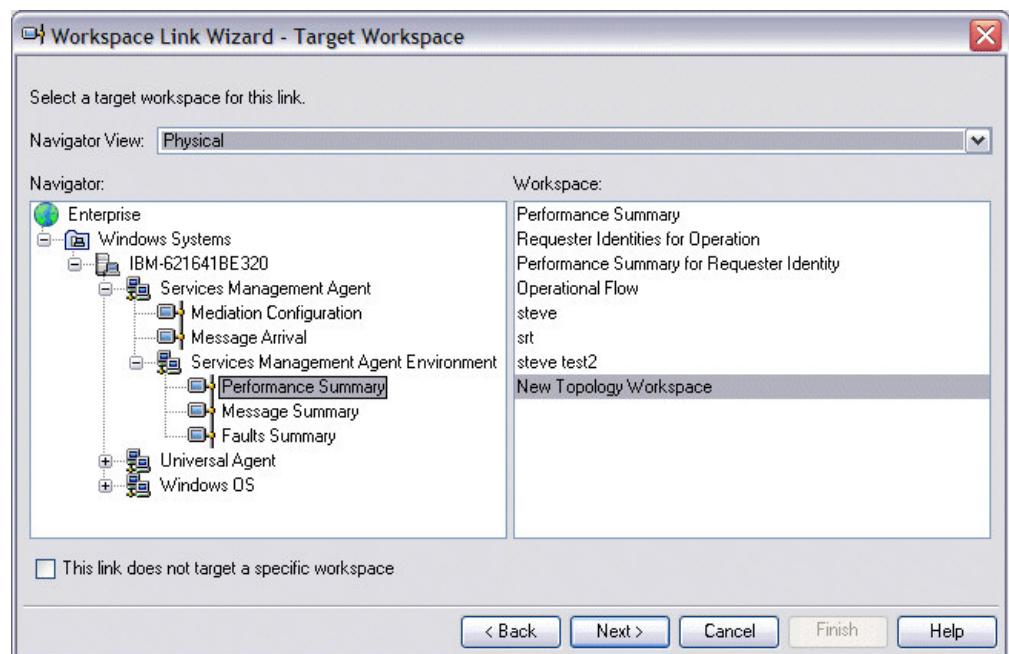


Figure 69. Selecting the new topology workspace as the target for the workspace link

12. Click **Next**. The **Workspace Link Wizard - Target Filters** page is displayed. Use this page to specify an expression that identifies the path to the target computer system. For this example, use the IP address filter.
13. Select the *IP address* row in the table, and click **Modify Expression**. The **Expression Editor - IP address** page is displayed.
14. Click **Symbol**. The Symbols page is displayed.
15. Scroll down the list of attributes that are available in the Services Inventory table, and select *Local IP Address (Unicode)*.

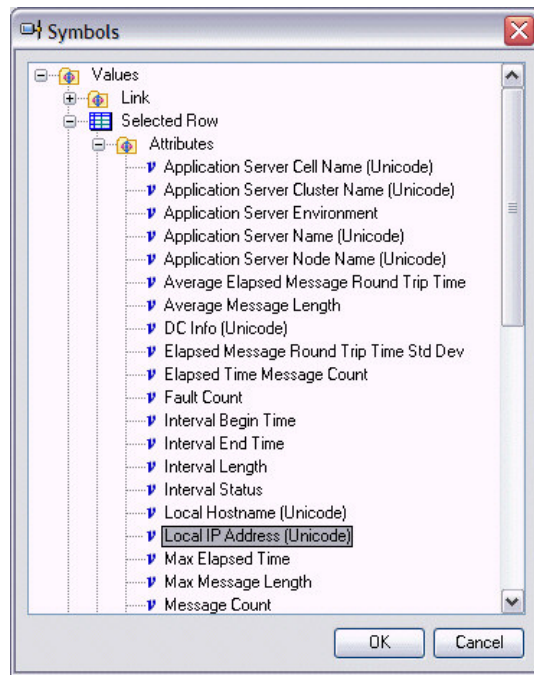


Figure 70. Selecting the *Local IP Address (Unicode)* attribute to define the filter path to the target computer system

16. Click **OK**. You are returned to the **Expression Editor - IP address** page with the expression for the IP address attribute displayed.
17. Click **OK**. The **Expression Editor - IP address** page is closed and you are returned to the **Workspace Link Wizard - Target Filters** page, with the IP address filter now assigned to the expression for the Local IP Address (Unicode) attribute.

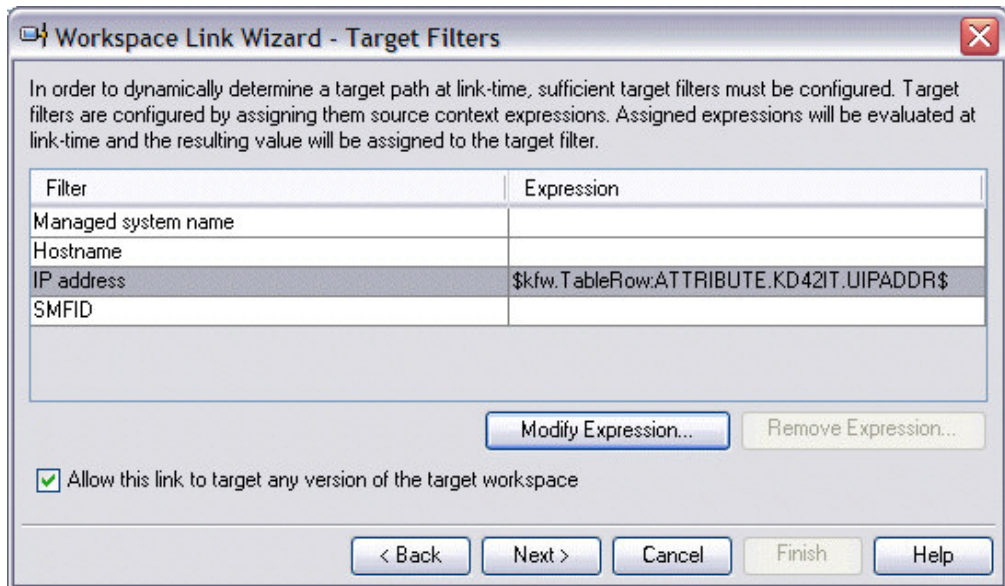


Figure 71. The IP address is assigned to the expression for the Local IP Address (Unicode) attribute to define the filter path to the target computer system

18. Click **Next**. The **Workspace Link Wizard - Parameters** page is displayed. This is where you add the additional link symbols supported by ITCAM for SOA to define the service port and operation pair context used by the query to the SOA Domain Management Server. Refer to Table 29 on page 185 for the list of supported link symbols that you need to add to this link definition.
 19. Now, to test whether the link comes from a provider row or a requester row in the Services Inventory table, configure the `linkIsEnabled` symbol by completing these steps:
 - a. Select the `linkIsEnabled` row and click **Modify Expression**.
 - b. Because the link is coming from a provider row, in the **Expression Editor - linkIsEnabled** page, enter the following value for `linkIsEnabled`:
`$kfw.TableRow.ATTRIBUTE.KD42IT.SVCTYPE$=1`
 If the link was coming from a requester row, you would create a link to a workspace containing the Operational Flow for Application Server workspace, and you would specify the value for `linkIsEnabled` as:
`$kfw.TableRow.ATTRIBUTE.KD42IT.SVCTYPE$=0`
 - c. The **Expression Editor - linkIsEnabled** page now displays the expression for the `linkIsEnabled` symbol. Click **OK**.
 - d. You are returned to the **Workspace Link Wizard - Parameters** page, showing the expression for the `linkIsEnabled` link symbol.
 20. Because you are linking from a provider row, you must add and configure additional link symbols for operation name, operation namespace, service port name, service port namespace, and application server environment. Click **Add Symbol**. The **Add Symbol** window is displayed.
- Tip:** If you are linking from a requester row, these additional link symbols are not required.
21. In the **Symbol** field, type `OperationName` exactly as shown (no spaces, case-sensitive), and then click **OK**. The `OperationName` link symbol is added to the list of symbols in the **Workspace Link Wizard - Parameters** page, under the existing `linkIsEnabled` symbol.

22. Select the OperationName row and click **Modify Expression**. The **Expression Editor - OperationName** page is displayed.
23. Click **Symbol**. The Symbols page is displayed.
24. Scroll down the list of attributes that are available in the Services Inventory table, and select *Operation Name (Unicode)*.
25. Click **OK**. You are returned to the **Expression Editor - OperationName** page with the expression for the OperationName symbol displayed.
26. Click **OK**. The **Expression Editor - OperationName** page is closed and you are returned to the **Workspace Link Wizard - Parameters** page, with the link symbol definition for the OperationName link symbol.

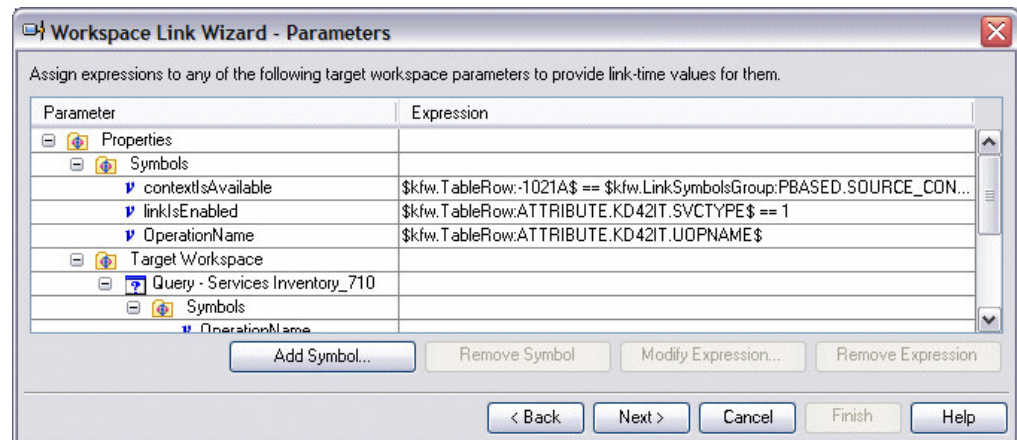


Figure 72. The OperationName link symbol is defined with the expression for the Operation Name attribute in the Services Inventory table

27. Repeat steps 20 on page 192 through 26 to add the other required link symbols to the link definition for linking to the Operational Flow for Operation workspace, specifying the link symbol names and attributes shown in Table 30:

Table 30. Link symbol names and their associated attributes from the Services Inventory table

Link symbol name	Attribute
OperationNamespace	Operation Namespace (Unicode)
ServicePortName	Service Port Name (Unicode)
ServicePortNamespace	Service Port Namespace (Unicode)
AppServerEnvironment	Application Server Environment

28. You must add one more link symbol to define the operational flow context, depending on whether you are linking from a provider row to the Operational Flow for Operation workspace, or from a requester row to the Operational Flow for Application Server workspace.

Add the *OpFlowContext* link symbol to the link definition, specifying the expression exactly as shown in Table 31

Table 31. The OpFlowContext link symbol name and expression

Service Type	Attribute expression for OpFlowContext
Provider	CALL("com.ibm.management.soa.tepui.utils.SCPLinkUtil", "getOpFlowContext", null)

Table 31. The OpFlowContext link symbol name and expression (continued)

Service Type	Attribute expression for OpFlowContext
Requester	CALL("com.ibm.management.soa.tepui.utils.SCPLinkUtil", "getAppOpFlowContext", null)

After you add in all of the link symbols, the **Workspace Link Wizard - Parameters** page will appear like Figure 73.

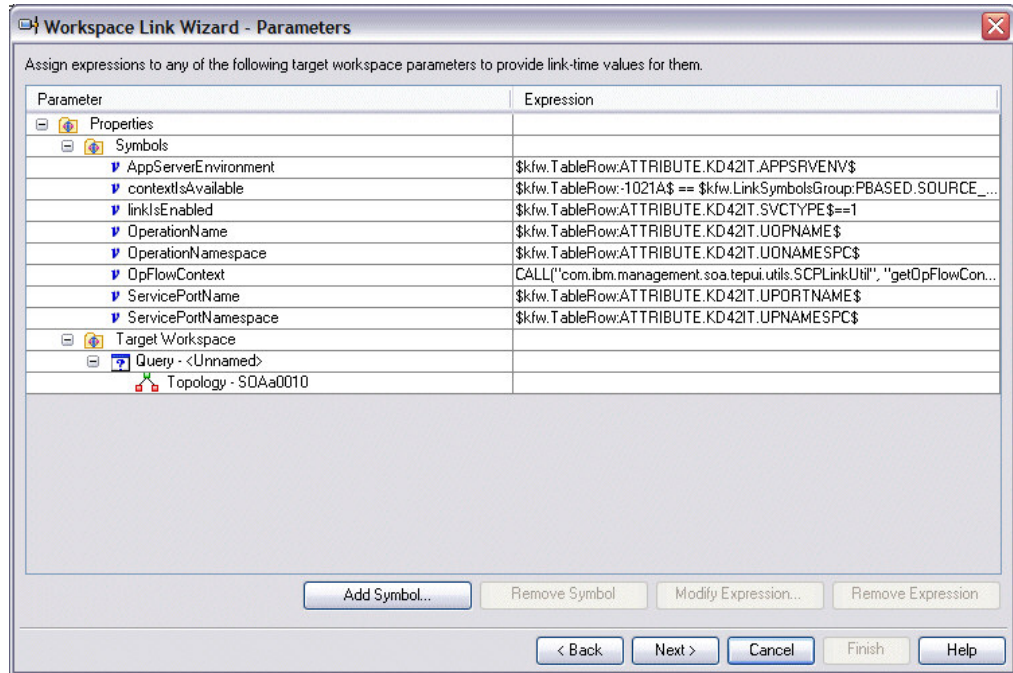


Figure 73. Link symbols defined in the workspace link definition

29. Click **Next**. The **Workspace Link Wizard - Summary** page is displayed. Confirm the task and click **Finish**.

You can now select your newly defined workspace link from a row in the Services Inventory table. From any row in the table, select the workspace link icon and select the new dynamic link, referred to in this example as *My New Workspace Link*, or right-click the row and select **Link to -> My New Workspace Link**.

If you had not previously attempted to drop a topology view into the available empty area of the workspace, this view is still empty. Select the Topology icon from the toolbar at the top of the workspace and then click in the empty view space to drop the topology view into place. The generic Topology view is initially loaded, but then as the context is retrieved from the SOA Domain Management Server, the view changes to the Operational Flow for Operation view, and the view is populated with data, similar to that shown in Figure 74 on page 195.

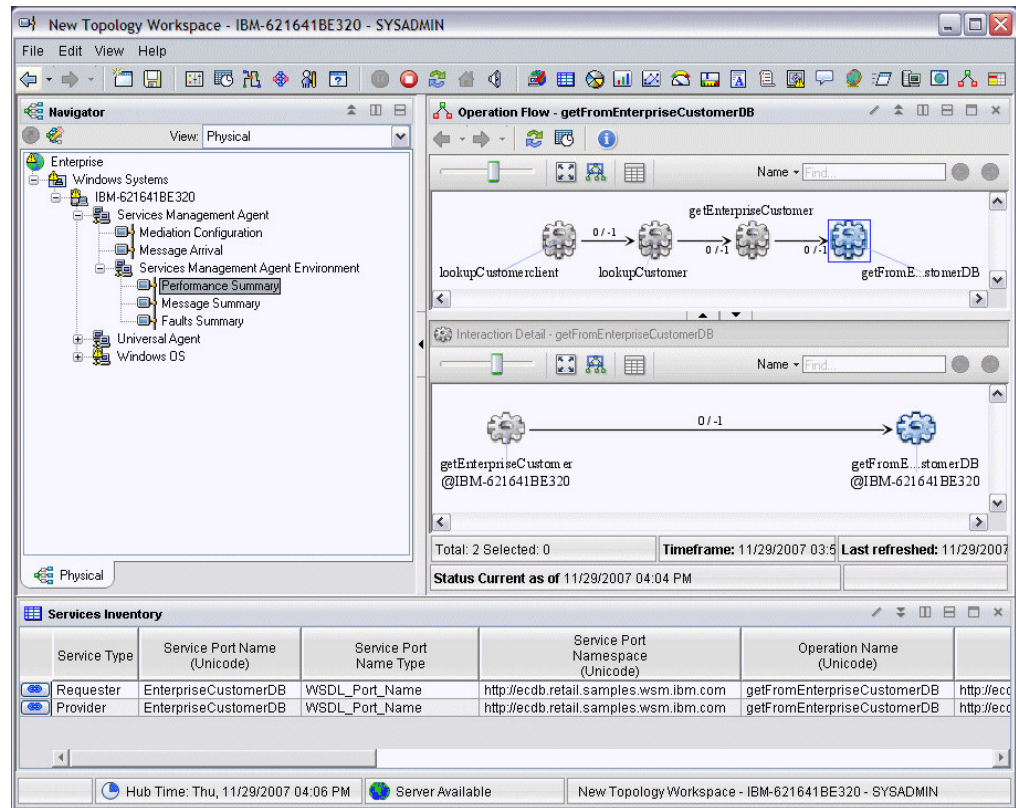


Figure 74. The new workspace populated with data in the Operational Flow for Operation view

Linking from the Situation Event Results workspace

You might want to create a dynamic workspace link from a row in the Initial Situation Values view in the Situation Event Results workspace to a new workspace that contains an Operational Flow for Operation topology view or the Operational Flow for Application Server topology view. You can create links for situations that are based on the ITCAM for SOA Services Inventory_610 or Services Inventory Requester Identity_610 attribute groups. These attribute groups contain the attributes that are used to create the link symbols required for linking to the Operational Flow workspaces.

To use the Service Type attribute to determine the target Operational Flow workspace for the link, it is recommended to configure the link symbol *linkIsEnabled*. This means that you create two link definitions:

- A link definition that uses the link symbol *linkIsEnabled* to check if the Service Type attribute is set to *provider* and links to the Operational Flow for Operation workspace.
- A link definition that uses the link symbol *linkIsEnabled* to check if the Service Type attribute is set to *requester* and links to the Operational Flow for Application Server workspace.

For example, assume that a new workspace was created that is similar to the example in “Creating the workspace” on page 187, which is a secondary workspace on the Performance Summary node, defined as the target of a

workspace link, and containing an Operational Flow for Operation topology view. In addition, you might define another secondary workspace, also defined as a target of a workspace link, and containing an Operational Flow for Application Server workspace, for use in linking from a row in the Initial Situation Values view for an ITCAM for SOA based situation with a service type of *requester*.

Also, assume that a situation (for example, the MessageSize_610 situation) is triggered, and that the Situation Event Results workspace is opened for this event.

You can define a new workspace link from a row in the Initial Situation Values view to the *New Topology Workspace* by completing these steps:

1. In the Initial Situation Values view, select the workspace link icon and select **Link Wizard**, or right-click on a row in the table and select **Link To -> Link Wizard**. The **Workspace Link Wizard - Welcome** page is displayed.
2. Select the radio button to **Create a new link** and click **Next** (unless otherwise directed, click **Next** to proceed step by step through the Link Wizard).
3. In the **Workspace Link Wizard - Link Name** page, type the name and an optional description for the workspace link. This is the name that is displayed in the **Link To** menu when you want to select this workspace link. For purposes of this example, the name is specified as *My Situation Workspace Link*.
4. In the **Workspace Link Wizard - Link Type** page, select the radio button for the **Dynamic** link type.
5. In the **Workspace Link Wizard - Target Workspace** page, verify that the Navigator View field at the top of the page points to the *Physical* view. To display the Performance Summary node link that is associated with your newly created secondary workspace, expand the Navigator node tree on the left.
6. To highlight it, click the Performance Summary. The Workspace list on the right displays all of the available target workspaces that you can link to from this Performance Summary node, including the new workspace, *New Topology Workspace*.
7. Select *New Topology Workspace* and click **Next**.
8. In the **Workspace Link Wizard - Target Filters** page, specify an expression that identifies the path to the target computer system. For this example, use the IP address filter. Select the *IP address* row in the table, and click **Modify Expression**.
9. In the **Expression Editor - IP address** page, click **Symbol**.
10. In the Symbols page, scroll down the list of attributes that are available in the Initial Situation Values table, and select *Local IP Address (Unicode)*.
11. You are returned to the **Expression Editor - IP address** page with the expression for the IP Address attribute displayed. Click **OK**.
12. In the **Workspace Link Wizard - Target Filters** page, the IP address filter is now assigned to the expression for the Local IP Address (Unicode) attribute. Click **Next**.
13. To test whether the situation comes from a provider or a requester, you must configure the *linkIsEnabled* symbol by following these steps:
 - a. Select the *linkIsEnabled* row and click **Modify Expression**.
 - b. Because the situation is coming from a provider, in the **Expression Editor - linkIsEnabled** page, enter the following value for *linkIsEnabled*:

```
TOINT($EVENTRESULTI:ATTRIBUTE.KD42IT.UTABLEVER$)>1 &&
$EVENTRESULTI:ATTRIBUTE.KD42IT.SVCTYPE$ == 1
```


If the situation was coming from a requester, you would create a link to a workspace containing the Operational Flow for Application Server workspace, and you would specify the value for *linkIsEnabled* as:

```
TOINT($EVENTRESULT1:ATTRIBUTE.KD42IT.UTABLEVER$)>1 &&  
$EVENTRESULT1:ATTRIBUTE.KD42IT.SVCTYPE$ == 0
```

- c. The **Expression Editor - linkIsEnabled** page now displays the expression for the *linkIsEnabled* symbol. Click **OK**.
 - d. You are returned to the **Workspace Link Wizard - Parameters** page, showing the expression for the *linkIsEnabled* link symbol.
14. In the **Workspace Link Wizard - Parameters** page, add the additional link symbols that are supported by ITCAM for SOA to define the service port and operation pair context used by the Operational Flow for Operation workspace. For the list of supported link symbols that you must add to this link definition, refer to Table 29 on page 185.

Tip: If you are linking to the Operational Flow for Application Server workspace for a situation with a service type of *requester*, you only have to define the *OpFlowContext* link symbol.

To create these link symbols, complete these steps:

- a. Click **Add Symbol**.
- b. In the **Symbol** field of the **Add Symbol** window, type *OperationName* exactly as shown (no spaces, case-sensitive), and click **OK**. The *OperationName* link symbol is added to the list of symbols in the **Workspace Link Wizard - Parameters** page, under the existing *linkIsEnabled* symbol.
- c. Select the *OperationName* row and click **Modify Expression**.
- d. In the **Expression Editor - OperationName** page, click **Symbol**.
- e. Scroll down the list of attributes that are available in the Services Inventory table, and select *Operation Name (Unicode)* and click **OK**.
- f. The **Expression Editor - OperationName** page now displays the expression for the *OperationName* symbol. Click **OK**.
- g. You are returned to the **Workspace Link Wizard - Parameters** page, showing the expression for the *OperationName* link symbol.
- h. To add the other required link symbols to the link definition, repeat these steps, specifying the link symbol names and attributes that are shown in Table 30 on page 193.

After you have added in all of the link symbols, the **Workspace Link Wizard - Parameters** page should look similar to Figure 75 on page 198.

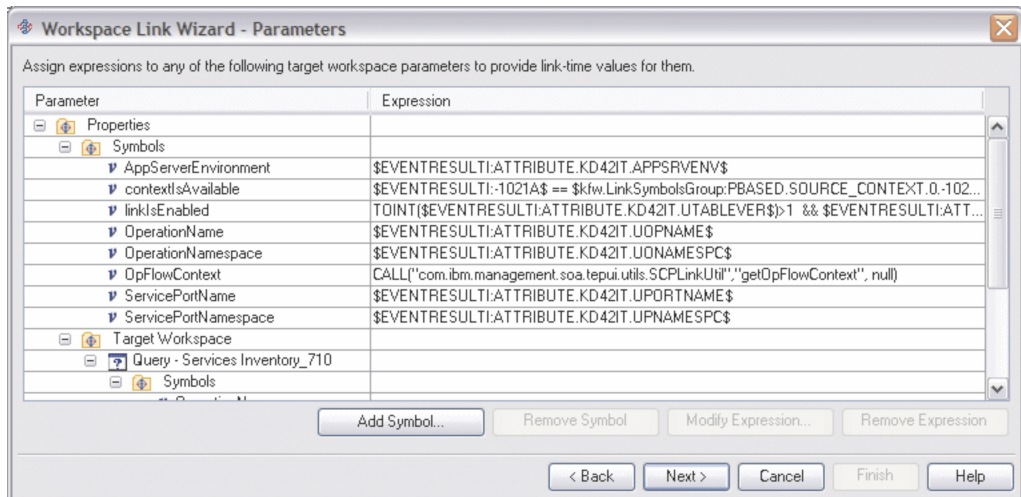


Figure 75. Link symbols defined in the workspace link definition

Important: The expressions for the link symbols in Figure 75 are not the same as the expressions defined in Figure 73 on page 194. This is because the context for the service port and operation pair is obtained from the Situation Event Results workspace, rather than from the Services Inventory table in the Performance Summary workspace.

15. Click **Next**. The **Workspace Link Wizard - Summary** page is displayed. Confirm the task and click **Finish**.
16. In the **Input** page, type the Product Code *kd4* and click **OK**.

You can now select your newly defined workspace link from a row in the Initial Situation Values table view. From any row in the table, select the workspace link icon and select the new dynamic link, referred to in this example as *My Situation Workspace Link*, or right-click the row and select **Link to -> My Situation Workspace Link**. The *New Topology Workspace* is then displayed, similar to the example shown in Figure 74 on page 195.

Chapter 10. Situations

ITCAM for SOA provides predefined situations to help you monitor critical activity. They serve as templates for creating customized situations for your own use. These predefined situations are activated when they are distributed to the node that you are monitoring. After they are configured, the alerting functions that are included as part of these predefined situations trigger event notification.

Situations that are associated with the Services Management Agent level use data from the Message Arrival Threshold_610 attributes. Situations at the Services Management Agent Environment level use data from the Services Inventory_610 attributes group.

You can also use Services Inventory Requester Identity_610 attributes to define a response time threshold setting for a specific requester identity (see Chapter 5, “Workspaces for monitoring by requester identity,” on page 47).

Most situations are distributed and started automatically. You must manually distribute and start the MaxMessageSize_610, MaxResponseTimeWarning_610 and MaxResponseTimeCritical_610 situations. To distribute a situation to a specific agent or managed system agent list, use the Tivoli Enterprise Portal Situation Editor.

You can also use the Situation Editor to associate a situation with a wanted workspace node. For information about creating situations and associating them with workspace nodes, see the *IBM Tivoli Monitoring User's Guide*.

Modify predefined situations for your environment: These predefined situations are provided with ITCAM for SOA and serve as a *starting point* for your situation development. If you create and deploy your own new situations tailored for your environment, do *not* deploy them. Copy and modify these predefined situations as required, and deploy them as your own customized versions, preserving the original predefined situations for future use if necessary.

Default sampling intervals: If you create your own new situation from these predefined situations, be aware that the default sampling interval is defined as 15 minutes. This setting might not be what you want for your situation. Therefore, set it to the correct value for your environment.

Attribute descriptions: When you select an attribute in the Situation Editor, hover help information is displayed. For additional information about the various attributes that are used in the predefined situations that are provided with ITCAM for SOA, see Chapter 12, “Attribute groups,” on page 257.

For more information about creating or modifying situations, refer to the Tivoli Enterprise Portal online help or the *Tivoli Enterprise Portal Administrator's Guide*.

How the situations work

Situations are tests of system conditions that you select to monitor. The tests are expressed in an IF-TRUE format, and the tested value is an attribute expressed in the form *attribute-group.attribute-name*. If the specified condition occurs or exists, the situation is true, and an alert is issued.

Avoid using negative values

If you define situations that use a counter or a range of numbers, always provide a threshold or use values in a positive range of numbers. If the Tivoli Enterprise Monitoring Agent encounters an undefined value for an attribute, the agent interprets this value as a *negative value*. This value can cause a situation to be interpreted incorrectly as *true* if its threshold setting includes negative values. Examine some of the predefined situations described in the following sections that use a greater-than-or-equal-to-zero expression. Use this expression to prevent a situation from inadvertently becoming true when an attribute contains an undefined value.

Expert advice

When a situation becomes true, Tivoli Enterprise Portal displays event indicators on the associated node in the Navigator Physical view. When you move the cursor over the event indicator, additional flyover information about situations is displayed. To open the event workspace, click one of the situations in this list. One of the views in this workspace is Expert Advice, which provides guidance on what is happening and how to handle a situation when it becomes true.

Predefined situations

Table 32 describes the predefined situations.

Table 32. Predefined situations for monitoring agents

Situation name	Brief description
"The Fault_610 situation" on page 201	Triggered if web services faults occurred, based on defined filtering criteria or application problems. The initial threshold is set to 0.
"The MessageSize_610 situation" on page 209	Triggered if the average message length of all messages during the time interval is greater than expected. The initial threshold is set to 1600 bytes.
"The MaxMessageSize_610 situation" on page 202	Triggered if the message length of any single message during the time interval is greater than expected. The initial threshold is set to 1600 bytes.
"The MessageArrivalCritical_610 situation" on page 207	Triggered if excessive web services traffic is received from specified remote clients during a specified time interval. The initial settings detect traffic from all combinations of service port name and namespace, operation name and namespace, and remote IP address, over the three most recent 30-second sampling intervals (time interval set to 90 seconds). The initial threshold is set to 50 messages. This situation is cleared when the MessageArrivalClearing_610 situation is triggered.

Table 32. Predefined situations for monitoring agents (continued)

Situation name	Brief description
"The MessageArrivalClearing_610 situation" on page 205	Clears a previous MessageArrivalCritical_610 situation. Triggered when web services traffic received from specified remote clients during a specified time interval falls below the threshold level. The initial settings detect traffic from all combinations of service port name and namespace, operation name and namespace, and remote IP address, over the three most recent 30-second sampling intervals (time interval set to 90 seconds). The initial threshold is set to 40 messages. This situation clears itself after 5 minutes.
"The ResponseTimeCritical_610 situation" on page 210	Triggered if the average elapsed round-trip response time to complete web service requests during the time interval exceeds the expected threshold. The threshold is initially set to greater than 10000 ms (10 seconds).
"The ResponseTimeWarning_610 situation" on page 211	Triggered if the average elapsed round-trip response time to complete web service requests during the time interval exceeds the expected threshold. The threshold is initially set to greater than 8000 ms (8 seconds) but less than or equal to 10000 ms (10 seconds).
"The MaxResponseTimeCritical_610 situation" on page 203	Triggered if the elapsed round-trip response time to complete any single web service request exceeds the expected threshold during the time interval. The threshold is initially set to greater than 10000 ms (10 seconds).
"The MaxResponseTimeWarning_610 situation" on page 204	Triggered if the elapsed round-trip response time to complete any single web service request exceeds the expected threshold during the time interval. The threshold is initially set to greater than 8000 ms (8 seconds) but less than or equal to 10000 ms (10 seconds).
"The BusinessProcessFault situation" on page 212	Triggered when a Business Process Definition (BPD) exits with a failure condition. Available with ITCAM for SOA monitoring agents version 7.2 or later.
"The BusinessProcessTerminated situation" on page 212	Triggered when a BPD exits because it was terminated by the user. Available with ITCAM for SOA monitoring agents version 7.2 or later.

The BusinessProcessFault and BusinessProcessTerminated situations require the ITCAM for SOA monitoring agent version 7.2 or later. All other situations operate with ITCAM for SOA monitoring agents at version 6.1.0 or later

The Fault_610 situation

The Fault_610 situation alerts you when a web services fault occurs during the most recently completed monitoring interval. The situation uses two values from the Services Inventory_610 attributes group:

Fault_Count

The integer count of the number of faults observed during the current time interval.

Interval_Status

The status of the current time interval, and has one of the two values, *Complete* or *Incomplete*.

If the fault count is greater than 0, and if the interval status has the value *Complete*, the situation is triggered.

The Fault_610 situation is described by the following formula:

```
**IF *VALUE Services_Inventory_610.Fault_Count *GT 0
*AND *VALUE Services_Inventory_610.Interval_Status *EQ Complete
```

For each SOAP fault that is received in response to a service operation, the Fault_Count attribute is incremented. This fault might occur for one of the following reasons:

- The message was rejected according to filtering criteria defined in the monitoring agent.
- There might be an application-based fault. For example, the application might not be running in the application server runtime environment.

Navigate to the Faults Summary workspace and examine the summary of faults that were reported. In the Fault Details table view, locate the fault of interest and examine the contents of the Fault String column for more information. Then, to correct any problems with your applications, take the necessary action.

If the applications are not the source of the problem, in the Navigator Physical view, select the Services Management Agent node and examine the filter criteria that is defined in the Data Collector Filter Control Configuration view. Filter criteria are defined to reject messages based on a specified combination of service port name, service port namespace, operation name, operation namespace, and remote IP address. If these criteria are not defined correctly, undesired faults might be received as a result.

To modify your filter criteria as required, use the **AddFiltrCntrl_610** or the **DelFiltrCntrl_610** Take Action commands.

The MaxMessageSize_610 situation

The MaxMessageSize_610 situation monitors the length, in bytes, of *each message* during the web services flow to detect whether the maximum length of any single message observed during the most recently completed monitoring interval exceeds the monitored threshold. This situation is initially defined with a maximum message length threshold of 1600 bytes. When this situation is triggered, the application might be sending a message larger than expected.

The MaxMessageSize_610 situation uses two values from the Services Inventory_610 attributes group:

Max Message Length

The length, in bytes, of the single largest message observed during the current monitoring interval.

Interval_Status

The status of the current time interval, and has one of the two values, *Complete* or *Incomplete*.

If the maximum message length is greater than 1600 bytes, and if the interval status has the value *Complete*, the situation is triggered.

The MaxMessageSize_610 situation is described by the following formula:

```
**IF *VALUE Services_Inventory_610.Max_Msg_Length *GT 1600
*AND *VALUE Services_Inventory_610.Interval_Status *EQ Complete
```

Examine the message that triggered this situation to determine if it is an acceptable length. If it is not, consider adjusting the threshold for this situation to suit your environment.

Distributing the situation: By default, the MaxMessageSize_610 situation is not distributed to the agent and run at startup because you might not want this situation active at the same time as the MessageSize_610 situation, which does run at startup and uses the same threshold settings for checking the average message length. To configure the startup condition and adjust thresholds as needed, use the Situation Editor.

Detecting possibly incomplete messages: You can use this situation as a template to create a similar situation that monitors for a message length *lower* than the specified threshold. You might want to use this type of situation to detect a problem with an application not sending an entire message. When you create a situation from this predefined situation, change the comparison operator from GT (Greater than) to LT (Less than) and set the threshold to check for a *minimum* message length, in bytes, that is appropriate for what you expect in your environment.

For example, you might want to create a situation called MinMessageSize_610, that detects message lengths less than 500 bytes. The formula to describe that situation might be as follows:

```
*IF *VALUE Services_Inventory_610.Min_Msg_Length *LT 500
*AND *VALUE Services_Inventory_610.Interval_Status *EQ Complete
```

The MaxResponseTimeCritical_610 situation

The MaxResponseTimeCritical_610 situation monitors the maximum elapsed round-trip response time, in milliseconds, for the completion of a web service request, observed during the most recently completed monitoring interval. This situation is triggered when the elapsed response time for a web service request exceeds the specified threshold value. This situation is initially defined with a maximum threshold of 10000 milliseconds (10 seconds).

The situation uses two values from the Services Inventory_610 attributes group:

Max Elapsed Time

The longest elapsed time, in milliseconds, of a single message observed during the time interval.

Interval Status

The status of the current time interval, indicated by one of two valid values, *Complete* or *Incomplete*.

The MaxResponseTimeCritical_610 situation is described by the following formula:

```
**IF *VALUE Services_Inventory_610.Max_Elapsed_Time *GT 10000
*AND *VALUE Services_Inventory_610.Interval_Status *EQ Complete
```

If this service call then calls other services, the delay might be due either to the response time of the other services, or possibly a network or communication problem between the service calls.

Check the response time of other services used by this service, or other services deployed to the same application server, and look for any patterns that might indicate a problem with a particular service, computer, application server, or part of your network.

You might also want to check monitoring information from other IBM Tivoli Monitoring, IBM Tivoli Composite Application Manager, and IBM OMEGAMON products that monitor resources used by your application (for example, database monitors, or operating system monitors). These monitors might detect problems that contribute to the excessive response time detected by this situation.

If the response time for your web services request is normal for your environment, consider adjusting this range threshold value to a more appropriate level. If you also have a `MaxResponseTimeWarning_610` situation defined, compare the threshold settings between these two situations and adjust them as needed to reflect appropriate critical and warning levels for your environment.

Distributing the situation: By default, the `MaxResponseTimeCritical_610` situation is not distributed to the agent and run at startup because you might not want this situation active at the same time as the `ResponseTimeCritical_610` situation, which does run at startup and uses the same threshold settings for checking the average elapsed round-trip response time. To configure the startup condition and adjust thresholds as needed, use the Situation Editor.

The `MaxResponseTimeWarning_610` situation

The `MaxResponseTimeWarning_610` situation monitors the maximum elapsed round-trip response time, in milliseconds, for the completion of a web service request, observed during the most recently completed monitoring interval. This situation is triggered when the round-trip response time for a web service request exceeds the specified threshold value. This situation is initially defined with a range threshold of above 8000 milliseconds (8 seconds) up to 10000 milliseconds (10 seconds).

The situation uses two values from the `Services_Inventory_610` attributes group:

Max Elapsed Time

The longest elapsed time, in milliseconds, of a single message observed during the time interval.

Interval_Status

The status of the current time interval, and has one of the two values, *Complete* or *Incomplete*.

The `MaxResponseTimeWarning_610` situation is described by the following formula:

```
*IF *VALUE Services_Inventory_610.Max_Elapsed_Time *GT 8000
*AND *VALUE Services_Inventory_610.Max_Elapsed_Time *LE 10000
*AND *VALUE Services_Inventory_610.Interval_Status *EQ Complete
```


If this service call then calls other services, the delay can be due either to the response time of the other services, or possibly a network or communication problem between the service calls.

Check the response time of other services used by this service, or other services deployed to the same application server, and look for any patterns that might indicate a problem with a particular service, computer, application server, or part of your network.

You might also want to check monitoring information from other IBM Tivoli Monitoring, IBM Tivoli Composite Application Manager, and IBM OMEGAMON products that monitor resources used by your application (for example, database monitors, or operating system monitors). These monitors might detect problems that contribute to the excessive response time detected by this situation.

If the response time for your web services request is normal for your environment, consider adjusting this range threshold value to a more appropriate level. If you also have a `MaxResponseTimeCritical_610` situation defined, compare the threshold settings between these two situations and adjust them as needed to reflect appropriate critical and warning levels for your environment.

Distributing the situation: By default, the `MaxResponseTimeWarning_610` situation is not distributed to the agent and run at startup because you might not want this situation active at the same time as the `ResponseTimeWarning_610` situation, which does run at startup and uses the same threshold settings for checking the average elapsed round-trip response time. To configure the startup condition and adjust thresholds as needed, use the Situation Editor.

The `MessageArrivalClearing_610` situation

The `MessageArrivalClearing_610` situation clears a previously triggered `MessageArrivalCritical_610` situation. The `MessageArrivalCritical_610` situation is triggered when more message traffic than expected occurs within a specified time interval. At a later time, when message traffic is reduced to an acceptable level for a specified time interval, this `MessageArrivalClearing_610` situation is triggered to clear the previous `MessageArrivalCritical_610` situation.

The `MessageArrivalClearing_610` situation monitors the number of requests that are sent to a web services server from a remote client. The number of requests that are counted can be controlled by situation criteria that you define for a specific combination of service port name, service port namespace, operation name, operation namespace, and remote IP address, or can be generalized to count requests for any service port name, service port namespace, operation name, operation namespace, and remote IP address. You can set the threshold for the number of messages counted, and the time interval for the message count to remain below the threshold. When these criteria are met, this situation is triggered, which in turn causes the `MessageArrivalCritical_610` situation to be cleared until the next time that message traffic exceeds the critical threshold.

The situation uses these values from the `Message Arrival Threshold_610` attributes:

Current Message Count

The number of actual messages intercepted during the current monitoring interval. The threshold for this number is initially defined as 40.

Service Port Name (Unicode)

The name of the service port for which the threshold is defined.

Service Port Namespace (Unicode)

The namespace that is used to fully qualify the service port name for which the threshold is defined.

Operation Name (Unicode)

The name of the operation for which the threshold is defined.

Operation Namespace (Unicode)

The namespace that is used to fully qualify the operation for which the threshold is defined.

Remote IP Address (Unicode)

The IP address, if available, for the computer system that called the service for which the threshold is defined.

Time Interval

The sliding time interval, in seconds, for which the condition is applied.
The threshold for this attribute is initially defined as 90 seconds, or the equivalent of three sampling intervals of 30 seconds each.

The MessageArrivalClearing_610 situation is described by the following formula:

```
*IF *VALUE Message_Arrival_Threshold_Table_610.Current_Message_Count *LT 40
*AND
*VALUE Message_Arrival_Threshold_Table_610.Service_Port_Name_(Unicode) *EQ '*'
*AND *VALUE Message_Arrival_Threshold_Table_610.Service_Port_Namespace_(Unicode)
*EQ '*'
*AND *VALUE Message_Arrival_Threshold_Table_610.Operation_Name_(Unicode) *EQ '*'
*AND
*VALUE Message_Arrival_Threshold_Table_610.Operation_Namespace_(Unicode) *EQ '*'
*AND
*VALUE Message_Arrival_Threshold_Table_610.Remote_IP_Address_(Unicode) *EQ '*'
*AND *VALUE Message_Arrival_Threshold_Table_610.Time_Interval *EQ 90
*UNTIL ( *TTL 0:00:05:00 )
```

This situation is predefined to trigger if the number of messages from any combination of service port name, service port namespace, operation name, operation namespace, and remote IP address falls to less than 40 during the specified time interval of 90 seconds (three 30- second sampling intervals).

Table 33 displays the initial values of the settings for this predefined situation.

Table 33. Predefined settings for the MessageArrivalClearing_610 situation

Setting	Initial value
Message count threshold	Less than 40 messages during the time interval
Service port name	* (all service port names match the criteria)
Service port namespace	* (all service port namespaces match the criteria)
Operation name	* (all operation names match the criteria)
Operation namespace	* (all operation namespaces match the criteria)
Remote IP address	* (all remote IP addresses match the criteria)
Time interval	90 seconds (three 30-second sampling intervals)

These initial settings result in all messages (all combinations of service port name, service port namespace, operation name, operation namespace, and remote IP address) from all remote clients being counted as part of the monitored web services traffic. When messages are received during the 90-second time interval, they are counted and compared to the 40 message maximum threshold. When the count reaches 39 or less messages within the 90-second time interval, the situation is triggered. After each 30-second sampling interval, the count is adjusted for message traffic that is received during the most recent 90-second time interval (the past three 30 second sampling intervals).

Modify these initial settings for your environment: If you want to monitor specific message traffic, you might want to specify a particular combination of service port name, service port namespace, operation name, operation namespace, and remote IP address. Set the threshold for the message count to a value that makes sense for your environment and expected message traffic. The threshold setting for this situation is less than the threshold setting for the `MessageArrivalCritical_610` situation. This setting reduces the chance for repeated triggering of situations if the message count happens to fluctuate near the threshold settings. Specify the time interval as a multiple of 30 seconds (if you specify a different value, it is rounded up to the nearest 30-second increment).

This situation is triggered when the number of messages received from one or more remote clients with the specified combination of service port name, service port namespace, operation name, operation namespace, and remote IP address falls below the defined message count threshold level during the most recent 90-second time interval. This indicates that web services traffic decreased to acceptable levels, and a previously triggered `MessageArrivalCritical_610` situation can be cleared.

If you have a `MessageArrivalCritical_610` situation that is not cleared when expected, examine the settings in this `MessageArrivalClearing_610` situation and verify that the service port name, service port namespace, operation name, operation namespace, and remote IP address values are defined correctly, and the message count threshold setting and the time interval setting. The message count is sampled at 30-second intervals. The time interval setting, rounded up to the nearest 30-second increment, defines how long (that is, how many 30-second samples) the message count must stay below the threshold in order for the web services traffic to stabilize and the `MessageArrivalCritical_610` situation to be cleared.

After the `MessageArrivalClearing_610` situation is triggered and the message count remains less than the threshold value, the `MessageArrivalClearing_610` situation clears itself after a five-minute interval.

The `MessageArrivalCritical_610` situation

The `MessageArrivalCritical_610` situation is designed to alert you to excessive amounts of web services traffic (the number of messages received from one or more remote clients exceeds a specified threshold). This situation monitors the number of requests that were sent to a web services server from one or more remote clients during a specified time interval.

The requests that are included in the count must match selection criteria that you define for a specific combination of service port name, service port namespace, operation name, operation namespace, and remote IP address. You can also generalize the criteria to count the number of messages for any service port name, service port namespace, operation name, operation namespace, and remote IP

address. If the number of messages that meet this selection criteria within the specified time interval exceed the threshold value that you define, this situation is triggered to indicate that excessive traffic was detected.

The situation uses these values from the Message Arrival Threshold_610 attributes:

Current Message Count

The number of actual messages intercepted during the current monitoring interval. The threshold for this number is initially defined as 50.

Service Port Name (Unicode)

The name of the service port for which the threshold is defined.

Service Port Namespace (Unicode)

The namespace that is used to fully qualify the service port name for which the threshold is defined.

Operation Name (Unicode)

The name of the operation for which the threshold is defined.

Operation Namespace (Unicode)

The namespace that is used to fully qualify the operation for which the threshold is defined.

Remote IP Address (Unicode)

The IP address, if available, for the computer system that called the service for which the threshold is defined.

Time Interval

The sliding time interval, in seconds, for which the condition is applied. The threshold for this attribute is initially defined as 90 seconds, or the equivalent of three sampling intervals of 30 seconds each.

The situation is described by the following formula:

```
*IF *VALUE Message_Arrival_Threshold_Table_610.Current_Message_Count *GT 50
*AND
*VALUE Message_Arrival_Threshold_Table_610.Service_Port_Name_(Unicode) *EQ '*'
*AND *VALUE Message_Arrival_Threshold_Table_610.Service_Port_Namespace_(Unicode)
*EQ '*'
*AND *VALUE Message_Arrival_Threshold_Table_610.Operation_Name_(Unicode) *EQ '*'
*AND
*VALUE Message_Arrival_Threshold_Table_610.Operation_Namespace_(Unicode) *EQ '*'
*AND
*VALUE Message_Arrival_Threshold_Table_610.Remote_IP_Address_(Unicode) *EQ '*'
*AND *VALUE Message_Arrival_Threshold_Table_610.Time_Interval *EQ 90
*UNTIL ( *SIT MessageArrivalClearing_610 )
```

This situation is predefined to trigger if the number of messages from any combination of service port name, service port namespace, operation name, operation namespace, and remote IP address exceeds 50 during the specified time interval of 90 seconds (the three most recent 30-second sampling intervals).

Table 34 on page 209 displays the initial values of the settings for this predefined situation.

Table 34. Predefined settings for the MessageArrivalCritical_610 situation

Setting	Initial value
Message count threshold	More than 50 messages during the time interval
Service port name	* (all service port names match the criteria)
Service port namespace	* (all service port namespaces match the criteria)
Operation name	* (all operation names match the criteria)
Operation namespace	* (all operation namespaces match the criteria)
Remote IP address	* (all remote IP addresses match the criteria)
Time interval	90 seconds (three 30 - second sampling intervals)

These initial settings result in all messages (all combinations of service port name, service port namespace, operation name, operation namespace, and remote IP address) from all remote clients being counted as part of the monitored web services traffic. As messages are received during the 90-second time interval, they are counted and compared to the 50 message maximum threshold. When the count reaches 51 or more messages within the 90-second time interval, the situation is triggered. After each 30-second sampling interval, the count is adjusted for message traffic received during the most recent 90-second time interval (the three most recent 30-second sampling intervals).

Modify these initial settings for your environment: You might want to specify a particular combination of service port name, service port namespace, operation name, operation namespace, and remote IP address, to monitor specific message traffic. Set the threshold for the message count to a value that makes sense for your environment and expected message traffic. The time interval should be specified as a multiple of 30 seconds (if you specify a different value, it is rounded up to the nearest 30 second increment).

When the message count exceeds the specified threshold, the situation is triggered. This indicates that traffic to this service is more than expected, and might be caused by a problem in the requesting service. Examine the service that is associated with the remote clients that match the service port name, service port namespace, operation name, operation namespace, and remote IP address criteria for any problems. If the amount of web services traffic does not seem to be excessive in your environment, consider adjusting the Message Count and Time Interval settings in this situation to more acceptable levels. When web services traffic falls to more acceptable levels, this MessageArrivalCritical_610 situation is automatically reset according to the criteria that is defined in the MessageArrivalClearing_610 situation. For more information, see “The MessageArrivalClearing_610 situation” on page 205.

The MessageSize_610 situation

The MessageSize_610 situation monitors the average length, in bytes, of the messages that are in the web services flow during the most recently completed monitoring interval. If the average length of the observed messages, including headers when possible, is greater than the specified threshold value, then this situation is triggered. This situation is initially defined with an average message length threshold of 1600 bytes.

The situation uses two values from the Services Inventory_610 attributes group:

Average Message Length

The average message length, in bytes, observed during the current time interval.

Interval_Status

The status of the current time interval, and has one of the two values, *Complete* or *Incomplete*.

If the average message length is greater than 1600 bytes, and if the interval status has the value *Complete*, the situation is triggered.

The MessageSize_610 situation is described by the following formula:

```
**IF *VALUE Services_Inventory_610.Avg_Msg_Length *GT 1600
*AND *VALUE Services_Inventory_610.Interval_Status *EQ Complete
```

When this situation is triggered, one or more applications sending messages larger than expected might be the cause. Examine the typical lengths of messages being sent to determine if they are acceptable for your environment. If they are not, consider adjusting the threshold for this situation for your environment.

Detecting possibly incomplete messages: Use this situation as a template to create a similar situation that monitors for an average message length that is *under* the specified threshold. You might want to use this type of situation to detect a problem with an application not sending an entire message. For example, you might want to create a situation that detects average message lengths less than 1000 bytes. When you create a situation from this predefined situation, change the comparison operator from GT (Greater than) to LT (Less than) and set the threshold to check for a *minimum* average message length, in bytes, that is appropriate for what you expect in your environment.

The ResponseTimeCritical_610 situation

The ResponseTimeCritical_610 situation is designed to monitor the average elapsed round-trip response time, in milliseconds, for the completion of a web service request. This critical situation is triggered when the average response time exceeds the threshold value. This situation is initially defined with a maximum threshold of 10000 milliseconds (10 seconds).

The situation uses two values from the Services Inventory_610 attributes group:

Average Elapsed Message Round Trip Time

The average elapsed round-trip time, in milliseconds.

Interval_Status

The status of the current time interval, indicated by one of the two valid values, *Complete* or *Incomplete*.

The ResponseTimeCritical_610 situation is described by the following formula:

```
*IF *VALUE Services_Inventory_610.Average_Elapsed_Message_Round_Trip_Time *GT 10000
*AND *VALUE Services_Inventory_610.Interval_Status *EQ Complete
```

This situation triggers when the average elapsed round-trip time for web service requests exceeds the specified threshold. If this service call then calls other services, the delay might be due either to the response time of the other services, or a network or communication problem between the service calls.

Check the response time of other services used by this service, or other services deployed to the same application server, and look for any patterns that might indicate a problem with a particular service, computer, application server, or part of your network. You might also want to check monitoring information from other IBM Tivoli Monitoring, IBM Tivoli Composite Application Manager, and IBM OMEGAMON products that monitor resources used by your application (for example, database monitors, or operating system monitors) to see whether these monitors detected problems that might contribute to the excessive response time detected by this situation.

If the response time for your web service request is normal for your environment, consider adjusting this threshold value to a more appropriate level. If you also have a ResponseTimeWarning_610 situation defined, compare the threshold settings between these two situations and adjust them as required to reflect appropriate critical and warning levels for your environment.

The ResponseTimeWarning_610 situation

The ResponseTimeWarning_610 situation is designed to monitor the average elapsed round-trip response time, in milliseconds, for the completion of a web service request. This warning situation is triggered when the average response time exceeds the threshold value. This situation is initially defined with a range threshold of above 8000-10000 milliseconds (or 8-10 seconds).

The situation uses two values from the Services_Inventory_610 attributes group:

Average Elapsed Message Round Trip Time

The average elapsed round-trip time, in milliseconds.

Interval_Status

The status of the current time interval, indicated by one of the two valid values, *Complete* or *Incomplete*.

The ResponseTimeWarning_610 situation is described by the following formula:

```
*IF *VALUE Services_Inventory_610.Average_Elapsed_Message_Round_Trip_Time *GT 8000
*AND *VALUE Services_Inventory_610.Average_Elapsed_Message_Round_Trip_Time *LE 10000
*AND *VALUE Services_Inventory_610.Interval_Status *EQ Complete
```

If this service call then calls other services, the delay might be due either to the response time of the other services, or possibly a network or communication problem between the service calls.

Check the response time of other services used by this service, or other services deployed to the same application server, and look for any patterns that might indicate a problem with a particular service, computer, application server, or part of your network. You might also want to check monitoring information from other IBM Tivoli Monitoring, IBM Tivoli Composite Application Manager, and IBM OMEGAMON products that monitor resources used by your application (for example, database monitors, or operating system monitors) to see whether these monitors detected problems that might contribute to the excessive response time detected by this situation.

If the response time for your web service request is normal for your environment, consider adjusting the range of the threshold value to a more appropriate level. If you also have a ResponseTimeCritical_610 situation defined, compare the threshold settings between these two situations and adjust them as needed to reflect appropriate critical and warning levels for your environment.

The BusinessProcessFault situation

The BusinessProcessFault situation is triggered when a Business Process Definition (BPD) exits with a failure condition.

The default status for this situation is **Critical**.

This situation uses the following value from the “Business_Process_Situation_Data attributes” on page 322:

Event_Nature

The type of event sent from the BPM server when the BPD finished with an error. The value for this situation is FAILED (3).

The BusinessProcessFault situation is described by the following formula:

```
*IF *VALUE Business_Process_Situation_Data.Event_Nature *EQ 3
```

This situation is triggered in real time when a BPD fault occurs. To find out the details of any BPD faults, use the flyover window or details window for the BPD node (see “Viewing details” on page 110).

The situation does not clear automatically. You must clear it manually in the Situation Event Console.

The BusinessProcessTerminated situation

The BusinessProcessTerminated situation is triggered when a Business Process Definition (BPD) exits because it was terminated by a user.

The default status for this situation is **Warning**.

This situation uses the following value from the “Business_Process_Situation_Data attributes” on page 322:

Event_Nature

The type of event sent from the BPM server when the BPD finished with an error. The value for this situation is TERMINATED (9).

The BusinessProcessTerminated situation is described by the following formula:

```
*IF *VALUE Business_Process_Situation_Data.Event_Nature *EQ 9
```

This situation is triggered in real time when a BPD is terminated. To find out the details of any BPD faults, use the flyover window or details window for the BPD node (see “Viewing details” on page 110).

To limit the situation to events that are terminated on an activity or a process, modify this situation definition in the Situation Editor and set the event type equal to activity or process.

The situation does not clear automatically. You must clear it manually in the Situation Event Console.

Creating your own situations

You can create your own situations by using the existing situations as a starting point, or you can create your own new situations. The situations that are provided with ITCAM for SOA are created from data in the following attribute tables:

- “Message Arrival Threshold_610 attributes” on page 269
- “Services Inventory_610 attributes” on page 271

You can use attributes from various ITCAM for SOA groups in your situations. In particular, using the Services Message Metric_610 table, you can generate situations based on each individual message record.

You can create situations that are based on averaged data across multiple records (for example, average response time) using any of the following attribute tables:

- Services Inventory_610
- Services Inventory Requester Identity_610
- Endpoint Inventory

These attribute tables contain aggregated data over each 5-minute monitoring interval.

Important: The Services Message Metric attributes table is still available if you want to continue to collect data for your own custom usage; however this table is no longer used by the product.

To create message arrival situations similar to the predefined Message Arrival Critical and Message Arrival Clearing situations, use the attributes in the Message Arrival Threshold_610 attributes table.

Attention: The default monitoring interval for newly created situations is 15 minutes. Change this value to 5 minutes to correspond to the interval used for calculation of average metrics in tables. When historical data collection is enabled, data in tables might also be cleared every 5-minute interval.

Renaming a situation: If you are running in IBM Tivoli Monitoring version 6.2.1 or later, and you change the name of a situation, the new name is not reflected in the operational flow views or group summary views until the next time the SOA Domain Management Server updates its information about the defined situations.

Creating situations for message arrival traffic

You can create a situation using attributes from the Message Arrival Threshold_610 attribute group to monitor specific message arrival traffic. You must create the situation so that it contains **all** of the fields that are used in the predefined MessageArrivalClearing_610 or MessageArrivalCritical_610 situations. In this case, you create the arrival monitoring criteria that is displayed as rows in the Message Arrival Details table view.

To create a new situation, make a copy of one of the predefined situations, modify the copy, and save it with a new situation name.

If you create a situation using the Current Message Count attribute in the Message Arrival Threshold_610 attributes group, you cannot use delta or percent functions. The monitoring agent supports only comparisons against the actual value. If you create and distribute a situation using delta or percent functions, the situation is not displayed in the Message Arrival Details view and is ignored.

Creating situations using the requester identity attribute

If you are creating situations using attributes from the Services Inventory Requester Identity_610 attributes table, create the situation so that it includes the Requester Identity attribute and any other metrics that you want to check. For example:

```
*IF *VALUE Services_Inventory_ReqID_610.Avg_Elapsed_Time *GT 8000
*AND *VALUE Services_Inventory_ReqID_610.Avg_Elapsed_Time *LE 10000
*AND *VALUE Services_Inventory_ReqID_610.Interval_Status *EQ Complete
*AND *VALUE Services_Inventory_ReqID_610.Requester_Identity *EQ 'User A'
```

Measuring service unavailability

You can determine whether a monitored service is *unavailable*. IBM Tivoli Composite Application Manager for SOA supports various ways to customize your situation definitions to determine if a service is not available. For example:

- You can create a custom situation that identifies a service operation as being unavailable if a fault message is observed. For an example of this type of situation, see “Creating a situation to detect faults” on page 215.
- You can restrict your definition of unavailability by creating a similar situation that is triggered when a particular fault code or fault string is observed in the message. For an example of this type of situation, see “Creating a situation to detect a fault code or string” on page 216.
- You might define unavailability as a condition that occurs when the average response time exceeds critical threshold, configuring an appropriate predefined situation or customizing your own for your environment.
- You might identify a service operation as being unavailable by creating a custom situation that watches for the number of response messages to drop to zero. This approach requires the use of a mechanism to generate regular, expected *heartbeat* messages. For example, if you have an automated function that performs an operation every three minutes, then you can determine that the service operation is unavailable any time that you have a five-minute monitoring interval with no responses. If you do not use an automated heartbeat function, you cannot distinguish between a service operation that is unavailable and the number of response messages dropping to zero only because all of your requesters have no work to do.
- You can take advantage of several additional service unavailability metrics provided in the Services Inventory_610 and Services Inventory Requester Identity_610 attribute group tables. These metrics might show when a service operation is unavailable because of responses not being sent by the service provider or received by the service requester. For a description of these metrics, see “Service unavailability metrics” on page 217.

Unavailability in groups

If an unavailability situation similar to these examples is associated with a front-end service as part of a defined group, when this situation is triggered, the group is displayed as unavailable. For additional information and examples of displaying service unavailability at the group level, see “Configuring for unavailability” on page 180.

You can customize the impact of a situation on the status and unavailability of a particular group. See “Changing the situation impact on status of group” on page 159.

Situation unavailability in the Group Summary view and the Operation Flow views takes into account only situations that are for the provider-side of an

operation. Therefore, the service type in the situation must be *Provider* if it is a situation that is defined from attributes from the Services Inventory or Service Inventory for Requester ID tables.

Creating a situation to detect faults

You can create a custom situation that identifies a service operation as being unavailable if a fault message is observed.

For example, you might start by using the predefined Fault_610 situation as a template, and creating a new customized situation to warn you that a service might be unavailable when at least one fault message is observed during the most recently completed monitoring interval. To create a custom situation, complete the following steps:

1. From the Navigator view, right-click the Faults Summary node under your preferred application server node.
2. To open the Situation Editor, select **Situations**. The situations that are associated with the Faults Summary node are displayed in the node tree.
3. In the Situation Editor, select the Fault_610 situation to display the predefined conditions for this situation.

This predefined situation uses two attributes from the Services Inventory_610 attributes group:

Fault Count

The number of faults observed during this monitoring interval. For this definition, a threshold value of *greater than 0* is defined.

Interval Status

The status of the monitoring interval, either *Incomplete* or *Complete*. For this definition, the threshold condition of *Complete* is defined.

The predefined formula for this situation causes the situation to be triggered when at least one fault is observed during the most recently completed monitoring interval. At the Faults Summary node level, when this situation is triggered, the state of the situation is configured to display an *Informational* icon on the Faults Summary node in the Navigator view.

4. Use the **Create Another** function to create a duplicate situation:
 - a. Name the new situation *Unavail_Warning_Due_To_Faults*.
 - b. Add a short description for the situation.
 - c. Optional: Modify the formula for the Fault Count attribute, setting the threshold to some minimum number of faults that must be observed in the most recently completed monitoring interval before triggering the situation. For this example, the same value *greater than 0* for the Fault_610 situation is used.
 - d. Change the state of the situation from **Informational** to **Warning**.
5. To save your situation definition and close the Situation Editor, click **OK**.
6. Later, as message traffic is observed, this situation is triggered when one or more faults are observed in the monitored environment. From the Navigator view, select the Faults Summary node to display the Faults Summary workspace.
7. In the Navigator Physical view, position the mouse cursor over the displayed fault subnode under the Faults Summary node. The hover help that identifies the triggered situation is displayed. Click the link indicator to open the Situation Event Results workspace to see what data caused the situation to be triggered, along with the affected operation.

8. In the Situation Event Results workspace, you can create a link to the Operational Flow workspace, and then take that link to display the service-to-service topology for the affected operation.
9. The topology view is displayed. The warning status indicator is displayed on the operation aggregate that is associated with the triggered situation. When you right-click the operation aggregate, additional help information is displayed. Double-clicking the operation aggregate displays the detail in the Interaction Detail portion of the view.
10. In the Interaction Detail portion of the view, you can display the details about the operation instance, and see the additional information about the situations open against the operation instance.

Creating a situation to detect a fault code or string

You can further restrict your definition of unavailability by creating a similar situation that is triggered when a particular fault code or fault string is observed in the message. An example of a situation definition that uses the Fault Code and Fault String attributes in the Fault Log_610 attribute group to specify a particular fault that triggers the situation is displayed in Figure 76.

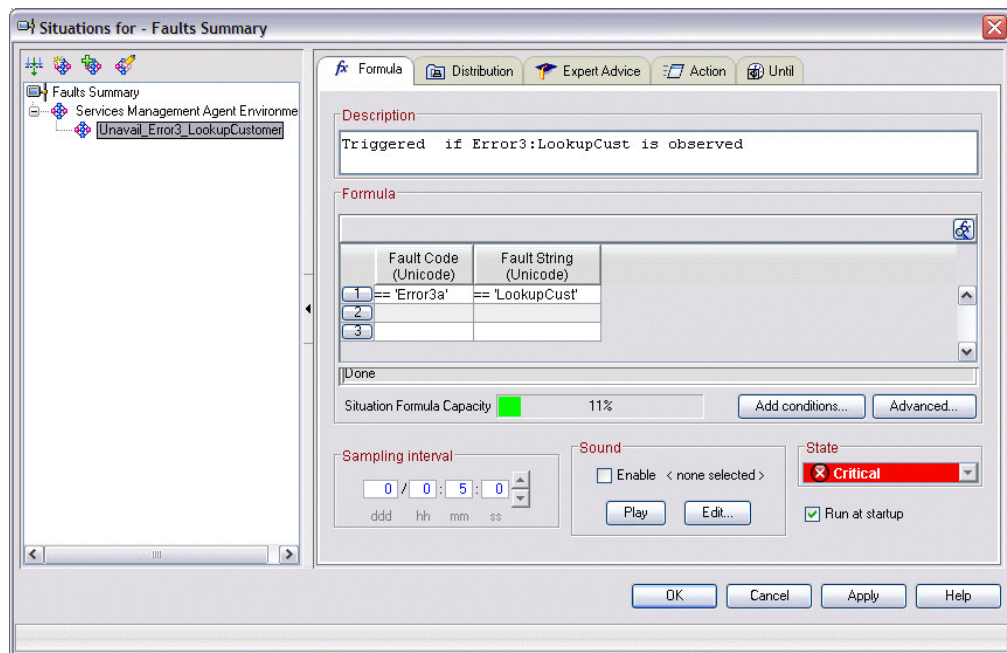


Figure 76. A situation definition specifying a fault code and string.

When this fault code and fault string is observed in the message, the situation is triggered and displayed in the Tivoli Enterprise Portal.

From the flyover window about the triggered situation, follow the link to the Situation Event Results workspace to see what caused the situation to be triggered, and the affected operation. Then, assuming that you created links to the Operational Flow workspace for this situation in the Situation Event Results workspace, you can take that link to the associated Operational Flow workspace to see the topology view.

Displaying the accompanying topology view displays a *Critical* status indicator on the affected operation aggregate. Double-clicking the operation aggregate displays the specific operation instance affected by the situation in the Interaction Detail

portion of the view. The flyover window on the operation instance includes the details about the specific situation that is triggered.

Metrics in the Fault Log table are not accumulated over the entire Tivoli Enterprise Monitoring Agent monitoring interval. Therefore, the situations that are written against this table are triggered when the situation sampling interval expires and there is a matching entry in the Fault Log table. A row can remain in the Fault Log table for up to 10 minutes. Therefore, the situation does not close until the row that matches the situation condition is removed.

Service unavailability metrics

To more accurately determine when a service operation is unavailable because responses are not being sent by the service provider or being received by the service requester, ITCAM for SOA provides the following metrics in the Services Inventory and Services Inventory for Requester ID tables:

Missing Response Percentage (MISSRSPCT)

This metric represents the percentage of request messages that are observed during the monitoring interval that did not have a corresponding response message. A response whose request was observed in a previous interval is not counted. This metric also counts a fault as a response. This differentiates this metric from the Missing Valid Response Percentage metric (see below), which does *not* count a fault as a response.

Missing Valid Response Percentage (MSSVRSPCT)

This metric represents the percentage of request messages that are observed during the monitoring interval that did not have a corresponding *valid* response message. A response message is considered to be valid if it is not a fault. A response whose request was observed in a previous interval is not counted. This metric is different from the Missing Response Percentage metric (see above), which counts a fault as a response.

Valid Response Percentage (VLDRSPCT)

This metric represents the percentage of response messages during the monitoring interval that are not faults. Responses are counted both for requests observed in the current interval and for requests that were observed in the previous interval.

Because the Missing Response Percentage and the Missing Valid Response Percentage metrics are comparing the number of requests against responses, they might not be completely accurate, because a request and its response might not be observed during the same monitoring interval.

For example, as shown in Figure 77 on page 218, a request is sent at 9:03 am and is observed in the monitoring interval from 9:00 am to 9:05 am. If the response is not received until 9:07 am, it is observed and counted (only for the Valid Response Percent metric) in the next monitoring interval (from 9:05 am to 9:10 am).

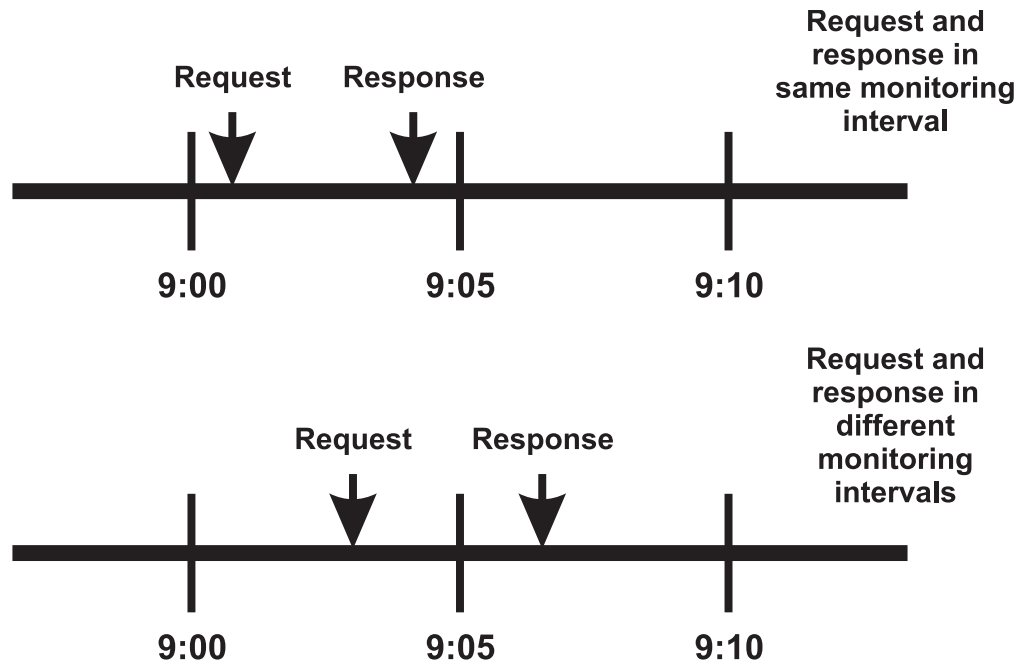


Figure 77. Request and response in the same monitoring interval compared to different monitoring intervals.

To ensure that these two metrics are as accurate as possible given the monitoring interval boundary, the ITCAM for SOA monitoring agent only factors in responses that correspond to requests observed during the same monitoring interval. Two additional metrics are provided to track these responses:

Interval Response Count (INTRSPCNT)

This metric represents the number of response messages that are observed during the monitoring interval and whose corresponding request message is also observed during this same interval. This count includes both valid responses and fault responses. This count is always less than or equal to the value in the Elapsed Time Message Count (ETMSGCOUNT) attribute, because it does not count responses whose requests were observed during a previous monitoring interval.

Interval Fault Count (INTFLTCNT)

This metric represents the number of fault response messages that are observed during the monitoring interval and whose corresponding request message is also observed during this same interval. This count is always less than or equal to the value in the Fault Count (FLTCOUNT) attribute, because it does not count fault responses whose requests were observed in a previous monitoring interval.

Use the new percentage metrics to define a situation that is triggered if the number of missing responses (or missing valid responses) observed during a monitoring interval exceeds some threshold. For example, if 20 percent of the requests observed during a monitoring interval cannot be associated with a corresponding response observed during the same monitoring interval, you might trigger the situation to indicate an unavailability condition.

The situation definition, however, must also include a second threshold for the minimum number of requests that must be observed during the monitoring interval, to ensure that the number of requests is statistically significant. For example, if you only observed four requests during the monitoring interval and

one of them had a missing response, the situation would be triggered because the missing response threshold of 20 percent would be exceeded. However, the number of requests would be statistically insignificant. Therefore, using the Request Count attribute, a second threshold must be defined for the situation:

Request Count (REQCOUNT)

This metric represents the number of request messages that were sent during the monitoring interval.

Figure 78 displays an example situation that you can create, called *Unavail_Missing_Responses*. This situation uses three attributes from the Services Inventory_610 attributes group. The Missing Response Percentage attribute is selected and assigned a threshold value of *greater than or equal to 20* (percent). The Request Count attribute is selected and assigned a threshold value of *greater than or equal to 50*. The Interval Status attribute is selected and set to *Complete* so the situation triggers only when the monitoring interval is completed.

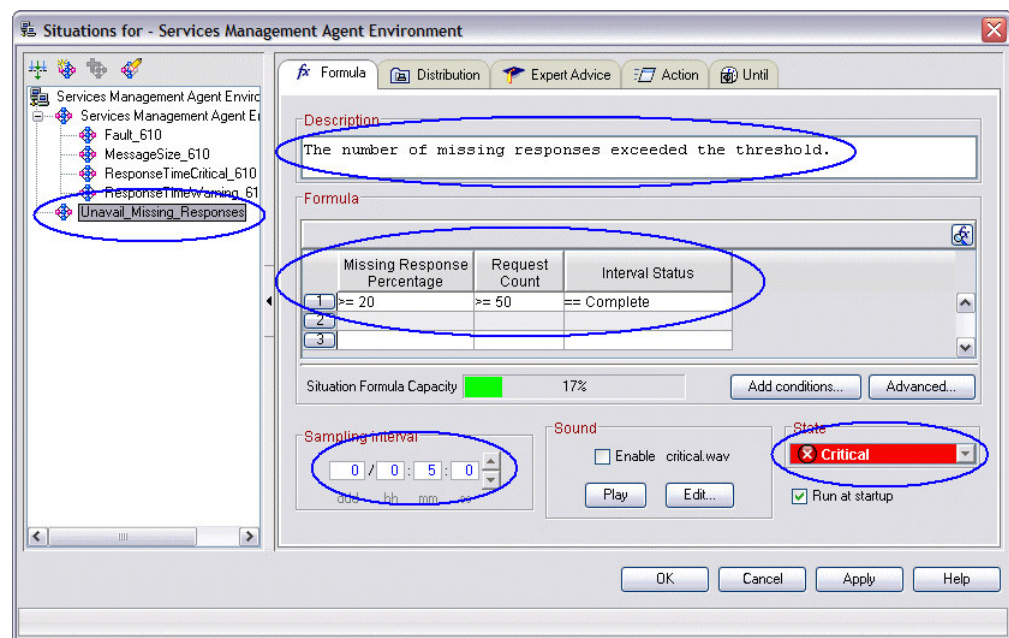


Figure 78. Example situation definition using the unavailability attributes.

This situation is triggered if at least 50 requests are observed during the monitoring interval, and at least 10 of these requests (20% or more) do not have corresponding responses in the same interval. Figure 79 on page 220 displays an example of a customized workspace that includes a customized view called *Service Inventory metrics*. This view displays the various unavailability attributes and metrics in table form. In this example, the situation is triggered because the percentage of missing responses (33% in the Missing Response Percentage column) exceeds the threshold (20%), and the total number of requests (69 in the Request Count column) is large enough to be considered statistically significant.

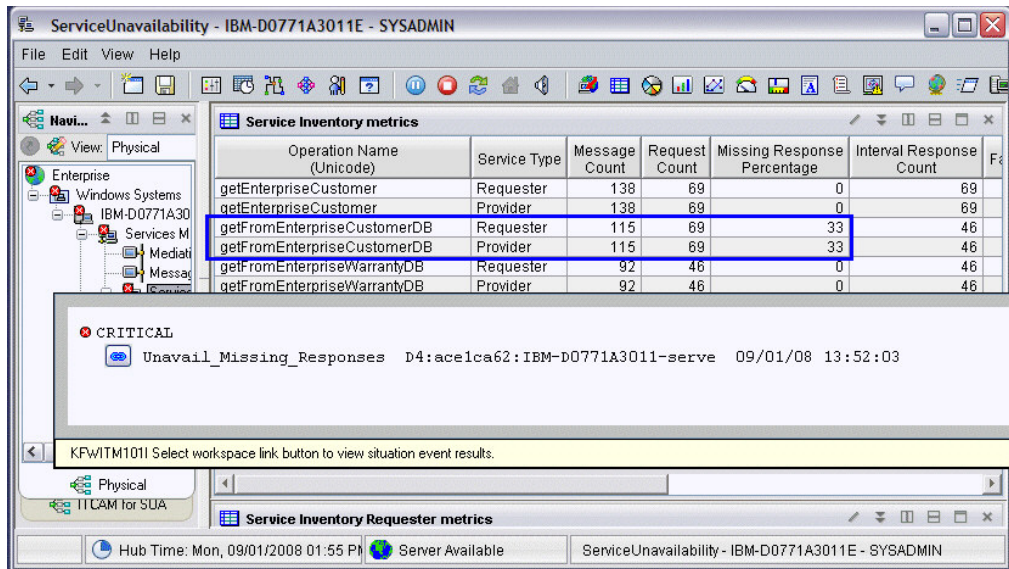


Figure 79. A customized workspace showing metric data for unavailability attributes.

The value in the Message Count column includes the count of *all* request and response messages that were sent during the monitoring interval, *including any responses in the interval whose requests were in a previous interval*. Notice that the interval Response Count includes only those responses that are observed in the monitoring interval that correspond to a request in the *same* interval. For these reasons, the value in the Message Count column might be greater than the sum of the Request Count and Interval Response Count columns.

To see why the situation was triggered, follow the link in the hover help to the Situation Event Results workspace. Assuming that you have a workspace link defined for the situation, you can link from the Situation Event Results workspace to corresponding Operational Flow workspace for this triggered situation.

Additional considerations

Using the various metrics that are provided in the tables, the percentage metrics are set to a value of -1 in the following cases:

- If REQCOUNT is 0 or -1, MISSRSPCT and MSSVRSPCT are set to -1.
- If ETMSGCOUNT is 0 or -1, VLDRSPCT is set to -1.

Displaying unavailability attributes: These unavailability attributes and metrics are not included in the default workspaces and views provided with ITCAM for SOA. However, you can create custom views in Tivoli Enterprise Portal and include these metrics when required. For an example, see Figure 79.

Setting meaningful thresholds: These metrics are useful when the average response time is less than the monitoring interval, and when you can approximate the number of requests whose responses might be observed in a later monitoring interval. To guide you in setting the appropriate thresholds, you might have to collect some sample data.

Operation-specific situations: Your thresholds might differ depending on the operation name. To create situations that measure unavailability on a per operation basis, you can include the operation name in the definition of the situation.

Using IBM Tivoli Performance Analyzer: You might require more complex calculations to determine when a service operation is unavailable. Use IBM Tivoli Performance Analyzer to complete calculations against multiple, existing metrics in the ITCAM for SOA attribute tables and create situations against the analysis results. IBM Tivoli Performance Analyzer uses data that is retrieved from the Tivoli Data Warehouse. Therefore, it is more useful for examining historical data than in providing an indication of a problem in real time.

One-way services: The service unavailability metrics for missing response percentage, missing valid response percentage, and valid response percentage cannot be used to detect unavailability problems for one-way services because they rely on getting response messages.

Using workspace links

You can create your own situation using attributes from the Services Inventory_610 attribute group or Services Inventory Requester Identity_610 attribute group. You might want to link from the Situation Event Results workspace to the Operational Flow for Operation workspace (for a situation where the Service Type is *Provider*), or to the Operational Flow for Application Server workspace (for a situation where the Service Type is *Requester*). To enable such linking, you must create workspace links for each new situation.

For examples, see the links that are provided with the predefined situations. For an example of creating links to Operational Flow workspaces, see Chapter 9, “Creating custom workspaces and links,” on page 185.

The “Linking from the Situation Event Results workspace” on page 195 section describes how to create links from the Initial Situation Values view of the Situation Event Results workspace to a new workspace. Using the same procedure, you can create links from the Initial Values view to the predefined Operational Flow for Operation and Operational Flow for Application Server workspaces. Specify those workspaces as the target workspace rather than a new workspace.

You can also create a link for a row in the Current Situation Values view in the Situation Event Results workspace to the Operational Flow for Operation and Operational Flow for Application Server workspaces. To create the link, start the Link Wizard from the Current Situation Values view.

Chapter 11. Take Action commands

ITCAM for SOA provides several predefined Take Action commands.

About Take Action commands

You can run a Take Action command from your desktop or include it in a situation or policy.

When included in a situation, the command runs when the situation becomes true, and is called a *reflex automation*. When you enable a Take Action command in a situation, you automate a response to system conditions. For example, you can use a Take Action command to send a command to restart a process on the managed system or to send a text message to a cell phone.

Advanced automation uses *policies* to complete actions, schedule work, and automate manual tasks. A policy comprises a series of automated steps called *activities* that are connected to create a *workflow*. After an activity is completed, Tivoli Enterprise Portal receives return code feedback, and advanced automation logic responds with subsequent activities prescribed by the feedback.

A basic Take Action command displays the return code of the operation in a message box that is displayed after the action completes. After you close the window, no further information is available for this action.

For more information about Take Action commands, see the documentation for Tivoli Monitoring.

More information about Take Action commands

For more information about working with Take Action commands, see the *IBM Tivoli Monitoring User's Guide*.

For a list of the Take Action commands for this monitoring agent and a description of each command, refer to "Predefined Take Action commands" and the information in that section for each individual command.

Predefined Take Action commands

ITCAM for SOA has the following Take Action commands:

- Managing filter control settings:
 - "AddFltrCntrl_610 Take Action command" on page 226
 - "DelFltrCntrl_610 Take Action command" on page 238
- Managing monitor control settings:
 - "AddMntrCntrl_610 Take Action command" on page 229
 - "DelMntrCntrl_610 Take Action command" on page 241
- Enabling and disabling monitor and filter controls:
 - "EnableDC_610 Take Action command" on page 245
 - "DisableDC_610 Take Action command" on page 243
- Managing logging and tracing functions:

- “updateLogging_610 Take Action command” on page 249
- “updateTracing_610 Take Action command” on page 251
- “UpdMntrCntrl_610 Take Action command” on page 252
- Managing settings for monitoring by requester identity:
 - “EnableReqIDMntr_610 Take Action command” on page 247
 - “DisableReqIDMntr_610 Take Action command” on page 244
 - “AddRequesterIdentity_610 Take Action command” on page 233
 - “DeleteRequesterIdentity_610 Take Action command” on page 235
 - “SetReqIDTypeHostIP Take Action command” on page 247
 - “SetReqIDTypeUserInfo Take Action command” on page 248
- Managing data collector subnodes:
 - “DeleteSubnode Take Action command” on page 236

Certain Take Action commands are only supported for specific versions of the ITCAM for SOA monitoring agent, as described in Table 35.

Table 35. Take Action commands available for ITCAM for SOA monitoring agents

Take Action	ITCAM for SOA monitoring agent versions supported				Description
	version 6.1.0 and later	version 6.1.0 Fix Pack 1 and later	version 7.1.0 and later	version 7.2 and later	
“AddFltrCntrl_610 Take Action command” on page 226	X	X	X	X	Creates new filter control settings to reject messages
“AddMntrCntrl_610 Take Action command” on page 229	X	X	X	X	Creates new monitor control settings
“DelFltrCntrl_610 Take Action command” on page 238	X	X	X	X	Deletes existing filter control settings
“DelMntrCntrl_610 Take Action command” on page 241	X	X	X	X	Deletes existing monitor control settings
“EnableDC_610 Take Action command” on page 245	X	X	X	X	Enables data collection and the ability to reject messages
“DisableDC_610 Take Action command” on page 243	X	X	X	X	Disables data collection and the ability to reject messages
“updateLogging_610 Take Action command” on page 249	X	X	X	X	Defines the level of logging information
“UpdMntrCntrl_610 Take Action command” on page 252	X	X	X	X	Updates existing message logging levels for monitor control
“updateTracing_610 Take Action command” on page 251	X	X	X	X	Enables or disables tracing

Table 35. Take Action commands available for ITCAM for SOA monitoring agents (continued)

Take Action	ITCAM for SOA monitoring agent versions supported				Description
	version 6.1.0 and later	version 6.1.0 Fix Pack 1 and later	version 7.1.0 and later	version 7.2 and later	
“EnableReqIDMntr_610 Take Action command” on page 247		X	X	X	Enables (turns on) the aggregation of data collection for all web service requester identities that you want to monitor
“DisableReqIDMntr_610 Take Action command” on page 244		X	X	X	Disables (turns off) the aggregation of data collection for all web service requester identities being monitored.
“AddRequesterIdentity_610 Take Action command” on page 233		X	X	X	Add a requester identity to the list of requester identities in the Requester Identity Monitoring Configuration workspace.
“DeleteRequesterIdentity_610 Take Action command” on page 235		X	X	X	Deletes a requester identity from the list of requester identities in the Requester Identity Monitoring Configuration workspace.
“SetReqIDTypeHostIP Take Action command” on page 247			X	X	Enables the host name or IP address value for the requester identity
“SetReqIDTypeUserInfo Take Action command” on page 248			X	X	Enables the user credential value for the requester identity
“DeleteSubnode Take Action command” on page 236			X	X	If an ITCAM for SOA data collector is no longer deployed to monitor an application server environment, use the DeleteSubnode action to instruct the ITCAM for SOA monitoring agent to remove information about operation instances associated with a specified data collector subnode.

The following information is provided about each Take Action command:

Description

A description of the actions that the command performs on the system to which it is sent.

Sending the action

The step by step procedure for selecting the Take Action command from the available list, and any special instructions for specifying parameters

Arguments

The list of parameters, if any, for the Take Action command with a short description and default value, if available.

Return codes

Information that the Take Action command returns

AddFiltrCntrl_610 Take Action command

Description

The **AddFiltrCntrl_610** Take Action command is used to define the criteria to control the start and stop of services by rejecting messages that flow through the data collector to a service operation. The only available filter control is *reject*.

You can reject the message based on a specific caller host name or IP address to a specific service port name (also referred to as the Web Services Description Language (WSDL) port name) and operation name. Each message is evaluated based on the filtering criteria you specify, and if the message matches the filter criteria, it is rejected and not allowed to flow through the data collector. When a message is rejected, an entry is written to a monitoring action log file.

This filter criteria is displayed in the Data Collector Filter Control Configuration table view in the Services Management Agent workspace. Each row of the table represents a unique filter criteria definition. For more information about the columns in this table, see “Data Collector Filter Control_610 attributes” on page 261.

Important: Certain interception points exist where the IP address or host name is not available from the message. In these cases, if a specific host name or IP address is indicated in the filter criteria, the message is not rejected based on those criteria, and is allowed to flow through. However, if the host name and IP address are configured with the asterisk (*) wildcard value, then messages from all host names or IP addresses are rejected.

You can specify the filtering criteria as a host name or IP address and some combination of service port name, service port namespace, operation name or operation namespace. Each of these fields must specify either the full value, or the asterisk (*) wildcard value, which matches all values for that criteria. Partial expressions are not valid. For example, you can specify an operation name of *lookupCustomer* but not *lookup**. The values are not validated against existing service port names or operation names.

You can define multiple combinations of filter criteria to apply to a specific application server runtime environment. Multiple filter criteria are evaluated in a specific order, from least specific to more specific.

Table 36 on page 227 lists the valid sample filter criteria combinations. Any other possible combinations not listed in this table are not valid and are rejected with a return code of 2.

Table 36. Filter criteria combinations, in order of least specific to more specific.

Remote IP address or host name	Service port namespace	Service port	Operation namespace	Operation	Priority
*	*	*	*	*	14- This criteria is the least specific and is evaluated first
*	*	XYZ	*	*	13
*	*	XYZ	*	XYZ	12
*	*	XYZ	XYZ	XYZ	11
*	XYZ	XYZ	*	*	10
*	XYZ	XYZ	*	XYZ	9
*	XYZ	XYZ	XYZ	XYZ	8
IP/Hostname	*	*	*	*	7
IP/Hostname	*	XYZ	*	*	6
IP/Hostname	*	XYZ	*	XYZ	5
IP/Hostname	*	XYZ	XYZ	XYZ	4
IP/Hostname	XYZ	XYZ	*	*	3
IP/Hostname	XYZ	XYZ	*	XYZ	2
IP/Hostname	XYZ	XYZ	XYZ	XYZ	1

In Table 36, XYZ represents an instance of the service port name, service port namespace, operation name, or operation namespace. *IP/Hostname* represents a specific IP address or host name. To specify multiple client IP addresses, define a new filter criteria for each specific IP address to be filtered. You cannot specify a subnet.

The Filtering Control Index column displays a value used to uniquely identify each filter control within each combination of application server and application server runtime environment.

If a host name is specified instead of an IP address, it is converted to its equivalent IP address in the filter criteria. The host name is also added to the criteria, and the filtering criteria is matched as specified to the incoming message. For example, if you specify a host name of *mycomputer.domain.com*, the incoming message must contain the exact same string or the corresponding IP address. If you specify an IP address, the corresponding host name or host names are also added to the internal filtering criteria.

Important: You cannot edit a filter control after you defined it and added it to the table. You must use either the **DelFltrCntrl_610** to delete the filter criteria, and then use **AddFltrCntrl_610** again to add a new definition.

Sending the action

To send the **AddFltrCntrl_610** Take Action command, complete these steps:

1. From one of the views in the current workspace of the Tivoli Enterprise Portal, right-click a bar in a bar chart or in a row in a table view to select it. Select **Take Action->Select** to display the Take Action window.

2. Under Action, to display the list of available actions, click the **Name** field. The list of available actions is displayed.
3. In the list, click **AddFiltrCtrl_610**.
4. The **Command** field of the Take Action window is partially completed with the command syntax, and the Edit Argument Values window is displayed, for you to enter the filtering criteria. The fields in the **Value** column are pre-filled with data from the Services Inventory_610 attributes table or the Data Collector Filter Control_610 attributes table. Enter the values that you want to filter for each argument. Use the asterisk (*) wildcard value to match all messages for that argument.
5. After entering the values, click **OK**. The Edit Argument Values window closes, and you are returned to the Take Action window, with the entered filter criteria now included in the **Command** field.
6. The **Destination System(s)** field contains one or more target systems where you can send the command. Click the wanted system name and then click **OK**.
7. The Action Status window is displayed with the resulting return code. To close the window, click **OK**. The Take Action window also closes.
8. If you received a successful return code, refresh the display of the Services Management Agent workspace to see your new filter criteria included as a new row in the Data Collector Filter Control Configuration table view.

After defining filter controls, subsequent messages that match the filter criteria are rejected.

Arguments

Table 37 describes the arguments that can be specified for this Take Action command.

Table 37. Arguments for the AddFiltrCtrl_610 Take Action command.

Name	Value
Application_ServerName_U	The name of the application server (for example, <i>server1</i> for a WebSphere environment)
Application_ServerEnv	The application server environment. The valid values are: <ul style="list-style-type: none"> • WebSphere Application Server • .NET • WebLogic Server • JBoss • CICS • SAP • WebSphere Community Edition
Local_Hostname_U	The local host name. Either the full host name or the asterisk (*) wildcard value.

Table 37. Arguments for the AddFtrCntrl_610 Take Action command. (continued)

Name	Value
Port_Namespace_U	The service port namespace (also known as the Web Services Description Language port namespace). Either the full service port namespace or the asterisk (*) wildcard value, according to the limitations that are described in the filter criteria combinations table.
Service_Name_U	The service port name (also known as the Web Services Description Language port name). Either the full service port name or the asterisk (*) wildcard value, according to the limitations that are described in the filter criteria combinations table.
Operation_Namespace_U	The operation namespace associated with the specified service. Either the full operation namespace or the asterisk (*) wildcard value, according to the limitations that are described in the filter criteria combinations table.
Operation_Name_U	The operation name associated with the specified service. Either the full operation name or the asterisk (*) wildcard value, according to the limitations that are described in the filter criteria combinations table.
RemoteIPAddress	The remote IP address. This field can also be a host name, which is then converted to its equivalent IP address. Either the full IP address or host name, or the asterisk (*) wildcard value, according to the limitations that are described in the filter criteria combinations table. You cannot specify subnets.

Return codes

When you run the Take Action command, you receive one of the following return codes:

- 0 = The action completed successfully and the filter control was added to the filter control table.
- 1 = No change was made to the filter criteria.
- 2 = The action completed unsuccessfully. There might be an error in your filter criteria. The filter criteria were not added to the filter control table.

AddMntrCntrl_610 Take Action command

Description

The **AddMntrCntrl_610** Take Action is used to control the information that is collected by the ITCAM for SOA monitoring agents. You can specify the

command so that only data that is associated with certain combinations of service port name and operation name is collected.

You can choose to record individual messages based on a specific service port name (also referred to as the Web Services Description Language (WSDL) port name) and operation name, and you can specify the amount of data that is logged (none, header information only, body information only, or both header and body information). Each message is evaluated based on the filtering criteria that you specify, and if the message matches the filter criteria, then monitoring data for that message is collected and logged.

The data is recorded in a separate log file in the ITCAM for SOA log directory, named `content.log`. You can use the data with the Web Service Navigator tool.

This monitor criteria is displayed in the Data Collector Monitor Control Configuration table of the Services Management Agent workspace. Each row of the table represents a unique monitor criteria definition. For more information about the columns in this table, see “Data Collector Monitor Control_610 attributes” on page 264.

When you specify the monitoring criteria as a combination of service port name and operation name, each of these fields must specify either the full value, or the asterisk (*) wildcard value, which matches all values for that criteria. Partial expressions are not valid (for example, you can specify an operation name of `lookupCustomer` but not `lookup*`).

You can define multiple combinations of monitor criteria to apply to a specific application server runtime environment. Multiple monitor criteria are evaluated in a specific order, from most specific to least specific.

Table 38 lists the valid monitor criteria combinations. Any other possible combinations not listed in this table are not valid and is rejected with a return code of 2.

Table 38. Monitor criteria combinations, in order of most specific to least specific.

Service port namespace	Service port	Operation namespace	Operation	Priority
XYZ	XYZ	XYZ	XYZ	7- This criteria is most specific and is evaluated first.
XYZ	XYZ	*	XYZ	6
XYZ	XYZ	*	*	5
*	XYZ	XYZ	XYZ	4
*	XYZ	*	XYZ	3
*	XYZ	*	*	2
*	*	*	*	1

In Table 38, XYZ represents an instance of the service port namespace, service port, operation namespace, or operation name. The criteria is evaluated in order of most specific to least specific because as soon as a criteria is satisfied, the agent logs the information for that message, and further criteria are not evaluated.

The Monitoring Control Index column displays a value that is used to uniquely identify each monitor control within each combination of application server and application server runtime environment.

Restriction: You cannot edit a monitor control (other than the `Message_Logging_Level` argument) after you have defined it and added it to the table. The level of monitoring can be adjusted, using the **UpdMntrCntrl_610** command. If you want to edit a field other than the level of monitoring, you must use the **DelMntrCntrl_610** command to delete the monitor criteria, and then use **AddMntrCntrl_610** again to add a definition.

Sending the action

To run the **AddMntrCntrl_610** action, complete the following steps:

1. From one of the views in the current workspace of the Tivoli Enterprise Portal, right-click a bar in a bar chart or in a row in a table view to select it. Select **Take Action->Select** to display the Take Action window.
2. Under Action, to display the list of available actions, click in the **Name** field.
3. From the list of available actions, click **AddMntrCntrl_610**.
4. The **Command** field in the Take Action window is partially completed with the command syntax, and the Edit Argument Values window displays for you to enter the monitoring criteria. The fields in the **Value** column are pre-filled with data from the `Services Inventory_610` attributes or the `Data Collector Monitor Control_610` attributes. Enter or modify the values for the monitoring criteria for each argument. Use the asterisk (*) wildcard value in the fields for service port name or operation name to match all messages for that argument. For the `Message_Logging_Level` argument, you must specify one of the valid values (wildcard values are not allowed).
5. After entering the values, click **OK**. The Edit Argument Values window closes, and you are returned to the Take Action window, with the entered monitor criteria now included in the **Command** field.
6. The **Destination System(s)** field contains one or more target systems where you can send the command. Click the wanted system name and then click **OK**.
7. The Action Status window displays with the resulting return code. To close the window, click **OK**. The Take Action window also closes.
8. If you received a successful return code, refresh the display of the Services Management Agent workspace to see your new monitor criteria included as a new row in the Data Collector Monitor Control Configuration table view.

After defining monitor controls, data for subsequent messages that match the monitor criteria are collected. A message that does not match any of the criteria is ignored.

Arguments

Table 39 describes the arguments that you can specify for this Take Action command.

Table 39. Arguments for the AddMntrCntrl_610 Take Action command.

Name	Value
------	-------

Table 39. Arguments for the AddMntrCntrl_610 Take Action command. (continued)

Application_ServerName_U	The name of the application server (for example, <i>server1</i> for a WebSphere environment)
Application_ServerEnv	The application server environment. The valid values are: <ul style="list-style-type: none"> • WebSphere Application Server • .NET • WebLogic Server • JBoss • DataPower • CICS • SAP • WebSphere Community Edition • WebSphere Message Broker
Local_Hostname_U	The local host name. Either the full host name or the asterisk (*) wildcard. This field is ignored.
Port_Namespace_U	The service port namespace (also known as the Web Services Description Language port namespace). Either the full service port namespace or the asterisk (*) wildcard value, according to the limitations that are described in the monitor criteria combinations table.
Service_Name_U	The service port name (also known as the Web Services Description Language port name). Either the full service port name or the asterisk (*) wildcard value, according to the limitations that are described in the monitor criteria combinations table.
Operation_Namespace_U	The operation namespace associated with the specified service. Either the full operation namespace or the asterisk (*) wildcard value, according to the limitations that are described in the monitor criteria combinations table.
Operation_Name_U	The operation name associated with the specified service. Either the full operation name or the asterisk (*) wildcard value, according to the limitations that are described in the monitor criteria combinations table.

Table 39. Arguments for the AddMntrCntrl_610 Take Action command. (continued)

DataCollectorMessageLoggingLevel	<p>The amount of data that is to be collected for messages that match the monitoring criteria. The valid values are:</p> <p>None No information is logged.</p> <p>Header Only message header information is logged.</p> <p>Body Only the body of the message is logged.</p> <p>Full Both the header and body of the message are logged.</p>
----------------------------------	---

Return codes

When you run this Take Action command, you receive one of the following return codes:

- 0 = The action completed successfully and the monitor control was added to the monitor control table.
- 1 = No change was made to the monitoring criteria.
- 2 = The action completed unsuccessfully. There might be an error in your monitor criteria. The monitor criteria were not added to the monitor control table.

AddRequesterIdentity_610 Take Action command

Description

Use the **AddRequesterIdentity_610** Take Action command to add or include a specified web service requester identity to a list of identities that are being monitored.

The web service requester identity is displayed in the Monitored Requester Identities view of the subsidiary Requester Identity Monitoring Configuration workspace.

Considerations for specifying the requester identity: Keep in mind these considerations when specifying user IDs as requester identities:

- **Monitor only the requester identities that you need:** The monitoring agent must store information in memory for every operation started, and for every configured requester identity. If you monitor many identities (perhaps by using the asterisk (*) wildcard), you can significantly increase the memory that is used by the monitoring agent.
- The collection of monitored data by remote host name or IP address depends on the interfaces that are provided by the vendor application server. Be sure that you understand the types, values, and formats of requester identities that you intend to use before monitoring by requester identity.
- If you are monitoring requester identities by remote host name or IP address, the host name is used as the requester identity by default. If the host name is not available from the application server environment, the remote IP address is used as the requester identity. If both the remote host name and remote IP address are not available, the message is not monitored by requester identity, unless you specify the asterisk (*) wildcard character as the requester identity.

- The list of configured requester identities is maintained as a case-sensitive list because the same Take Action commands are used to add and delete both types of requester identities. However, host names are treated as not case-sensitive by the monitoring agent when it determines if the remote host that sent a message matches a configured service requester identity.

Sending the action

To run the `AddRequesterIdentity_610` action, complete the following steps:

1. From the Tivoli Enterprise Portal, navigate to the Services Management Agent workspace.
2. Right-click the **Services Management Agent** node and select **Workspace -> Requester Identity Monitoring Configuration**.
3. In the Monitored Requester Identities view, right-click and select **Take Action -> Select** to display the Take Action window.
4. To display the list of available actions for this agent, in the **Action** area, click the **Name** field.
5. From the list of available actions, click **AddRequesterIdentity_610**.
6. The **Command** field is completed with the command syntax. The Edit Argument Values window opens.
7. After editing the argument, click **OK**. The Edit Argument Values window is closed, and you are returned to the Take Action window, with the entered monitor criteria now included in the **Command** field.
8. The **Destination System(s)** field contains one or more target systems where you can send the command. Click the wanted system name and then click **OK**.
9. The Action Status window displays with the resulting return code. Click **OK** to close the window. The Take Action window also closes.

Arguments

Table 40 on page 235 describes the arguments that you can specify for this Take Action command.

Table 40. Arguments for the AddRequesterIdentity_610 Take Action command

Name	Value
ReqID_Mntr_Ctrl_610.Requester_Identity_U	<p>The requester identity that you are adding or including to the list of identities.</p> <p>The asterisk (*) wildcard character indicates that all requester identities can be monitored.</p> <p>Partial matching is not performed. "*.ibm.com" is treated as a literal string, not as a regular expression. Only the single asterisk character by itself is treated as a wildcard.</p> <p>If the requester identity type is host name or IP address and IPv6 addresses are being used for the requester identity, the IPv6 addresses entered for this Take Action command must match the format of the IPv6 addresses in the ITCAM for SOA data collector metric log files.</p> <p>If the requester identity type is user ID, the requester identity is case-sensitive (for example, <i>joe@us.ibm.com</i> is different from <i>JOE@us.ibm.com</i> in some user registries) so you must enter it in the case that is used when web service requests are issued.</p>

Return codes

When you run the Take Action command, you might receive one of the following return codes:

- 0 = The action completed successfully and the requester identity was added to the list of identities that are being monitored.
- 2 = The action completed unsuccessfully and the list of identities that are being monitored was not changed.

For example, you entered a requester identity that was not valid, used a bad configuration file, or the control for the requester identity exists.

DeleteRequesterIdentity_610 Take Action command

Description

The **DeleteRequesterIdentity_610** action is used to delete a specified web service requester identity from a list of identities that are being monitored.

Web service requester identities that are being monitored are displayed in the Monitored Requester Identities view of the subsidiary Requester Identity Monitoring Configuration workspace.

When you delete a requester identity from the list, you are only stopping future data from being collected for that requester identity. Any data that was previously collected for the requester identity is not immediately removed from tables, but is eventually removed because of the normal aging process (for example, associated metric data is displayed with values of -1 after the five-minute interval expires).

Sending the action

To run the DeleteRequesterIdentity_610 action, complete the following steps:

1. From the Tivoli Enterprise Portal, navigate to the Services Management Agent workspace.
2. Right-click the **Services Management Agent** node and select **Workspace -> Requester Identity Monitoring Configuration**.
3. To display the Take Action window, in the Monitored Requester Identities view, select the identity row, and right-click and select **Take Action -> Select**.
4. In the **Action** area, to display the list of available actions for this agent, click the **Name** field.
5. From the list of available actions, click **DeleteRequesterIdentity_610**. The **Command** field is completed with the command syntax and the requester identity value from the command row that you selected.
6. The Edit Argument Values window is pre-filled with the requester identity value from the identity row that you selected. Click **Arguments** to change the requester identity. Click **OK** and the Edit Argument Values window is closed, and you are returned to the Take Action window, with the entered monitor criteria now included in the **Command** field.
7. The **Destination System(s)** field contains one or more target systems where you can send the command. Click the wanted system name and then click **OK**.
8. The Action Status window displays with the resulting return code. To close the window, click **OK**. The Take Action window also closes.

Arguments

Table 41 describes the arguments that you can specify for this Take Action command.

Table 41. Arguments for the DeleteRequesterIdentity_610 Take Action command

Name	Value
ReqID_Mntr_Ctrl_610.Requester_Identity_U	The requester identity that you are deleting from the list of identities.

Return codes

When you run the Take Action command, you might receive one of the following return codes:

- 0 = The action completed successfully and the requester identity was removed from the list of identities that are being monitored.
- 1 = The action completed unsuccessfully because the requester identity was not found in the list of identities that are being monitored. Requester identities are case-sensitive. Verify that the requester identity was defined with the right case sensitivity.
- 2 = The action completed unsuccessfully and the list of identities that are being monitored was not changed.

DeleteSubnode Take Action command

Description

Data collector subnodes for ITCAM for SOA monitoring agents at version 7.1.0 or later are displayed as being online if the monitoring agent for that computer system is online. If you uninstalled an application server or disabled data collection for an application server and you do not plan to uninstall the ITCAM for SOA monitoring agent because you are monitoring another application server on that computer system, use the **DeleteSubnode** take action to instruct the monitoring agent to remove information for all operation instances and relationships that are associated with the data collector subnode.

Changing the time interval to allow a subnode to be deleted: The data collector subnode cannot be deleted using this Take Action command unless the data collector for that subnode has not observed any message traffic for the past 24 hours. Adjust this time frame to set the number of hours that a subnode must be inactive before allowing the subnode to be deleted. The default time interval is 24 hours.

To change the time interval, create a new Take Action command for the Services Management Agent monitored application that changes the value of the `kd4.ira.subnodeDeleteInactivity` property for the monitoring agent. The Take Action specifies this command:

```
setKeyVal710 kd4.ira.subnodeDeleteInactivity=x
```

In this command, *x* is the number of hours that the data collector node must be inactive before its data can be deleted using **DeleteSubnode**. Then, issue the new Take Action command for each ITCAM for SOA monitoring agent where you want to change the inactivity time.

Important: The monitoring agent does not remove metric log files for the subnode when it processes the **DeleteSubnode** Take Action command. If you reduce the inactivity time to 2 hours or less, there might still be metric log files for the subnode in the `KD4/logs/KD4.DCA.CACHE` directory for the monitoring agent. In this case, the subnode might be recreated if the monitoring agent is restarted at some later point.

Sending the action

To run the **DeleteSubnode** action, complete the following steps:

1. From the Tivoli Enterprise Portal, in the Navigator Physical view, right-click the subnode to be deleted and select **Take Action -> Select** to display the Take Action window.
2. To display the list of available actions for this node, in the **Action** area, click the **Name** field.
3. From the list of available actions, click **DeleteSubnode**.
4. The data collector subnode you selected in step 1 should already be selected in the **Destination Systems** field. To issue the take action command, click **OK**.
5. The Action Status window displays with the resulting return code. To close the window, click **OK**. The Take Action window also closes.

After the **DeleteSubnode** Take Action command completes successfully, the data collector subnode is displayed as offline in the Navigator Physical view. You can then remove this subnode from the list of subnodes by right-clicking the offline subnode, and selecting the **Clear offline entry** option.

If the data collector is later redeployed and begins monitoring web service traffic, the data collector subnode becomes active and is displayed again in the Navigator Physical view.

After successfully removing the data collector subnode from the Navigator Physical view, to permanently remove the subnode, complete these additional steps:

1. Run the `deleteUnmanagedSubnodes` script to remove the operation instances for the deleted data collector subnode from the service-to-service topology views. For more information, see “Deleting unmanaged subnodes” on page 123.
2. If you are using the ITCAM for SOA Discovery Library Adapter to discover service data for the static topology views, run the DLA with the **refresh** option. This action creates a DLA book that does not contain information about the service ports and operations associated with the deleted data collector subnode.

After you load the new refreshed book into the Tivoli Common Object Repository using the bulk load program, the static topology views do not show service ports and operations for the deleted data collector node.

Arguments

There are no arguments that you can specify for this Take Action command.

Return codes

When you run the Take Action command, you might receive one of the following return codes:

- 0 = The action completed successfully and data for the selected data collector subnode was removed by the ITCAM for SOA monitoring agent.
- 1 = The action completed unsuccessfully because the subnode was not found in the list of application server environments that are being monitored.
- 2 = The action completed unsuccessfully and the list of application server environments that are being monitored was not changed.
- 3 = The action completed unsuccessfully because the data collector node has not been inactive for 24 hours (or for the number of hours specified by the `kd4.ira.subnodeDeleteInactivity` monitoring agent property).

DelFiltrCntl_610 Take Action command

Description

The **DelFiltrCntl_610** action is used to delete filter controls that were defined with the **AddFiltrCntl_610** Take Action command.

Filter controls are displayed in the Data Collector Filter Control Configuration table view in the Services Management Agent workspace. Each row of the table represents a unique filter criteria definition. For more information about the columns in this table, see “Data Collector Filter Control_610 attributes” on page 261.

To delete a filter control from the Data Collector Filter Control Configuration table, right-click the row and select the **DelFiltrCntl_610** action from the Take Action command menu.

After you delete a filter control from the Data Collector Filter Control Configuration table view, the index numbers in the Filtering Control Index column are adjusted as required to maintain the order of evaluation from least specific to most specific.

Sending the action

To run the **DelFiltrCntrl_610** action, complete the following steps:

1. From one of the views in the current workspace of the Tivoli Enterprise Portal, right-click a bar in a bar chart or in a row in a table view to select it. Select **Take Action->Select** to display the Take Action window.
2. To display the list of available actions, under Action, click in the **Name** field.
3. From the list of available actions, click **DelFiltrCntrl_610**. The **Command** field in the Take Action window is partially completed with the command syntax, and the Edit Argument Values window is displayed for you to enter the filtering criteria. The fields in the **Value** column are pre-filled with data from the Services Inventory_610 attributes group or the Data Collector Filter Control_610 attributes group. Enter or modify the values for the filtering criteria for each argument. Use the asterisk (*) wildcard value in the fields for service port name or operation name to match all messages for that argument.
4. After entering the values, click **OK**. The Edit Argument Values window is closed, and you are returned to the Take Action window, with the entered filter criteria now included in the **Command** field.
5. The **Destination System(s)** field contains one or more target systems where you can send the command. Click the wanted system name and then click **OK**.
6. The Action Status window displays with the resulting return code. To close the window, click **OK**. The Take Action window also closes.
7. If you received a successful return code, refresh the display of the Services Management Agent workspace to verify that the filter control was removed from the Data Collector Filter Control Configuration table view. The index numbers in the Filtering Control Index column are automatically adjusted to maintain the order of filter control evaluation from least specific to most specific.

After removing filter controls, subsequent messages that match the remaining filter criteria are rejected. If all filter controls are removed, all messages are allowed to flow through the data collector.

Arguments

Table 42 describes the arguments that can be specified for this Take Action command.

Table 42. Arguments for the **DelFiltrCntrl_610** Take Action command.

Name	Value
Application_ServerName_U	The name of the application server (for example, <i>server1</i> for a WebSphere environment)

Table 42. Arguments for the DelFtrCntrl_610 Take Action command. (continued)

Application_ServerEnv	The application server environment. The valid values are: <ul style="list-style-type: none"> • WebSphere Application Server • .NET • WebLogic Server • JBoss • CICS • SAP • WebSphere Community Edition
Hostname_U	The local host name. Either the full host name or the asterisk (*) wildcard value.
Port_Namespace_U	The service port namespace (also known as the Web Services Description Language port namespace). Either the full service port namespace or the asterisk (*) wildcard value, according to the limitations that are described in the filter criteria combinations table.
Service_Name_U	The service port name (also known as the Web Services Description Language port name). Either the full service port name or the asterisk (*) wildcard value, according to the limitations that are described in the filter criteria combinations table.
Operation_Namespace_U	The operation namespace associated with the specified service. Either the full operation namespace or the asterisk (*) wildcard value, according to the limitations that are described in the filter criteria combinations table.
Operation_Name_U	The operation name associated with the specified service. Either the full operation name or the asterisk (*) wildcard value, according to the limitations that are described in the filter criteria combinations table.
RemoteIPAddress	The remote IP address. This field can also be a host name, which is then converted to its equivalent IP address. Either the full IP address or host name or the asterisk (*) wildcard value, according to the limitations that are described in the filter criteria combinations table. You cannot specify subnets.

Return codes

When you run this Take Action command, you receive one of the following return codes:

- 0 = The action completed successfully and the filter control was removed from the filter control table.
- 1 = No change was made to the filtering criteria.
- 2 = The action completed unsuccessfully. The filter criteria were not removed from the filter control table.

DelMntrCntrl_610 Take Action command

Description

The **DelMntrCntrl_610** action is used to delete the monitor controls that have been defined with the **AddMntrCntrl_610** action.

Monitor controls are displayed in the Data Collector Monitor Control Configuration table view in the Services Management Agent workspace. Each row of the table represents a unique monitoring criteria definition. For more information about the columns in this table, see the “Data Collector Monitor Control_610 attributes” on page 264.

To delete a monitor control from the Data Collector Monitor Control Configuration table, right-click the row and select **DelMntrCntrl_610** from the Take Action command menu. After you delete a monitor control from the Data Collector Monitor Control Configuration table view, the index numbers in the Monitoring Control Index column are adjusted as required to maintain the order of evaluation from most specific to least specific.

Sending the action

To run the **DelMntrCntrl_610** action, complete the following steps:

1. From one of the views in the current workspace of the Tivoli Enterprise Portal, right-click a bar in a bar chart or in a row in a table view to select it. Select **Take Action->Select** to display the Take Action window.
2. Under Action, click in the **Name** field to display the list of available actions.
3. From the list of available actions, click **DelMntrCntrl_610**. The **Command** field in the Take Action window is partially completed with the command syntax, and the Edit Argument Values window is displayed for you to enter the filtering criteria. The fields in the **Value** column are pre-filled with data from the Services Inventory_610 attributes group or the Data Collector Monitor Control_610 attributes group. Enter the values, and click **OK**. The Edit Argument Values window closes, and you are returned to the Take Action window, with the entered filter criteria now included in the **Command** field.
4. The **Destination System(s)** field contains one or more target systems where you can send the command. Click the wanted system name and then click **OK**.
5. The Action Status window is displayed with the resulting return code. To close the window, click **OK**. The Take Action window also closes.
6. If you received a successful return code, refresh the display of the Services Management Agent workspace to verify that the monitor control was removed from the Data Collector Monitor Control Configuration table view. The index numbers in the Monitoring Control Index column are automatically adjusted to maintain the order of monitor control evaluation from most specific to least specific.

After removing monitor controls, data for subsequent messages that match the remaining monitor criteria are logged.

Arguments

Table 43 describes the arguments that you can specify for this Take Action command.

Table 43. Arguments for the *DelMntrCntrl_610* Take Action command

Name	Value
Application_ServerName_U	The name of the application server (for example, <i>server1</i> for a WebSphere environment)
Application_ServerEnv	The application server environment. The valid values are: <ul style="list-style-type: none">• WebSphere Application Server• .NET• WebLogic Server• JBoss• DataPower• CICS• SAP• WebSphere Community Edition• WebSphere Message Broker
Hostname_U	The local host name. Either the full host name or the asterisk (*) wildcard. This field is ignored.
Port_Namespace_U	The service port namespace (also known as the Web Services Description Language port namespace). Either the full service port namespace or the asterisk (*) wildcard value, according to the limitations that are described in the monitor criteria combinations table.
Service_Name_U	The service port name (also known as the Web Services Description Language port name). Either the full service port name or the asterisk (*) wildcard value, according to the limitations that are described in the monitor criteria combinations table.
Operation_Namespace_U	The operation namespace associated with the specified service. Either the full operation namespace or the asterisk (*) wildcard value, according to the limitations that are described in the monitor criteria combinations table.
Operation_Name_U	The operation name associated with the specified service. Either the full operation name or the asterisk (*) wildcard value, according to the limitations that are described in the monitor criteria combinations table.

Return codes

When you run this Take Action command, you receive one of the following return codes:

- 0 = The action completed successfully and the monitor control was removed from the monitor control table.

- 1 = No change was made to the monitor criteria.
- 2 = The action completed unsuccessfully. The monitor criteria were not removed from the monitor control table.

DisableDC_610 Take Action command

Description

The **DisableDC_610** action is used to disable the data collector in an application server environment and to disable filtering. Use this command to selectively turn off data collection and filtering of certain message traffic.

The data collector configuration status is displayed in the Data Collector Global Configuration table in the Services Management Agent workspace. Each row of the table represents a unique data collector for a particular application server runtime environment. The **Data Collector On/Off** column indicates whether the data collector is enabled or disabled. For more information about the columns in this table, see the “Data Collector Global Configuration_610 attributes” on page 263.

To disable a data collector, you must set the value in the **Data Collector On/Off** column to **Off**. To make this change, right-click the row in the Data Collector Global Configuration table and, from the Take Action command menu, select the **DisableDC_610** action.

After you disable a data collector in the Data Collector Global Configuration table, subsequent message traffic for that data collector is turned off until you enable it again using the **EnableDC_610** Take Action command.

Sending the action

To run the **DisableDC_610** action, complete the following steps:

1. From the Tivoli Enterprise Portal, navigate to the Services Management Agent workspace.
2. In the Data Collector Global Configuration table, right-click the row in the table for the data collector to be disabled (the value in the **Data Collector On/Off** column must be set to **On**) and select **Take Action -> Select** to display the Take Action window.
3. To display the list of available actions, under Action, click in the **Name** field.
4. From the list of available actions, click **DisableDC_610**.
5. The Take Action window is displayed and the selected data collector information included in the **Command** field. Either Accept the argument values for the command, or, to edit the argument values, click **Arguments**.
6. The **Destination System(s)** field contains one or more target systems where you can send the command. Click the wanted system name and then click **OK**.
7. The Action Status window displays with the resulting return code. To close the window, click **OK**. The Take Action window closes.
8. If you received a successful return code, refresh the display of the Services Management Agent workspace to verify that the value in the **Data Collector On/Off** column in the Data Collector Global Configuration table is changed to **Off**.

After disabling the data collector, the metric data is not logged for the data collector until it is enabled using the **EnableDC_610** command.

Arguments

Table 44 describes the arguments that you can specify for this Take Action command.

Table 44. Arguments for the *DisableDC_610* Take Action command

Name	Value
Application_ServerName_U	The name of the application server (for example, <i>server1</i> for a WebSphere environment)
Application_ServerEnv	The application server environment. The valid values are: <ul style="list-style-type: none">• WebSphere Application Server• .NET• WebLogic Server• JBoss• DataPower• CICS• SAP• WebSphere Community Edition• WebSphere Message Broker
Hostname_U	The local host name. Either the full host name or the asterisk (*) wildcard value. This field is ignored.

Return codes

When you run the Take Action command, you receive one of the following return codes:

- 0 = The action completed successfully and the data collector was disabled.
- 1 = No change was made to the status of data collection.
- 2 = The action completed unsuccessfully. The data collector was not disabled.

DisableReqIDMntr_610 Take Action command

Description

The **DisableReqIDMntr_610** action is used to disable (turn off) the aggregation of data collection for all web service requester identities that you are monitoring.

The web service requester identity status is displayed in the Requester Identity Monitoring Status view of the subsidiary Requester Identity Monitoring Configuration workspace.

Sending the action

To run the **DisableReqIDMntr_610** action, complete the following steps:

1. From the Tivoli Enterprise Portal, navigate to the Services Management Agent workspace.
2. Right-click the Services Management Agent node and select **Workspace -> Requester Identity Monitoring Configuration**.
3. In the Requester Identity Monitoring Status view, select the row, right-click, and select **Take Action -> Select** to display the Take Action window.

4. In the **Action** area, click the **Name** field to display the list of available actions for this agent.
5. From the list of available actions, click **DisableReqIDMntr_610**.
6. The **Destination System(s)** field contains one or more target systems where you can send the command. Click the wanted system name and then click **OK**.
7. The Action Status window displays with the resulting return code. To close the window, click **OK**. The Take Action window also closes.

Arguments

You cannot specify any arguments for this Take Action command.

Return codes

When you run the Take Action command, you might receive one of the following return codes:

- 0 = The action completed successfully and requester identity monitoring was disabled.
- 2 = The target agent does not support this command.

EnableDC_610 Take Action command

Description

The **EnableDC_610** action is used to enable the data collector in a specific application server runtime environment. Use this command to define criteria to selectively enable data collection and filtering of certain web services traffic.

The data collector configuration status displays in the Data Collector Global Configuration table view in the Services Management Agent workspace. Each row of the table represents a unique data collector for a particular application server runtime environment. The **Data Collector On/Off** column indicates whether the data collector is enabled or disabled. For more information about the columns in this table, see “Data Collector Global Configuration_610 attributes” on page 263.

To enable a data collector, you must set the value in the **Data Collector On/Off** column to *On*. Right-click the row in the Data Collector Global Configuration table, and select **EnableDC_610** from the Take Action command menu.

After you enable a data collector in the Data Collector Global Configuration table view, subsequent Web traffic for that data collector is activated until you disable it again using the **DisableDC_610** Take Action command.

Sending the action

To run the **EnableDC_610** action, complete the following steps:

1. From the Tivoli Enterprise Portal, navigate to the Services Management Agent workspace.
2. In the Data Collector Global Configuration view, right-click the row in the table for the data collector to be enabled (the value in the **Data Collector On/Off** column should be set to *Off*) and select **Take Action** -> **Select** to display the Take Action window.
3. To display the list of available actions, under Action, click in the **Name** field.
4. From the list of available actions, click **EnableDC_610**.

5. The Take Action window is displayed and the selected data collector information is included in the **Command** field. Either accept the argument values for the command or, to edit the argument values, click **Arguments**.
6. The **Destination System(s)** field contains one or more target systems where you can send the command. Click the wanted system name and then click **OK**.
7. The Action Status window displays with the resulting return code. To close the window, click **OK**. The Take Action window also closes.
8. If you received a successful return code, refresh the display of the Services Management Agent workspace to verify that the value in the **Data Collector On/Off** column in the Data Collector Global Configuration table is changed to *On*.

After enabling the data collector, web services data is logged for the data collector until you disable data collection using the **DisableDC_610** command.

Arguments

Table 45 describes the arguments that you can specify for this Take Action command.

Table 45. Arguments for the EnableDC_610 Take Action command

Name	Value
Application_ServerName_U	The name of the application server (for example, <i>server1</i> for a WebSphere environment)
Application_ServerEnv	The application server environment. The valid values are: <ul style="list-style-type: none"> • WebSphere Application Server • .NET • WebLogic Server • JBoss • DataPower • CICS • SAP • WebSphere Community Edition • WebSphere Message Broker
Hostname_U	The local host name. Either the full host name or the asterisk (*) wildcard value. This field is ignored.

Return codes

When you run the Take Action command, you receive one of the following return codes:

- 0 = The action completed successfully and the data collector was enabled.
- 1 = No change was made to the status of data collection.
- 2 = The action completed unsuccessfully. The data collector was not enabled.

EnableReqIDMntr_610 Take Action command

Description

The **EnableReqIDMntr_610** action is used to enable (turn on) the aggregation of data collection for all web service requester identities that you want to monitor.

The web service requester identity status is displayed in the Requester Identity Monitoring Status view of the subsidiary Requester Identity Monitoring Configuration workspace.

DataPower firmware upgrade: If you want to monitor web service requesters that are sending requests through a DataPower appliance, the DataPower appliance must be using firmware version 3.6.1 or later. Refer to the *Installation Guide* for details on configuring the DataPower appliance for monitoring.

Sending the action

To run the **EnableReqIDMntr_610** action, complete the following steps:

1. From the Tivoli Enterprise Portal, navigate to the Services Management Agent workspace.
2. Right-click the Services Management Agent node and select **Workspace -> Requester Identity Monitoring Configuration**.
3. To display the Take Action window, in the Requester Identity Monitoring Status view, select the row, right-click, and select **Take Action -> Select**.
4. To display the list of available actions for this agent, in the **Action** area, click the **Name** field.
5. From the list of available actions, click **EnableReqIDMntr_610**.
6. The **Destination System(s)** field contains one or more target systems where you can send the command. Click the wanted system name and then click **OK**.
7. The Action Status window is displayed with the resulting return code. Click **OK** to close the window. The Take Action window also closes.

Arguments

You cannot specify any arguments for this Take Action command.

Return codes

When you run the Take Action command, you might receive one of the following return codes:

- 0 = The action completed successfully and requester identity monitoring was enabled.
- 2 = The target agent does not support this command.

SetReqIDTypeHostIP Take Action command

Description

The **SetReqIDTypeHostIP** action sets an attribute value that instructs the monitoring agent to use the available host name or IP address from the remote requester computer as the requester identity, when requester identity monitoring is enabled. A valid value for a host name might be *abc.raleigh.ibm.com* and a valid IP address might be *132.174.95.5*. The host name is used as the requester identity by default. If it is not available from the application server runtime environment, the ITCAM for SOA data collector uses the IP address as the requester identity, if it is available. The

host name or IP address value for the web service requester identity is displayed in the Requester Identity Monitoring Status table view of the Requester Identity Monitoring Configuration workspace.

Sending the action

To run the SetReqIDTypeHostIP action, complete the following steps:

1. From the Tivoli Enterprise Portal, navigate to the Services Management Agent workspace.
2. Right-click the **Services Management Agent** node and select **Workspace -> Requester Identity Monitoring Configuration**.
3. In the Requester Identity Monitoring Status view, select a row in the table, and right-click and select **Take Action -> Select** to display the Take Action window.
4. To display the list of available actions for this agent, in the **Action** area, click the **Name** field.
5. From the list of available actions, click **SetReqIDTypeHostIP**. The **Command** field is completed with the command syntax.
6. The **Destination Systems** field contains one or more target systems where you can send the command. Click the wanted system name and then click **OK**.
7. The Action Status window is displayed along with the resulting return code. To close the window, click **OK**. The Take Action window also closes.

Arguments

You cannot specify any arguments for this Take Action command.

Return codes

When you run the Take Action command, you might receive one of the following return codes:

- 0 = The action completed successfully and monitoring of requester identity records the host name or IP address from the requester computer.
- 2 = The action completed unsuccessfully.

SetReqIDTypeUserInfo Take Action command

Description

The **SetReqIDTypeUserInfo** action sets an attribute value that instructs the monitoring agent to use the available user information as the requester identity, when requester identity monitoring is enabled. The requester identity for a service request might be the user ID that you log in with when you access an application. For example, *abc@us.ibm.com* or *cn=abc@us.ibm.com,ou=Widget Division,ou=Austin,o=IBM,c=US*.

Sending the action

To run the SetReqIDTypeUserInfo action, complete the following steps:

1. From the Tivoli Enterprise Portal, navigate to the Services Management Agent workspace.
2. Right-click the Services Management Agent node and select **Workspace -> Requester Identity Monitoring Configuration**.
3. In the Requester Identity Monitoring Status view, select a row in the table, right-click and select **Take Action -> Select** to display the Take Action window.

4. To display the list of available actions for this agent, in the **Action** area, click the **Name** field.
5. From the list of available actions, click **SetReqIDTypeUserInfo**.
6. The **Command** field is completed with the command syntax.
7. The **Destination Systems** field contains one or more target systems where you can send the command. Click the wanted system name and then click **OK**.
8. The Action Status window displays with the resulting return code. To close the window, click **OK**. The Take Action window also closes.

Arguments

You cannot specify any arguments for this Take Action command.

Return codes

When you run the Take Action command, you might receive one of the following return codes:

- 0 = The action completed successfully and monitoring of requester identity records the user information as the requester identity.
- 2 = The action completed unsuccessfully.

updateLogging_610 Take Action command

Description

The **updateLogging_610** action is used to set the level of logging for the associated message interception point. Log information is categorized as either an error, a warning, or informational.

Data collector configuration status is displayed in the Data Collector Global Configuration table view in the Services Management Agent workspace. Each row of the table represents a unique data collector for a particular application server runtime environment. The **Debug Log Level** column indicates the level of data collection for this data collector. Valid values in this column are *Error*, *Warning*, or *Info*. For more information about the columns in this table, see the “Data Collector Global Configuration_610 attributes” on page 263.

To set the logging level for a data collector, right-click the row in the Data Collector Global Configuration table, and select the **updateLogging_610** action from the Take Action menu.

Sending the action

To run the **updateLogging_610** action, complete the following steps:

1. From the Tivoli Enterprise Portal, navigate to the Services Management Agent workspace.
2. In the Data Collector Global Configuration view, right-click a row in the table for the data collector to be configured and select **Take Action->Select** to display the Take Action window.
3. Under Action, click in the **Name** field to display the list of available actions.
4. From the list of available actions, click **updateLogging_610**.
5. The **Command** field in the Take Action window is completed with the command syntax. Click **Arguments** to display the Edit Argument Values window.
6. For the *Debug_Log_Level* argument, edit the value to *Error*, *Warning*, or *Info*.

7. After editing the argument, click **OK**. The Edit Argument Values window is closed, and you are returned to the Take Action window, with the entered monitor criteria now included in the **Command** field.
8. The **Destination System(s)** field contains one or more target systems where you can send the command. Click the wanted system name and then click **OK**.
9. The Action Status window displays with the resulting return code. To close the window, click **OK**. The Take Action window also closes.
10. If you received a successful return code, refresh the display of the Services Management Agent workspace to verify that the value in the Debug Log Level column in the Data Collector Global Configuration table is changed to the new value.

Arguments

Table 46 describes the arguments that you can specify for this Take Action command.

Table 46. Arguments for the updateLogging_610 Take Action command

Name	Value
Application_ServerName_U	The name of the application server (for example, <i>server1</i> for a WebSphere environment)
Application_ServerEnv	The application server environment. The valid values are: <ul style="list-style-type: none"> • WebSphere Application Server • .NET • WebLogic Server • JBoss • DataPower • CICS • SAP • WebSphere Community Edition • WebSphere Message Broker
Hostname_U	The local host name. Either the full host name or the asterisk (*) wildcard value.
Debug_Log_Level	The level of logging. The valid values are: <p>Error Only error level data is logged.</p> <p>Warning Error and warning level data is logged.</p> <p>Info Informational, warning, and error level data is logged.</p>

Return codes

When you run the Take Action command, you receive one of the following return codes:

- 0 = The action completed successfully and the data collector logging level was modified.

- 1 = No change was made to the level of logging.
- 2 = The action completed unsuccessfully. The data collector logging level was not changed.

updateTracing_610 Take Action command

Description

The **updateTracing_610** action is used to turn on or off data collector tracing. The data collector configuration status displays in the Data Collector Global Configuration table view in the Services Management Agent workspace. Each row of the table represents a unique data collector for a particular application server runtime environment. The **Data Collector Tracing On/Off** column indicates whether data collector tracing is turned on or off. For more information about the columns in this table, see the “Data Collector Global Configuration_610 attributes” on page 263.

To turn on or turn off tracing for a data collector, you must set the value in the **Data Collector Tracing On/Off** column to *On* or *Off*. Right-click a row in the Data Collector Global Configuration table, and select **updateTracing_610** from the Take Action command menu.

After you turn on tracing for a data collector, trace information associated with subsequent web services traffic through this data collector is written to a trace log until you turn it off using the **updateTracing_610** action from the Take Action menu. This trace log is in the **<ITCAM4SOA_Home>/KD4/logs** directory. For information about the value of **<ITCAM4SOA_Home>**, see “Operating system-dependent variables and paths” on page xi.

Sending the action

To run the **updateTracing_610** action, complete the following steps:

1. From the Tivoli Enterprise Portal, navigate to the Services Management Agent workspace.
2. In the Data Collector Global Configuration table, right-click the row in the table for the data collector to be configured and select **Take Action** → **Select** to display the Take Action window.
3. Under Action, click the **Name** field to display the list of available actions for this agent.
4. From the list of available actions, click **updateTracing_610**.
5. The **Command** field in the Take Action window is completed with the command syntax. To display the Edit Argument Values window, click **Arguments**.
6. For the TraceOnOff argument, edit the value to *On* or *Off*.
7. After editing the argument, click **OK**. The Edit Argument Values window is closed, and you are returned to the Take Action window, with the updated tracing criteria now included in the **Command** field.
8. The **Destination System(s)** field contains one or more target systems where you can send the command. Click the wanted system name and then click **OK**.
9. The Action Status window displays the resulting return code. To close the window, click **OK**. The Take Action window also closes.

After turning on tracing, trace information for the associated data collector is written to the trace log until it is turned off.

Arguments

Table 47 describes the arguments that you can specify for this Take Action command.

Table 47. Arguments for the *updateTracing_610* Take Action command

Name	Value
Application_ServerName_U	The name of the application server (for example, <i>server1</i> for a WebSphere environment)
Application_ServerEnv	The name of the application server (for example, <i>server1</i> for a WebSphere environment) <ul style="list-style-type: none">• WebSphere Application Server• .NET• WebLogic Server• JBoss• DataPower• CICS• SAP• WebSphere Community Edition• WebSphere Message Broker
Hostname_U	The local host name. Either the full host name or the asterisk (*) wildcard.
TraceOnOff	The indicator of whether tracing is turned on or off for this data collector. The valid values are <i>On</i> or <i>Off</i> .

Return codes

When you run the Take Action command, you receive one of the following return codes:

- 0 = The action completed successfully and tracing was either turned on or turned off.
- 1 = No change was made to the trace setting.
- 2 = The action completed unsuccessfully. The tracing for the data collector was not changed.

UpdMntrCntrl_610 Take Action command

Description

The **UpdMntrCntrl_610** action is used to change the message logging level for an existing monitor control configuration. This changes the amount of data that is collected by the ITCAM for SOA monitoring agent. This action must be used only to modify the *Message_Logging_Level* argument. Other arguments cannot be modified.

Specify the amount of data that is to be logged using one of the following choices:

- None
- Header information only
- Body information only
- Both header and body information

Each message is evaluated based on the monitoring criteria that you specify. If the message matches the monitor criteria, then monitoring data for that message is collected. If no criteria are specified for the agent, then information about messages for all service port names and operation names are logged. Changing the amount of data that is logged does not alter the order of evaluation for multiple monitoring criteria.

The monitoring criteria are displayed in the Data Collector Monitor Control Configuration table view in the Services Management Agent workspace. Each row of the table represents a unique monitor criteria definition. For more information about the columns in this table, see “Data Collector Monitor Control_610 attributes” on page 264.

To modify the other arguments in the monitor control, use the **DelMntrCntrl_610** Take Action command to delete the monitor criteria. Then, to add a definition, use **AddMntrCntrl_610**.

Sending the action

To run the **UpdMntrCntrl_610** action, complete the following steps:

1. From one of the views in the current workspace of the Tivoli Enterprise Portal, right-click a bar in a bar chart or in a row in a table view to select it. Select **Take Action->Select** to display the Take Action window.
2. Under Action, click in the **Name** field to display the list of available actions.
3. From the list of available actions, click **UpdMntrCntrl_610**. The **Command** field in the Take Action window is partially completed with the command syntax, and the Edit Argument Values window displays for you to enter the filtering criteria. The fields are pre-filled for you with data from the Services Inventory_610 attributes group. Enter the values for the monitoring criteria for each argument. Use the asterisk (*) wildcard value in the fields for service port name or operation name to match all messages for that argument. For the **Message_Logging_Level** argument, you must specify one of the valid values (wildcard values are not allowed).
4. After entering the values, click **OK**. The Edit Argument Values window is closed, and you are returned to the Take Action window, with the entered monitor criteria now included in the **Command** field.
5. The **Destination System(s)** field contains one or more target systems where you can send the command. Click the wanted system name and then click **OK**.
6. The Action Status window displays with the resulting return code. To close the window, click **OK**. The Take Action window also closes.
7. If you received a successful return code, refresh the display of the Services Management Agent workspace to see your new monitor criteria included as a new row in the Data Collector Monitor Control Configuration table view. Depending on the values that you specified for service port name and operation name, the monitor control criteria are placed in the table in the proper order.

After updating the message logging level, the amount of data that is collected at the message interception point matches the defined monitoring criteria. A message that does not match any of the criteria is ignored.

Arguments

Table 48 on page 254 describes the arguments that you can specify for this Take Action command.

Table 48. Arguments for the UpdMntrCntrl_610 Take Action command

Name	Value
Application_ServerName_U	The name of the application server (for example, <i>server1</i> for WebSphere environment)
Application_ServerEnv	The application server environment. The valid values are: <ul style="list-style-type: none"> • WebSphere Application Server • .NET • WebLogic Server • JBoss • DataPower • CICS • SAP • WebSphere Community Edition • WebSphere Message Broker
Hostname_U	The local host name. Either the full host name or the asterisk (*) wildcard value. This field is ignored.
Port_Namespace_U	The port namespace (also known as the Web Services Description Language port namespace). Either the full service port namespace or the asterisk (*) wildcard value, according to the limitations that are described in the monitor criteria combinations table.
Service_Name_U	The service port name (also known as the Web Services Description Language port name). Either the full service port name or the asterisk (*) wildcard value, according to the limitations that are described in the monitor criteria combinations table.
Operation_Namespace_U	The operation namespace associated with the specified service. Either the full operation namespace or the asterisk (*) wildcard value, according to the limitations that are described in the monitor criteria combinations table.
Operation_Name_U	The operation name associated with the specified service. Either the full operation name or the asterisk (*) wildcard value, according to the limitations that are described in the monitor criteria combinations table.

Table 48. Arguments for the UpdMntrCntrl_610 Take Action command (continued)

Name	Value
Message_Logging_Level	<p>The amount of data that is to be collected for messages that match the monitoring criteria. The valid values are:</p> <p>None No information is logged.</p> <p>Header Only message header information is logged.</p> <p>Body Only the body of the message is logged.</p> <p>Full Both the header and body of the message are logged.</p>

Return codes

When you run the Take Action command, you receive one of the following return codes:

- 0 = The action completed successfully and the monitor control was updated in the monitor control table.
- 1 = No change was made to the monitoring criteria.
- 2 = The action completed unsuccessfully. You might have specified a logging level other than the valid values. The monitoring criteria were not changed in the monitor control table.

If you receive a return code of 2, verify that you specified one of the valid logging levels (see the arguments table) and try again.

Chapter 12. Attribute groups

IBM Tivoli Monitoring gathers data from the various monitoring agents that are installed on the computers in your environment. Information about services is collected for ITCAM for SOA and stored for display in tables of *attributes*. Each attribute is a characteristic of an object. For example, *Service Port Name* is an attribute of a message that identifies the name of the service port where the message was intercepted.

Data that is stored in attribute tables is displayed in various table views in the Tivoli Enterprise Portal workspaces. Each table view corresponds to an *attribute group*. Columns in the table view correspond to the attributes in the group. Some of these attributes are used as parameters to create situations, or to specify Take Action commands.

See the descriptions of the attribute groups for the attribute names and the descriptions of the provided data. Some of the attribute groups are available for historical data collection.

The **(Unicode)** designation is displayed on attribute table column titles that contain UTF-8 character set data. The UTF-8 character set allows the strings of data for different languages to be translated and displayed in the attribute table columns.

Attribute groups by product version

If you upgraded your ITCAM for SOA installation to version 7.2, some older version agents might still be in use in your enterprise. Each version of ITCAM for SOA includes new attributes and groups. These attributes and groups are not available from the monitoring agents that remain at an older version.

Table 49. Attribute groups and supported versions of ITCAM for SOA monitoring agents

Attribute group	Supported ITCAM for SOA monitoring agents			
	version 6.1.0 and later	version 6.1.0 Fix Pack 1 and later	version 7.1.0 and later	version 7.2 and later
Agent Global Configuration_610		X	X	X
Data Collector Filter Control_610	X	X	X	X
Data Collector Global Configuration_610	X	X	X	X
Data Collector Monitor Control_610	X	X	X	X
Environment Mapping attributes			X	X
Fault Log_610 attributes	X	X	X	X
Message Arrival Threshold_610 attributes	X	X	X	X
Relationship Request Metrics attributes			X	X
Relationship Response Metrics attributes			X	X
Relationships attributes			X	X
Requester Identity Monitor Control _610 attributes		X	X	X

Table 49. Attribute groups and supported versions of ITCAM for SOA monitoring agents (continued)

Attribute group	Supported ITCAM for SOA monitoring agents			
	version 6.1.0 and later	version 6.1.0 Fix Pack 1 and later	version 7.1.0 and later	version 7.2 and later
Service Flow Metrics attributes			X	X
Service Port Operation Mapping attributes			X	X
Services Inventory_610 attributes	X	X	X	X
Services Inventory Requester Identity_610 attributes		X	X	X
Services Message Metric_610 attributes	X	X	X	X
Subnode Environment Mapping attributes			X	X
“Endpoint Inventory attributes” on page 285				X
“Business Process Events attributes” on page 290				X
“BPM Associated Errors attributes” on page 294				X

Also, some individual attributes within a group are only available with certain versions of the monitoring agent. The attribute descriptions indicate the versions.

Attribute groups used by predefined workspaces

Tivoli Enterprise Portal presents information in one or more *workspaces*. Workspaces are split into multiple *views*. These views display the monitored data in graphs, tables, or charts. Attributes from the various attribute groups are displayed in, and correspond to, columns in the views and in the charts and graphs. You can use the collected data that is displayed in these views to analyze and monitor performance. ITCAM for SOA provides several predefined workspaces to display its monitoring information in the Tivoli Enterprise Portal.

Table 50 displays the relationships between the predefined workspaces and their various attribute groups. In most cases, a workspace contains data or columns that have similar attributes in an attribute group. For more information about the predefined workspaces that are provided with ITCAM for SOA, see “Predefined workspaces” on page 24.

Table 50. Workspaces and their associated attribute groups

Workspace	Associated Attribute Groups
“Faults Summary workspace” on page 44	Fault Log _610 Services Inventory_610
“Message Arrival workspace” on page 37	Message Arrival Threshold_610
“Message Summary workspace” on page 42	Services Inventory_610
“Performance Summary workspace” on page 41	Services Inventory_610

Table 50. Workspaces and their associated attribute groups (continued)

"Requester Identities for Operation workspace" on page 50	Services Inventory Requester Identity_610
"Performance Summary for Requester Identity workspace" on page 51	Services Inventory Requester Identity_610
"Services Management Agent workspace" on page 36	Data Collector Filter Control_610 Data Collector Global Configuration_610 Data Collector Monitor Control_610
"Requester Identity Monitoring Configuration workspace" on page 49	Agent Global Configuration_610 Requester Identity Monitor Control_610
"Application Server Services Management workspace" on page 39 "Services Management Agent Environment workspace" on page 40	Services Inventory_610
"Services Management workspace" on page 55	This workspace and the accompanying views display the services and service relationships that are registered with WebSphere Service Registry and Repository (WSRR) that were discovered by the WSRR DLA and imported into the Tivoli Common Object Repository. Unlike other workspaces, this workspace does not have an explicit Tivoli Enterprise Portal query.
"Operational Flows workspace" on page 92 "Operational Flow for Operation workspace" on page 93 "Operational Flow for Application Server workspace" on page 95 "The Group Summary workspace" on page 152	These workspaces and the accompanying views display the interaction aggregated call path relationships between service port operations and a detailed interaction call path relationship between individual service port operations. Unlike other workspaces, these workspaces do not have an explicit Tivoli Enterprise Portal query.

Attribute groups and situations

Various attributes are used in the predefined situations for the product. To create your own situations to monitor the performance of your web services applications, use the ITCAM for SOA attributes. These situations monitor system resources or analyze multiple conditions to alert you to problems that occur when attribute values exceed their thresholds.

For more information about the situations that are provided with ITCAM for SOA, see Chapter 10, "Situations," on page 199.

Estimating table sizes in the Tivoli Data Warehouse database for historical data collection

For some attributes group, you can enable historical data collection in the Tivoli Data Warehouse database. The data can be used for historical reports. However, historical data collection for some attribute groups can create a significant load on the database, as many rows are created.

Table 51 describes how to calculate the number of rows that are created in the warehouse database for each applicable attribute group when historical data collection is configured and enabled.

Table 51. Estimating table sizes for historical data

Attribute group	Calculation of table size in the warehouse
Environment Mapping	One row per data collector subnode. If a WebSphere DataPower SOA Appliance is being monitored, there is one row per DataPower domain. Enable for history collection only if the IBM Web Services Navigator is used to retrieve data from the warehouse.
Relationship Request Metrics	One row for the <i>Provider-enter</i> relationship metrics for each operation. New rows are created for each 5-minute monitoring interval. If you want to see historical metrics in the Operational Flow workspaces, enable for history collection.
Relationship Response Metrics	One row for the <i>Provider-leave</i> relationship metrics for each operation, and one row for the <i>Client-response</i> relationship metrics for each operation. New rows are created for each 5-minute monitoring interval. Enable for history collection if you want to see historical metrics in the Operational Flow workspaces.
Relationships	One row for each service flow relationship for an operation. Enable for history collection only if the IBM Web Services Navigator is used to retrieve data from the warehouse.
Service Flow Metrics	Potentially very large. One row per call for each operation over the collection interval. Enable history collection only if the IBM Web Services Navigator is used to retrieve data from the warehouse.
Services Inventory_610	One row per unique combination of service port, operation, and service type (<i>provider</i> or <i>requester</i>) for each 5-minute monitoring interval.
Services Inventory Requester Identity_610	One row per unique combination of monitored requester ID, service port, and operation for each 5-minute monitoring interval.
Services Message Metric_610	Potentially very large. One row per call for each operation over the collection interval. This table is not used by ITCAM for SOA version 7.1 or later components. Do not enable it for history collection unless you have a custom use for it.
Service Port Operation Mapping	One row per unique combination of service port and operation. Enable for history collection only if the IBM Web Services Navigator is used to retrieve data from the warehouse.

Agent Global Configuration_610 attributes

This attribute group provides the property settings that are used by the agent to enable or disable the collection and aggregation of metric data tracked at the user level.

You can change the values of these attributes using the “EnableReqIDMntr_610 Take Action command” on page 247 and “DisableReqIDMntr_610 Take Action command” on page 244.

The following attributes are used in the “Requester Identity Monitoring Configuration workspace” on page 49.

Origin Node The node name that identifies the Tivoli Enterprise Monitoring Agent instance from which the data in this table originated. The format of this attribute is an alphanumeric text string with a maximum length of 32 characters.

Monitor Requester Identities On/Off A flag indicating whether the agent collects the metric data for the individual service requesters (On = 1, Off = 0). The format of this attribute is an integer. To identify service requesters, use the “AddRequesterIdentity_610 Take Action command” on page 233. To turn on data collection, use the “EnableReqIDMntr_610 Take Action command” on page 247. To turn off data collection, use the “DisableReqIDMntr_610 Take Action command” on page 244 Take Action command.

Table version (Unicode) The version of this table definition. This value does not represent the version of this product. The format of this attribute is an alphanumeric text string with a maximum length of 8 characters. This attribute is for internal use only and is not intended for inclusion in a query or to be displayed in a table view.

Requester Identity Type The type of requester identity to use when aggregating metrics by requester for the Services Inventory Requester Identity_610 table (User_ID = 1, Remote_Hostname = 2). The format of this attribute is an integer. The value of this attribute is displayed in the Requester Identity Monitoring Status view of the “Requester Identity Monitoring Configuration workspace” on page 49. This attribute is available only for monitoring agents that are provided with ITCAM for SOA version 7.1.0 or later.

Data Collector Filter Control_610 attributes

This attribute group defines the type of filtering that is applied to message data collected by the monitoring agent. You can change the values of these attributes using the “AddFltrCntrl_610 Take Action command” on page 226 and “DelFltrCntrl_610 Take Action command” on page 238.

The following attributes are used in the Data Collector Filter Control Configuration view in the “Services Management Agent workspace” on page 36.

Application Server Environment The type of environment in which the data collector is running.

The valid values for this attribute are as follows:

- 1 = WebSphere Application Server
- 2 = .NET
- 3 = WebLogic_Server
- 4 = JBoss
- 5 = CICS
- 6 = SAP
- 7 = WebSphere_Community_Edition
- 8 = DataPower
- 9 = Service_Component_Architecture
- 10 = WebSphere_Message_Broker

- 11 = WebSphere_MQ
- 12 = Oracle_Application_Server
- 13 = Generic_Application_Server

Application Server Name (Unicode) The name of the application server from which data is collected. For example, *server1*, if the Application Server Environment attribute is defined as *WebSphere_Application_Server*. The format of this attribute is an alphanumeric text string with a maximum of 64 characters.

Filtering Control Index An integer number that is automatically assigned to each filter control definition, indicating the order in which the criteria is evaluated, from least specific to most specific per application server.

Hostname (Unicode) The fully qualified name of the host computer on which the data collector is running (for example, *hostname.example.com*). The format of this attribute is an alphanumeric text string with a maximum of 64 characters. You can also specify an asterisk (*) as a wildcard.

Operation Name (Unicode) The name of the Web Services Description Language (WSDL) operation for which the filter criteria is applied. The format of this attribute is an alphanumeric text string with a maximum of 64 characters. You can also specify an asterisk (*) as a wildcard to include all operations for the associated service port name.

Operation Namespace (Unicode) The namespace of the Web Services Description Language (WSDL) operation for which the filter criteria is applied. The namespace, combined with the operation name, comprises a fully qualified name for the WSDL operation. The format of this attribute is an alphanumeric text string with a maximum of 128 characters. You can also specify an asterisk (*) as a wildcard to indicate any operation namespace.

Origin Node The node name of the managed system. The format of this attribute is an alphanumeric text string with a maximum of 32 characters.

Remote IP Address (Unicode) The remote IP address for which the filter criteria is applied. The format of this attribute is an alphanumeric text string with a maximum of 64 characters. You can also specify an asterisk (*) as a wildcard to indicate any IP address.

Service Port Name (Unicode) The name of the service port where the message was intercepted. This name is also known as the Web Services Description Language (WSDL) port name. The format of this attribute is an alphanumeric text string with a maximum of 64 characters. You can also specify an asterisk (*) as a wildcard to indicate any service port name.

Service Port Namespace (Unicode) The namespace of the Web Services Description Language (WSDL) service port for which the filter criteria is applied. The namespace, combined with the service port name, is a fully qualified name for the service port. The format of this attribute is an alphanumeric text string with a maximum of 128 characters. You can also specify an asterisk (*) as a wildcard to indicate any service port namespace.

Table version (Unicode) The version of this table definition. This value does not represent the version of this product. The format of this attribute is an alphanumeric text string with a maximum of 8 characters.

Data Collector Global Configuration_610 attributes

This attribute group defines the global configuration settings in the properties file that apply to data collectors running in the specified application server runtime environments.

These attributes complete the following functions:

- Enable data collection (using the “EnableDC_610 Take Action command” on page 245).
- Disable data collection (using the “DisableDC_610 Take Action command” on page 243).
- Define the level of logging information that is stored (using the “updateLogging_610 Take Action command” on page 249).
- Turn tracing functions on and off (using the “updateTracing_610 Take Action command” on page 251).

The following attributes are used in the Data Collector Global Configuration view in the “Services Management Agent workspace” on page 36.

Application Server Environment The type of environment in which the data collector is running.

The valid values for this attribute are as follows:

- 1 = WebSphere Application Server
- 2 = .NET
- 3 = WebLogic_Server
- 4 = JBoss
- 5 = CICS
- 6 = SAP
- 7 = WebSphere_Community_Edition
- 8 = DataPower
- 9 = Service_Component_Architecture
- 10 = WebSphere_Message_Broker
- 11 = WebSphere_MQ
- 12 = Oracle_Application_Server
- 13 = Generic_Application_Server

Application Server Name (Unicode) The name of the application server from which data is collected. For example, *server1*, if the Application Server Environment is defined as *WebSphere_Application_Server*. The format of this attribute is an alphanumeric text string with a maximum of 64 characters.

Data Collector On/Off A flag indicating that the data collector is enabled (On) or disabled (Off) at the associated application server. The value of this attribute is stored as an integer (0=Off, 1=On).

Data Collector Tracing On/Off A flag indicating that tracing is enabled (On) or disabled (Off) for the associated message interception point. The value of this attribute is stored as an integer (0=Off, 1=On).

Debug Log Level An indication of the level of logging that is written for the associated message interception point. The value of this attribute is stored as an integer (1=Error, 2=Warning, 3=Info).

The following values for this attribute are displayed in the Data Collector Global Configuration view of the Services Management Agent workspace:

Error Only error level data is logged.

Warning

Error and warning level data is logged.

Info Informational, warning, and error level data is logged.

Hostname (Unicode) The fully qualified name of the host computer on which the data collector is running (for example, hostname.example.com). The format of this attribute is an alphanumeric text string with a maximum of 64 characters. You can also specify an asterisk (*) as a wildcard.

Origin Node The node name of the managed system. The format of this attribute is an alphanumeric text string with a maximum of 32 characters.

Table version (Unicode) The version of this table definition. This value does not represent the version of this product. The format of this attribute is an alphanumeric text string with a maximum of eight characters.

Data Collector Monitor Control_610 attributes

This attribute group defines the type of data that is collected by the monitoring agent.

You can change these attributes using the following Take Action commands:

- “AddMntrCntrl_610 Take Action command” on page 229
- “DelMntrCntrl_610 Take Action command” on page 241
- “UpdMntrCntrl_610 Take Action command” on page 252

The following attributes are used in the Data Collector Monitor Control Configuration view in the “Services Management Agent workspace” on page 36.

Application Server Environment The type of environment in which the data collector is running.

The valid values for this attribute are as follows:

- 1 = WebSphere Application Server
- 2 = .NET
- 3 = WebLogic_Server
- 4 = JBoss
- 5 = CICS
- 6 = SAP
- 7 = WebSphere_Community_Edition
- 8 = DataPower
- 9 = Service_Component_Architecture
- 10 = WebSphere_Message_Broker
- 11 = WebSphere_MQ

- 12 = Oracle_Application_Server
- 13 = Generic_Application_Server

Application Server Name (Unicode) The name of the application server from which data is collected. The format of this attribute is an alphanumeric text string with a maximum of 64 characters.

Hostname (Unicode) The fully qualified name of the host computer on which the data collector is running (for example, hostname.example.com). The format of this attribute is an alphanumeric text string with a maximum of 64 characters. You can also specify an asterisk (*) as a wildcard.

Message Logging Level An indication of the level of detail about the message that is logged. The value of this attribute is stored as an integer (1=None, 2=Header, 3=Body, 4=Full).

The following values for this attribute are displayed in the Data Collector Monitor Control Configuration view of the Services Management Agent workspace:

None No message content information is logged.

Header

Message header information is logged.

Body The body of the message is logged.

Full Both header and body of the message are logged.

Monitoring Control Index An integer number that is automatically assigned to each monitoring control definition, and indicates the order in which the criteria is evaluated, from most specific to least specific, per application server.

Operation Name (Unicode) The name of the Web Services Description Language (WSDL) operation that is contained in the intercepted message. The format of this attribute is an alphanumeric text string with a maximum of 64 characters. Use the asterisk (*) wildcard character to indicate any operation name.

Operation Namespace (Unicode) The namespace of the Web Services Description Language (WSDL) operation for which you are defining the monitoring control. The namespace, combined with the operation name, comprises a fully qualified name for the WSDL operation. The format of this attribute is an alphanumeric text string with a maximum of 128 characters. Use the asterisk (*) wildcard character to indicate any operation namespace.

Origin Node The node name of the managed system. The format of this attribute is an alphanumeric text string no longer than 32 characters.

Service Port Name (Unicode) The name of the service port where the message was intercepted. This name is also known as the Web Services Description Language (WSDL) port name. The format of this attribute is an alphanumeric text string with a maximum of 64 characters. Use the asterisk (*) wildcard character to indicate any service port name.

Service Port Namespace (Unicode) The namespace of the Web Services Description Language (WSDL) service port for which you are defining the monitoring control. The namespace, combined with the service port name, is a fully qualified name for

the service port. The format of this attribute is an alphanumeric text string with a maximum of 128 characters. Use the asterisk (*) wildcard character to indicate any service port namespace.

Table version (Unicode) The version of this table definition. This value does not represent the version of this product. The format of this attribute is an alphanumeric text string with a maximum of 8 characters.

Fault Log_610 attributes

This attribute group defines the data that make up each fault (e.g. SOAP fault) that is received by the data collector. Use these attributes to create situations that monitor errors received by the data collector.

The following attributes are used in the Fault Details view in the “Faults Summary workspace” on page 44.

Application Server Environment The type of environment in which the data collector is running.

The valid values for this attribute are:

- 1 = WebSphere Application Server
- 2 = .NET
- 3 = WebLogic_Server
- 4 = JBoss
- 5 = CICS
- 6 = SAP
- 7 = WebSphere_Community_Edition
- 8 = DataPower
- 9 = Service_Component_Architecture
- 10 = WebSphere_Message_Broker
- 11 = WebSphere_MQ
- 12 = Oracle_Application_Server
- 13 = Generic_Application_Server

Application Server Name (Unicode) The name of the application server from which data is collected. The format of this attribute is an alphanumeric text string with a maximum of 64 characters. ¹

Application Server Node Name (Unicode) The node name of the application server where the message was intercepted. The format of this attribute is an alphanumeric text string with a maximum of 64 characters. This attribute is valid only for ITCAM for SOA version 7.2 Fix Pack 1 or later. Depending on the application server environment, this attribute might be empty. ²

Application Server Cell Name (Unicode) The name of the application server cell where the message was intercepted. The format of this attribute is an alphanumeric

1. For a DataPower SOA appliance, this attribute displays the value of the DataPower.displaygroup property if the property is set in the DataPower configuration file.

2. For a DataPower SOA appliance, this attribute displays the DataPower host fully qualified domain name (FQDN) unless the DataPower.multihost.group property is set to true in the DataPower configuration file. The DataPower.multihost.group property is introduced in ITCAM for SOA version 7.2 Fix Pack 1.

text string with a maximum of 64 characters. This attribute is valid only for ITCAM for SOA version 7.2 Fix Pack 1 or later. Depending on the application server environment, this attribute might be empty.³

Application Server Cluster Name (Unicode) The name of the application server cluster where the message was intercepted. The format of this attribute is an alphanumeric text string with a maximum of 64 characters. This attribute is valid only for ITCAM for SOA version 7.2 Fix Pack 1 or later. Depending on the application server environment, this attribute might be empty.⁴

Current Correlator (Unicode) The service flow identifier assigned at an interception point to an outgoing message, also known as the child correlator. The format of this attribute is an alphanumeric text string with a maximum of 128 characters.

Fault Code (Unicode) The SOAP fault code for which the message is to be rejected. The format of this attribute is an alphanumeric text string with a maximum of 128 characters.

Fault String (Unicode) The SOAP fault string. The format of this attribute is an alphanumeric text string with a maximum of 128 characters.

Error Code (Unicode) The error code of the DataPower transaction. The format of this attribute is an alphanumeric text string with a maximum of 128 characters. This attribute is valid only for ITCAM for SOA version 7.2 Fix Pack 1 or later.

Error SubCode (Unicode) The error subcode of the DataPower transaction. The error subcode either has the same value as the error code or it provides further details on the root cause of the error. The format of this attribute is an alphanumeric text string with a maximum of 128 characters. This attribute is valid only for ITCAM for SOA version 7.2 Fix Pack 1 or later.

Domain The DataPower domain name. The format of this attribute is an alphanumeric text string with a maximum of 128 characters. This attribute is valid only for ITCAM for SOA version 7.2 Fix Pack 1 or later.

Transaction Identity The identity of the DataPower transaction for a service. The format of this attribute is an alphanumeric text string with a maximum of 128 characters. This attribute is valid only for ITCAM for SOA version 7.2 Fix Pack 1 or later.

Requester Identity The identity of the DataPower service requester. The format of this attribute is an alphanumeric text string with a maximum of 128 characters. This attribute is valid only for ITCAM for SOA version 7.2 Fix Pack 1 or later.

Hostname (Unicode) The fully qualified name of the host computer on which the data collector is running. The format of this attribute is an alphanumeric text string with a maximum of 64 characters.

3. For a DataPower SOA appliance, this attribute displays the DataPower host alias if the `DataPower.alias` property is set in the DataPower configuration file. The `DataPower.alias` property is introduced in ITCAM for SOA version 7.2 Fix Pack 1.

4. For a DataPower SOA appliance, this attribute displays the DataPower domain name if the `DataPower.dimension.domain` property is set to true in the DataPower configuration file. The `DataPower.dimension.domain` property is introduced in ITCAM for SOA version 7.2 Fix Pack 1.

Message Interception Location The location where the message was intercepted (0=Client_Request, 1=Client_Response, 2=Server_Enter, 3=Server_Leave). The value of this attribute is stored as an integer.

Message Interception Time The date and time when the message was intercepted, expressed in Coordinated Universal Time (UTC) format.

Operation Name (Unicode) The name of the Web Services Description Language (WSDL) operation. The format of this attribute is an alphanumeric text string with a maximum of 64 characters.

Operation Namespace (Unicode) The namespace of the Web Services Description Language (WSDL) operation that is monitored by the data collector. The namespace, combined with the operation name, comprises a fully qualified name for the WSDL operation. The format of this attribute is an alphanumeric text string with a maximum length of 128 characters.

Origin Node The subnode name of the managed system. The format of this attribute is an alphanumeric text string no longer than 32 characters.

Service Port Name (Unicode) The name of the service port where the message was intercepted, also known as the Web Services Description Language (WSDL) port name. The format of this attribute is an alphanumeric text string with a maximum of 64 characters.

Service Port Name Type The type of service port, such as the Web Services Description Language (WSDL) port name, or the SCA Component name. The format of this attribute is an integer. This attribute is available only for monitoring agents from ITCAM for SOA version 7.1.0 or later.

The following values are valid for this attribute:

- WSDL_Port_Name (This value is stored as 1.)
- WSDL_Service_Name (This value is stored as 2.)
- WSDL_Service_URI (This value is stored as 3.)
- URI (This value is stored as 4.)
- SCA_Component_Name (This value is stored as 5.)
- Generic_Name (This value is stored as 6.)
- CICS_Web_Service_Name (This value is stored as 7.)
- Message_Flow_For_SOAP (This value is stored as 8.)
- Message_Flow_For_XML (This value is stored as 9.)
- Message_Flow_For_Other (This value is stored as 10.)
- Service_Integration_Bus (This value is stored as 11.)
- BPD (This value is stored as 12.)

Service Port Namespace (Unicode) The namespace of the Web Services Description Language (WSDL) service port that is monitored by the data collector. The namespace, combined with the service port name, comprises a fully qualified name for the service port. The format of this attribute is an alphanumeric text string with a maximum of 128 characters.

Unique Key (Unicode) This attribute is for internal use only and is used to uniquely identify the combination of service port namespace, service port name, operation namespace, operation name, and fault date in the row of data in the

table. The key that is generated from this combination is guaranteed to be unique within the table for the specific data collector, but might not be unique across multiple data collectors. This attribute is not displayed in a table view and is not intended for inclusion in a query based on this attribute group.

Table version (Unicode) The version of this table definition. This value does not represent the version of this product. The format of this attribute is an alphanumeric text string with a maximum of 8 characters.

Message Arrival Threshold_610 attributes

This attribute group represents the computed data table, used to create situations based on the arrival rate of messages that monitor message arrival data, such as the number of messages that arrive during a specified time interval.

The following attributes are used in the “Message Arrival workspace” on page 37.

Current Message Count The number of actual messages intercepted during the specified time interval. The format of this attribute value is an integer. When you use this attribute in a situation, delta or percent functions are not supported. The monitoring agent supports only comparisons against the actual value. If you create and distribute such a situation using delta or percent functions, the situation is not displayed in the Message Arrival Details view and is ignored.

Hostname (Unicode) The fully qualified name of the computer on which the data collector is running. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters. You can also specify an asterisk (*) as a wildcard.

Message Count Threshold The threshold for the number of messages that are allowed within the specified time interval. The format of this attribute is an integer.

Operation Name (Unicode) The name of the Web Services Description Language (WSDL) operation that is contained in the intercepted message. The format of this attribute is an alphanumeric text string with a maximum of 64 characters. Use the asterisk (*) wildcard character to indicate any operation name.

Operation Namespace (Unicode) The namespace of the Web Services Description Language (WSDL) operation monitored by the data collector. The namespace, combined with the operation name, comprises a fully qualified name for the WSDL operation. The format of this attribute is an alphanumeric text string with a maximum of 128 characters. Use the asterisk (*) wildcard character to indicate any operation namespace.

Origin Node The node name of the managed system. The format of this attribute is an alphanumeric text string with a maximum length of 32 characters.

Remote IP Address (Unicode) The remote IP address of the computer for which the filter criteria is applied. The format of this attribute is an alphanumeric text string with a maximum of 64 characters. Use the asterisk (*) wildcard character to indicate any remote IP address.

Service Port Name (Unicode) The name of the service port where the message was intercepted. This name is also known as the Web Services Description Language (WSDL) port name. The format of this attribute is an alphanumeric text string with

a maximum of 64 characters. Use the asterisk (*) wildcard character to indicate any service port name. If you specify the asterisk (*) wildcard character for the service port name then you must specify the asterisk (*) wildcard character for the operation name.

Service Port Namespace (Unicode) The namespace of the Web Services Description Language (WSDL) service port monitored by the data collector. The namespace, combined with the service port name, comprises a fully qualified name for the service port. The format of this attribute is an alphanumeric text string with a maximum of 128 characters. Use the asterisk (*) wildcard character to indicate any service port namespace.

Table version (Unicode) The version of this table definition. This value does not represent the version of this product. The format of this attribute is an alphanumeric text string with a maximum of eight characters.

Situation Name (Unicode) The name of the situation that corresponds to this threshold. The format of this attribute is an alphanumeric text string with a maximum of 32 characters.

Threshold Comparison Operator The logical operator that is applied to the Message Count Threshold. The format of this attribute is a 2-byte integer (0=Greater_Than, 1=Greater_Than_Or_Equal_To, 2=Less_Than, 3=Less_Than_Or_Equal_To).

Time Interval The sliding time interval for which the condition is applied, specified in seconds. The minimum allowed interval is 60 seconds. The value specified is rounded up to the next closest 30-second increment.

Requester Identity Monitor Control_610 attributes

This attribute group provides data about the collection and aggregation criteria that are used to indicate what specific requester IDs have their metric data tracked at the user level.

The attributes in this group can be changed using the “AddRequesterIdentity_610 Take Action command” on page 233 and “DeleteRequesterIdentity_610 Take Action command” on page 235.

The following attributes are used in the “Requester Identity Monitoring Configuration workspace” on page 49.

Origin Node The node name of the managed system. The format of this attribute is an alphanumeric text string with a maximum length of 32 characters.

Table version (Unicode) The version of this table definition. This value does not represent the version of this product. The format of this attribute is an alphanumeric text string with a maximum length of eight characters. This attribute is for internal use only and is not intended for inclusion in a query or to be displayed in a table view.

Requester Identity (Unicode) The identity of the service requester. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters. The requester identity for a service request might be the user ID that you log in with when you access an application. For example, *abc@us.ibm.com* or *cn=abc@us.ibm.com,ou=Widget Division,ou=Austin,o=IBM,c=US*.

The capitalization and spacing of these identities must match the way this string is returned by your monitored runtime environment. You might want to monitor all requester identities using the asterisk (*) wildcard character for a time or in a test environment, to confirm the proper spacing and capitalization for your identities.

Services Inventory_610 attributes

This attribute group provides data about current service inventory. It also contains aggregate metric data. Use the Services Inventory_610 attributes to track your services inventory and ensure that settings do not reach or exceed predefined thresholds.

The Services Inventory_610 attribute table contains one row per unique combination of service port name, service port namespace, operation name, operation namespace, and service type (provider or requester), for each 5-minute time interval. Only the incomplete and most recently completed time intervals are available for real-time queries. When historical data collection is enabled, each 5-minute interval record is collected as historical data.

The attributes are used in the following workspaces:

- “Message Summary workspace” on page 42
- “Performance Summary workspace” on page 41
- “Application Server Services Management workspace” on page 39
- “Services Management Agent Environment workspace” on page 40

The following attributes include both successful messages and fault messages by default. Beginning with ITCAM for SOA version 7.2 Fix Pack 1, to exclude fault messages, set the `kd4.ira.excludedefault.controls.enabled` property in the `KD4.dc.properties` file to 1:

- Average Message Length
- Message Count
- Elapsed Time Message Count
- Average Elapsed Message Round Trip Time
- Max Message Length
- Max Elapsed Time
- Min Message Length
- Min Elapsed Time
- Message Length Std Dev
- Elapsed Message Round Trip Std Dev

For more information, see “Excluding faults from the calculation of service metric values” in the *IBM Tivoli Composite Application Manager for SOA User’s Guide*.

Origin Node The application environment instance of the managed system. The format of this attribute is an alphanumeric text string with a maximum length of 32 characters.

Service Type The type of service (0=Requester, 1=Provider).

Service Port Name (Unicode) The name of the service port where the message was intercepted. Also known as the Web Services Description Language (WSDL) port name. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters. An asterisk (*) value includes all service port names.

Operation Name (Unicode) The name of the WSDL operation contained in the intercepted message. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters. An asterisk (*) value includes all operations for the associated service port name.

Local Hostname (Unicode) The fully qualified name of the host computer on which the data collector is running, and where the message was intercepted. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters.

Local IP Address (Unicode) The IP address of the computer on which the data collector is running, and where the message was intercepted. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters.

Application Server Environment The type of runtime environment in which the data collector is running. The format of this attribute is an integer.

The valid values for this attribute are:

- 1 = WebSphere Application Server
- 2 = .NET
- 3 = WebLogic_Server
- 4 = JBoss
- 5 = CICS
- 6 = SAP
- 7 = WebSphere_Community_Edition
- 8 = DataPower
- 9 = Service_Component_Architecture
- 10 = WebSphere_Message_Broker
- 11 = WebSphere_MQ
- 12 = Oracle_Application_Server
- 13 = Generic_Application_Server

Application Server Node Name (Unicode) The node name of the application server where the message was intercepted. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters. Depending on the application server environment, this attribute might be empty.⁵

Application Server Cell Name (Unicode) The name of the application server cell where the message was intercepted. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters. Depending on the application server environment, this attribute might be empty.⁶

Application Server Name (Unicode) The name of the application server where the message was intercepted. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters.⁷

5. For a DataPower SOA appliance, this attribute displays the DataPower host fully qualified domain name (FQDN) unless the `DataPower.multihost.group` property is set to true in the DataPower configuration file. The `DataPower.multihost.group` property is introduced in ITCAM for SOA version 7.2 Fix Pack 1.

6. For a DataPower SOA appliance, this attribute displays the DataPower host alias if the `DataPower.alias` property is set in the DataPower configuration file. The `DataPower.alias` property is introduced in ITCAM for SOA version 7.2 Fix Pack 1.

7. For a DataPower SOA appliance, this attribute displays the value of the `DataPower.displaygroup` property if the property is set in the DataPower configuration file.

Application Server Cluster Name (Unicode) The name of the application server cluster where the message was intercepted. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters. Depending on the application server environment, this attribute might be empty.⁸

Table version (Unicode) The version of this table definition. This value does not represent the version of this product. The format of this attribute is an alphanumeric text string with a maximum length of eight characters.

Interval Begin Time The inclusive beginning date and time of the monitoring interval when the message was intercepted.

When this attribute is displayed in a view in the Tivoli Enterprise Portal, it is expressed as a date and time in Greenwich Mean Time (GMT). This attribute is saved in the Tivoli Data Warehouse as a number in the format *CYYMMDDhhmmssnnn*. For example, a value for October 18 2008 at 19:45:17:010 GMT is stored as *1081018194517010*, where:

C = the century marker (for example, *1*)
YY = the year (for example, *08* for 2008)
MM = the month (for example, *10* for October)
DD = the day of the month (for example, *18* for the 18th day)
hh = the hour portion of the time (for example, *19* for 7:00 p.m.)
mm = the minutes portion of the time (for example, *45* for 45 minutes past the hour)
ss = the seconds portion of the time (for example, *17*)
nnn = the milliseconds portion of the time (for example, *010* for 10 milliseconds)

Interval End Time The exclusive end date and time of the monitoring interval when the message was intercepted.

When this attribute is displayed in a view in the Tivoli Enterprise Portal, it is expressed as a date and time in Greenwich Mean Time (GMT). This attribute is saved in the Tivoli Data Warehouse as a number in the format *CYYMMDDhhmmssnnn*. For example, a value for October 18 2008 at 19:45:17:010 GMT is stored as *1081018194517010*, where:

C = the century marker (for example, *1*)
YY = the year (for example, *08* for 2008)
MM = the month (for example, *10* for October)
DD = the day of the month (for example, *18* for the 18th day)
hh = the hour portion of the time (for example, *19* for 7:00 p.m.)
mm = the minutes portion of the time (for example, *45* for 45 minutes past the hour)
ss = the seconds portion of the time (for example, *17*)
nnn = the milliseconds portion of the time (for example, *010* for 10 milliseconds)

Interval Length The length of the monitoring interval, in minutes. The format of this attribute is an integer.

Average Message Length The average message length, in bytes, observed during this interval (including headers when possible). The value of this attribute is set to *-1* if no messages were observed for the operation during the monitoring interval. The format of this attribute is an integer.

8. For a DataPower SOA appliance, this attribute displays the DataPower domain name if the `DataPower.dimension.domain` property is set to true in the DataPower configuration file. The `DataPower.dimension.domain` property is introduced in ITCAM for SOA version 7.2 Fix Pack 1.

Message Count The number of messages transmitted during this interval. The format of this attribute is an integer.

Elapsed Time Message Count The number of messages transmitted during this interval that contain an elapsed time value. The format of this attribute is an integer. For a Business Process Definition (BPD), the value is always -1.

Average Elapsed Message Round Trip Time The average elapsed round-trip time, in milliseconds. This average does not include values for round-trip time that were set to -1 (which indicates entry requests, for which there is no elapsed response time, or which indicates that no responses were intercepted during the monitored interval). The format of this attribute is an integer. For a Business Process Definition (BPD), the value is always -1.

Service Port Name : Operation Name : Service Type (Unicode) The concatenation of the service port namespace, service port name, operation namespace, operation name, and service type. The format of this attribute is an alphanumeric text string with a maximum length of 396 characters.

Fault Count The number of faults observed during this interval. The format of this attribute is an integer.

Max Message Length The length of the largest message, in bytes, observed during this monitoring interval. When no messages are observed during this interval, the value of this attribute is set to -1. The format of this attribute is an integer.

Max Elapsed Time The longest elapsed time, in milliseconds, of any message observed during this monitoring interval. When no messages are observed during this interval, the value of this attribute is set to -1. The format of this attribute is an integer. For a Business Process Definition (BPD), the value is always -1.

Min Message Length The length of the shortest message, in bytes, observed during this monitoring interval. When no messages are observed during the monitoring interval, the value of this attribute is set to -1. The format of this attribute is an integer.

Min Elapsed Time The shortest elapsed time, in milliseconds, of any message observed during the monitoring interval. When no messages are observed during this interval, the value of this attribute is set to -1. The format of this attribute is an integer. For a Business Process Definition (BPD), the value is always -1.

Message Length Std Dev The standard deviation, in bytes, of all message lengths observed during the monitoring interval. The format of this attribute is an integer.

Elapsed Message Round Trip Time Std Dev The standard deviation of all elapsed round-trip times, in milliseconds, observed during this monitoring interval. This calculation does not include values for response time that were set to -1 (which indicates entry requests, or indicates that no responses were intercepted during the monitoring interval). The format of this attribute is an integer. For a Business Process Definition (BPD), the value is always -1.

Interval Status The status of this monitoring interval (Incomplete=1, Complete=2). A value of 1 (Incomplete) means that the end of the current monitoring interval is not yet reached, and it is possible that not all messages were received for the entire interval. The format of this attribute is an integer.

Filler1 This attribute is for internal use only and is not intended for inclusion in a query or to be displayed in a table view.

Service Port Namespace (Unicode) The namespace that is combined with the service port name to fully qualify the WSDL service port that is monitored by the data collector. The format of this attribute is an alphanumeric text string with a maximum of 128 characters.

Operation Namespace (Unicode) The namespace that is combined with the operation name to fully qualify the WSDL operation that is monitored by the data collector. The format of this attribute is an alphanumeric text string with a maximum length of 128 characters.

Port Number The number of the port, 0 - 65535, on which the application server runtime environment that is being monitored is listening. If the port number is unknown, this attribute is set to -1. The format of this attribute is an integer. If you configure your WebSphere Application Server environment with security to prevent unauthenticated users from connecting to your Java Management Extensions (JMX) server, the data collector cannot obtain the port number. As a result, this attribute is set to a value of 0.

Unique Key (Unicode) This attribute is for internal use only and is used to uniquely identify the combination of service port namespace, service port name, operation namespace, operation name, and service type in the row of data in the table. The key that is generated from this combination is guaranteed to be unique within the table for the specific data collector, but might not be unique across multiple data collectors. This attribute is not displayed in a table view and is not intended for inclusion in a query based on this attribute group.

This attribute is not useful for queries, but might be useful for situation definitions. Use it as the Display Item in the situation definition if you want to differentiate the situation and have IBM Tivoli Monitoring generate individual situation events for each row in the table that meets the situation threshold.

Tivoli Enterprise Monitoring Agent Info (Unicode) Provides information about the Tivoli Enterprise Monitoring Agent (TEMA) that might be used to link to other related workspaces. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters. This attribute is available only for ITCAM for SOA monitoring agents at version 7.1.0 or later.

DC Info (Unicode) Provides information about the data collector that might be used to link to other related workspaces. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters. This attribute is available only for ITCAM for SOA monitoring agents at version 7.1.0 or later.

Service Port Name Type The type of the service port. The port is identified in the Service Port Name attribute. The format of this attribute is an integer.

The possible enumerated values are:

- 1 = WSDL_Port_Name
- 2 = WSDL_Service_Name
- 3 = WSDL_Service_URI
- 4 = URI
- 5 = SCA_Component_Name

- 6 = Generic_Name
- 7 = CICS_Web_Service_Name
- 8 = Message_Flow_For_SOAP
- 9 = Message_Flow_For_XML
- 10 = Message_Flow_For_Other
- 11 = Service_Integration_Bus
- 12= BPD

Request Count The number of request messages transmitted during this interval. The format of this attribute is an integer. This attribute is valid only for ITCAM for SOA version 7.1.1 or later.

Interval Response Count The number of response messages transmitted during this interval whose corresponding request message was observed during this same interval. This count includes both valid responses and fault responses and is less than or equal to the Elapsed Time Message Count attribute. The format of this attribute is an integer. This attribute is valid only for ITCAM for SOA version 7.1.1 or later.

Interval Fault Count The number of fault response messages transmitted during this interval whose corresponding request message was observed during this same interval. This count is less than or equal to the Fault Count attribute. The format of this attribute is an integer. This attribute is valid only for ITCAM for SOA version 7.1.1 or later.

Missing Response Percentage The percentage of request messages transmitted during this interval that did not have a corresponding response. A response whose request was observed in a previous interval is not counted. This attribute is different from the Missing Valid Response Percentage attribute, because a fault is counted as a response. The format of this attribute is an integer. This attribute is valid only for ITCAM for SOA version 7.1.1 or later.

Missing Valid Response Percentage The percentage of request messages transmitted during this interval that did not have a corresponding valid response. A response whose request was observed in a previous interval is not counted. This attribute is different from the Missing Response Percentage attribute, because a fault is not counted as a valid response. The format of this attribute is an integer. This attribute is valid only for ITCAM for SOA version 7.1.1 or later.

Valid Response Percentage The percentage of response messages transmitted during this interval that were not faults. The format of this attribute is an integer. This attribute is valid only for ITCAM for SOA version 7.1.1 or later.

Services Inventory Requester Identity_610 attributes

This attribute group augments the Services Inventory_610 attributes by providing further detail for viewing service metrics and their relationships.

Use the Services Inventory Requester Identity_610 attributes to track the requester identities that are requesting your services, and to ensure that the use of services for any requester identity does not exceed predefined thresholds.

The Services Inventory ReqID_610 table contains one row per unique combination of requester identity, service port name, service port namespace, operation name,

operation namespaces, and message type (provider or requester), for each monitoring interval. Only the incomplete and most recently completed monitoring intervals are available for real-time queries. When historical data collection is enabled, each monitoring interval record is collected as historical data.

The attributes are used in the following workspaces:

- “Requester Identities for Operation workspace” on page 50
- “Performance Summary for Requester Identity workspace” on page 51

The following attributes include both successful messages and fault messages by default. Beginning with ITCAM for SOA version 7.2 Fix Pack 1, to exclude fault messages, set the `kd4.ira.exclude.default.controls.enabled` property in the `KD4.dc.properties` file to 1:

- Average Message Length
- Message Count
- Elapsed Time Message Count
- Average Elapsed Message Round Trip Time
- Max Message Length
- Max Elapsed Time
- Min Message Length
- Min Elapsed Time
- Message Length Std Dev
- Elapsed Message Round Trip Std Dev

For more information, see “Excluding faults from the calculation of service metric values” in the *IBM Tivoli Composite Application Manager for SOA User's Guide*.

Origin Node The application environment instance of the managed system. The format of this attribute is an alphanumeric text string with a maximum length of 32 characters.

Service Type The type of service (0=Requester, 1=Provider).

Service Port Name (Unicode) The name of the service port where the message was intercepted. Also known as the Web Services Description Language (WSDL) port name. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters. An asterisk (*) value includes all service port names.

Operation Name (Unicode) The name of the Web Services Description Language (WSDL) operation contained in the intercepted message. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters. An asterisk (*) value includes all operations for the associated service port name.

Local Hostname (Unicode) The fully qualified name of the host computer on which the data collector is running, and where the message was intercepted. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters.

Local IP Address (Unicode) The IP address of the computer on which the data collector is running, and where the message was intercepted. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters.

Application Server Environment The type of runtime environment in which the data collector is running. The format of this attribute is an integer.

The valid values for this attribute are:

- 1 = WebSphere Application Server
- 2 = .NET
- 3 = WebLogic_Server
- 4 = JBoss
- 5 = CICS
- 6 = SAP
- 7 = WebSphere_Community_Edition
- 8 = DataPower
- 9 = Service_Component_Architecture
- 10 = WebSphere_Message_Broker
- 11 = WebSphere_MQ
- 12 = Oracle_Application_Server
- 13 = Generic_Application_Server

Application Server Node Name (Unicode) The node name of the application server where the message was intercepted. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters. Depending on the application server environment, this attribute might be empty.⁹

Application Server Cell Name (Unicode) The name of the application server cell where the message was intercepted. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters. Depending on the application server environment, this attribute might be empty.¹⁰

Application Server Name (Unicode) The name of the application server where the message was intercepted. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters.¹¹

Application Server Cluster Name (Unicode) The name of the application server cluster where the message was intercepted. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters. Depending on the application server environment, this attribute might be empty.¹²

Table version (Unicode) The version of this table definition. This value does not represent the version of this product. The format of this attribute is an alphanumeric text string with a maximum length of eight characters.

9. For a DataPower SOA appliance, this attribute displays the DataPower host fully qualified domain name (FQDN) unless the `DataPower.multihost.group` property is set to true in the DataPower configuration file. The `DataPower.multihost.group` property is introduced in ITCAM for SOA version 7.2 Fix Pack 1.

10. For a DataPower SOA appliance, this attribute displays the DataPower host alias if the `DataPower.alias` property is set in the DataPower configuration file. The `DataPower.alias` property is introduced in ITCAM for SOA version 7.2 Fix Pack 1.

11. For a DataPower SOA appliance, this attribute displays the value of the `DataPower.displaygroup` property if the property is set in the DataPower configuration file.

12. For a DataPower SOA appliance, this attribute displays the DataPower domain name if the `DataPower.dimension.domain` property is set to true in the DataPower configuration file. The `DataPower.dimension.domain` property is introduced in ITCAM for SOA version 7.2 Fix Pack 1.

Interval Begin Time The inclusive beginning date and time of the monitoring interval when the message was intercepted.

When this attribute is displayed in a view in the Tivoli Enterprise Portal, it is expressed as a date and time in Greenwich Mean Time (GMT). This attribute is saved in the Tivoli Data Warehouse as a number in the format *CYYMMDDhhmmssnnn*. For example, a value for October 18 2008 at 19:45:17:010 GMT is stored as *1081018194517010*, where:

C = the century marker (for example, *1*)
YY = the year (for example, *08* for 2008)
MM = the month (for example, *10* for October)
DD = the day of the month (for example, *18* for the 18th day)
hh = the hour portion of the time (for example, *19* for 7:00 p.m.)
mm = the minutes portion of the time (for example, *45* for 45 minutes past the hour)
ss = the seconds portion of the time (for example, *17*)
nnn = the milliseconds portion of the time (for example, *010* for 10 milliseconds)

Interval End Time The exclusive end date and time of the monitoring interval when the message was intercepted.

When this attribute is displayed in a view in the Tivoli Enterprise Portal, it is expressed as a date and time in Greenwich Mean Time (GMT). This attribute is saved in the Tivoli Data Warehouse as a number in the format *CYYMMDDhhmmssnnn*. For example, a value for October 18 2008 at 19:45:17:010 GMT is stored as *1081018194517010*, where:

C = the century marker (for example, *1*)
YY = the year (for example, *08* for 2008)
MM = the month (for example, *10* for October)
DD = the day of the month (for example, *18* for the 18th day)
hh = the hour portion of the time (for example, *19* for 7:00 p.m.)
mm = the minutes portion of the time (for example, *45* for 45 minutes past the hour)
ss = the seconds portion of the time (for example, *17*)
nnn = the milliseconds portion of the time (for example, *010* for 10 milliseconds)

Interval Length The length of the monitoring interval, in minutes. The format of this attribute is an integer.

Average Message Length The average message length, in bytes, observed during this interval (including headers when possible). The value of this attribute is set to *-1* if no messages were observed for the operation during the monitoring interval. The format of this attribute is an integer.

Message Count The number of messages transmitted during this interval. The format of this attribute is an integer.

Elapsed Time Message Count The number of messages transmitted during this interval that contain an elapsed time value. The format of this attribute is an integer.

Average Elapsed Message Round Trip Time The average elapsed round-trip time, in milliseconds. This average does not include values for round-trip time that were set to *-1* (which indicates entry requests, for which there is no elapsed response time, or which indicates that no responses were intercepted during the monitored interval).

Service Port Name : Operation Name : Service Type : Requester ID (Unicode)

The concatenation of the service port namespace, service port name, operation namespace, operation name, service type, and requester identity. The format of this attribute is an alphanumeric text string with a maximum length of 464 characters.

Fault Count The number of faults observed during this interval. The format of this attribute is an integer.

Max Message Length The length of the largest message, in bytes, observed during this monitoring interval. When no messages are observed during this interval, the value of this attribute is set to -1. The format of this attribute is an integer.

Max Elapsed Time The longest elapsed time, in milliseconds, of any message observed during this monitoring interval. When no messages are observed during this interval, the value of this attribute is set to -1. The format of this attribute is an integer.

Min Message Length The length of the shortest message, in bytes, observed during this monitoring interval. When no messages are observed during the monitoring interval, the value of this attribute is set to -1. The format of this attribute is an integer.

Min Elapsed Time The shortest elapsed time, in milliseconds, of any message observed during the monitoring interval. When no messages are observed during this interval, the value of this attribute is set to -1. The format of this attribute is an integer.

Message Length Std Dev The standard deviation, in bytes, of all message lengths observed during the monitoring interval. The format of this attribute is an integer.

Elapsed Message Round Trip Time Std Dev The standard deviation of all elapsed round-trip times, in milliseconds, observed during this monitoring interval. This calculation does not include values for response time that were set to -1 (which indicates entry requests, or indicates that no responses were intercepted during the monitoring interval). The format of this attribute is an integer.

Interval Status The status of this monitoring interval (Incomplete=1, Complete=2). A value of 1 (Incomplete) means that the end of the current monitoring interval was not yet reached, and it is possible that not all messages were received for the entire interval. The format of this attribute is an integer.

Filler1 This attribute is for internal use only and is not intended for inclusion in a query or to be displayed in a table view.

Service Port Namespace (Unicode) The namespace that is combined with the service port name to fully qualify the Web Services Description Language (WSDL) service port that is monitored by the data collector. The format of this attribute is an alphanumeric text string with a maximum of 128 characters.

Operation Namespace (Unicode) The namespace that is combined with the operation name to fully qualify the Web Services Description Language (WSDL) operation that is monitored by the data collector. The format of this attribute is an alphanumeric text string with a maximum length of 128 characters.

Port Number The number of the port, 0 to 65535, on which the application server runtime environment that is being monitored is listening. If the port number is

unknown, this attribute is set to -1. The format of this attribute is an integer. If you configure your WebSphere Application Server version 6.1 environment with security to prevent unauthenticated users from connecting to your Java Management Extensions (JMX) server, the data collector cannot obtain the port number. As a result, this attribute is set to a value of 0.

Unique Key (Unicode) This attribute is for internal use only and is used to uniquely identify the combination of service port namespace, service port name, operation namespace, operation name, service type, and requester identity in the row of data in the table. The key that is generated from this combination is guaranteed to be unique within the table for the specific data collector, but might not be unique across multiple data collectors. This attribute is not displayed in a table view and is not intended for inclusion in a query based on this attribute group.

This attribute is not useful for queries, but might be useful for situation definitions. Use it as the Display Item in the situation definition if you want to differentiate the situation and have IBM Tivoli Monitoring generate individual situation events for each row in the table that meets the situation threshold.

Requester Identity (Unicode) The identity of the service requester. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters.

Tivoli Enterprise Monitoring Agent Info (Unicode) Provides information about the Tivoli Enterprise Monitoring Agent that might be used to link to other related workspaces. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters. This attribute is available only for ITCAM for SOA monitoring agents at version 7.1.0 or later.

DC Info (Unicode) Provides information about the data collector that might be used to link to other related workspaces. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters. This attribute is available only for ITCAM for SOA monitoring agents at version 7.1.0 or later.

Service Port Name Type The type of the service port. The port is identified in the Service Port Name attribute. The format of this attribute is an integer.

The valid values are:

- 1 = WSDL_Port_Name
- 2 = WSDL_Service_Name
- 3 = WSDL_Service_URI
- 4 = URI
- 5 = SCA_Component_Name
- 6 = Generic_Name
- 7 = CICS_Web_Service_Name
- 8 = Message_Flow_For_SOAP
- 9 = Message_Flow_For_XML
- 10 = Message_Flow_For_Other
- 11 = Service_Integration_Bus
- 12= BPD

Request Count The number of request messages transmitted during this interval. The format of this attribute is an integer. This attribute is valid only for ITCAM for SOA version 7.1.1 or later.

Interval Response Count The number of response messages transmitted during this interval whose corresponding request message was observed during this same interval. This count includes both valid responses and fault responses and is less than or equal to the Elapsed Time Message Count attribute. The format of this attribute is an integer. This attribute is valid only for ITCAM for SOA version 7.1.1 or later.

Interval Fault Count The number of fault response messages transmitted during this interval whose corresponding request message was observed during this same interval. This count is less than or equal to the Fault Count attribute. The format of this attribute is an integer. This attribute is valid only for ITCAM for SOA version 7.1.1 or later.

Missing Response Percentage The percentage of request messages transmitted during this interval that did not have a corresponding response. A response whose request was observed in a previous interval is not counted. This attribute is different from the Missing Valid Response Percentage attribute, because a fault is counted as a response. The format of this attribute is an integer. This attribute is valid only for ITCAM for SOA version 7.1.1 or later.

Missing Valid Response Percentage The percentage of request messages transmitted during this interval that did not have a corresponding valid response. A response whose request was observed in a previous interval is not counted. This attribute is different from the Missing Response Percentage attribute, because a fault is not counted as a valid response. The format of this attribute is an integer. This attribute is valid only for ITCAM for SOA version 7.1.1 or later.

Valid Response Percentage The percentage of response messages transmitted during this interval that were not faults. The format of this attribute is an integer. This attribute is valid only for ITCAM for SOA version 7.1.1 or later.

Services Message Metric_610 attributes

This attribute group represents metric information about instances of messages that flowed through an interception point observed by the ITCAM for SOA data collectors.

For ITCAM for SOA version 6.1.0, the data in this attribute group was used primarily by the IBM Web Services Navigator when it was configured to retrieve data from the Tivoli Data Warehouse. For ITCAM for SOA version 7.1.0 and later, the IBM Web Services Navigator no longer uses this attribute group, though it is still available for you to use if required. Do not enable historical data collection for this attribute group unless you are using it for some custom purpose.

The Services Message Metric_610 attributes can contain a significant number of rows depending on the number of times each service is called. Due to the large amount of data that can be generated, when historical data collection is not enabled, queries can retrieve data collected only during the previous 10 minutes. When historical data collection is enabled, or for any real-time query, data collected only during the previous 5 minutes is available. At any time, a maximum of 30,000 rows of data is kept in memory and available to be returned as a result of a query.

Application Server Cell Name (Unicode) The name of the application server cell where the message was intercepted. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters. Depending on the application server environment, this attribute might be empty.

Application Server Cluster Name (Unicode) The name of the application server cluster where the message was intercepted. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters. Depending on the application server environment, this attribute might be empty.

Application Server Environment The type of application server runtime environment in which the data collector is running.

The valid values for this attribute are:

- 1 = WebSphere Application Server
- 2 = .NET
- 3 = WebLogic_Server
- 4 = JBoss
- 5 = CICS
- 6 = SAP
- 7 = WebSphere_Community_Edition
- 8 = DataPower
- 9 = Service_Component_Architecture
- 10 = WebSphere_Message_Broker
- 11 = WebSphere_MQ
- 12 = Oracle_Application_Server
- 13 = Generic_Application_Server

The Service Component Architecture (SCA) application server runtime environment is not represented in this list of values because it shares the application server runtime environment with the WebSphere Application Server data collector.

Port Number The number of the port, 0 - 65535, on which the application server runtime environment that is being monitored is listening. If the port number value is unknown, it is set to -1. The format of this attribute is an integer.

Application Server Name (Unicode) The name of the application server where the message was intercepted. The format of this attribute is an alphanumeric text string with a maximum of 64 characters.

Application Server Node Name (Unicode) The node name of the application server where the message was intercepted. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters. Depending on the application server environment, this attribute might be empty.

Current Correlator (Unicode) The service flow identifier, also known as a child correlator, assigned at an interception point to an incoming or outgoing message, that allows the message to be traced back to the instance of a message flow in an interception point. The format of this attribute is an alphanumeric text string with a maximum length of 128 characters.

Elapsed Message Round Trip Time The elapsed round-trip time, in milliseconds, calculated at the data collector. The value is set to -1 for entry requests. The format of this attribute is an integer.

Fault Response Indication Indicates if a message contains a fault response (1=True, 0=False).

Local Hostname (Unicode) The fully qualified name of the host computer where the message was intercepted. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters.

Local IP Address (Unicode) The IP address of the computer where the message was intercepted. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters.

Message Interception Location The location where the message was intercepted (0= Client_Request, 1= Client_Response, 2= Server_Enter, 3= Server_Leave).

Message Interception Time The date and time when the message was intercepted.

When this attribute is displayed in a view in the Tivoli Enterprise Portal, it is expressed as a date and time in Greenwich Mean Time (GMT). This attribute is saved in the Tivoli Data Warehouse as a number in the format *CYYMMDDhhmmssnnn*. For example, a value for October 18 2008 at 19:45:17:010 GMT is stored as *1081018194517010*, where:

C = the century marker (for example, *1*)

YY = the year (for example, *08* for 2008)

MM = the month (for example, *10* for October)

DD = the day of the month (for example, *18* for the 18th day)

hh = the hour portion of the time (for example, *19* for 7:00 p.m.)

mm = the minutes portion of the time (for example, *45* for 45 minutes past the hour)

ss = the seconds portion of the time (for example, *17*)

nnn = the milliseconds portion of the time (for example, *010* for 10 milliseconds)

Message Length The length of the message (including headers when possible), in bytes. The format of this attribute is an integer.

One Way Message Specifies if the message is one-way (True) or two-way (False). Sometimes, a two-way message might be flagged as one-way if the application server runtime environment does not provide the data collector with a reliable one-way or two-way indicator at certain interception points.

Origin Node The subnode name of the managed system. The format of this attribute is an alphanumeric text string with a maximum length of 32 characters.

Operation Name (Unicode) The name of the Web Services Description Language (WSDL) operation that is contained in the intercepted message. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters.

Operation Namespace (Unicode) The namespace of the Web Services Description Language (WSDL) operation monitored by the data collector. The namespace, combined with the operation name, comprises a fully qualified name for the WSDL operation. The format of this attribute is an alphanumeric text string with a maximum length of 128 characters.

Previous Correlator (Unicode) The From Service Operation Code and To Service Operation Code identifier in a message that was assigned by the previous interception point, also known as the parent correlator. The format of this attribute is an alphanumeric text string with a maximum length of 128 characters.

Remote Hostname (Unicode) The name of the host computer that called the service. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters.

Remote IP Address (Unicode) The IP address of the remote computer that called the service. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters.

Service Port Name (Unicode) The name of the service port where the message was intercepted, also known as the Web Services Description Language (WSDL) port name. The format of this attribute is an alphanumeric text string with a maximum of 64 characters. Use the asterisk (*) wildcard character to include all service names.

Service Port Namespace (Unicode) The namespace of the Web Services Description Language (WSDL) port. The namespace, combined with the service port name, comprises a fully qualified name for the service port. The format of this attribute is an alphanumeric text string with a maximum of 128 characters. This attribute might contain no value (null).

Thread Identifier (Unicode) The identifier of the thread that started the data collector. The format of this attribute is an alphanumeric text string with a maximum length of 32 characters.

Table version (Unicode) The version of this table definition. This value does not represent the version of this product. The format of this attribute is an alphanumeric text string with a maximum length of 8 characters.

Endpoint Inventory attributes

This attribute group describes the attributes that make up the Endpoint Inventory table, which is used in the Endpoint Performance Summary workspace. The Endpoint Inventory table contains metrics for a Service Endpoint. This table is populated when the agent receives metric files that contain an endpoint address.

Origin Node Used to identify the application environment instance. The valid format is an alphanumeric string, with a maximum of 32 characters.

Service Type Specifies the type of service. The valid format is a 2-byte integer.

Valid values are:

- 0=Requester
- 1=Provider

Service Port Name (Unicode) The name of the service port where the message was intercepted. This name is also known as the *Web Services Description Language (WSDL) port name*. The valid format is an alphanumeric string, with a maximum of 64 characters. You can also specify an asterisk (*) as a wildcard to include all service port names.

Endpoint Address (Unicode) The address of the service endpoint that the message was received on or set to. The valid format is an alphanumeric string, with a maximum of 256 characters.

Endpoint Address Hash (Unicode) An MD5 sum of the raw endpoint address, before being truncated. The valid format is an alphanumeric string, with a maximum of 32 characters.

Local Hostname (Unicode) The fully qualified name of the host computer on which the data collector is running, and where the message was intercepted. For example, host01.example.com. The valid format is an alphanumeric string, with a maximum of 64 characters.

Local IP Address (Unicode) The IP address of the host where the message was intercepted. The valid format is an alphanumeric string, with a maximum of 64 characters.

Application Server Environment Specifies the application server environment. The valid format is a 2-byte integer.

Valid values are:

- 1 = WebSphere Application Server
- 2 = .NET
- 3 = WebLogic_Server
- 4 = JBoss
- 5 = CICS
- 6 = SAP
- 7 = WebSphere_Community_Edition
- 8 = DataPower
- 9 = Service_Component_Architecture
- 10 = WebSphere_Message_Broker
- 11 = WebSphere_MQ
- 12 = Oracle_Application_Server
- 13 = Generic_Application_Server

Application Server Node Name (Unicode) The node name of the application server where the message was intercepted. The valid format is an alphanumeric string, with a maximum of 64 characters. Depending on the application server environment, this attribute might be empty.

Application Server Cell Name (Unicode) The name of the application server cell where the message was intercepted. The valid format is an alphanumeric string, with a maximum of 64 characters. Depending on the application server environment, this attribute might be empty.

Application Server Name (Unicode) The name of the application server where the message was intercepted. The valid format is an alphanumeric string, with a maximum of 64 characters.

Application Server Cluster Name (Unicode) The name of the application server cluster where the message was intercepted. The valid format is an alphanumeric string, with a maximum of 64 characters. Depending on the application server environment, this attribute might be empty.

Table Version (Unicode) The version of this table definition. This value does not represent the version of this product. The valid format is an alphanumeric string, with a maximum of 8 characters.

Interval Begin Time The inclusive beginning date and time of the monitoring interval when the message was intercepted. The valid format is a 16-character timestamp.

When this attribute is displayed in a view in the Tivoli Enterprise Portal, it is expressed as a date and time in Greenwich Mean Time (GMT). This attribute is saved in the Tivoli Data Warehouse as a number in the format *CYYMMDDhhmmssnnn*. For example, a value for October 18, 2008 at 19:45:17:010 GMT is stored as *1081018194517010*, where:

C = the century marker
YY = the year
MM = the month
DD = the day of the month
hh = the hour portion of the time
mm = the minutes portion of the time
ss = the seconds portion of the time
nnn = the milliseconds portion of the time

Interval End Time The exclusive end date and time of the monitoring interval when the message was intercepted. The valid format is a 16-character timestamp.

When this attribute is displayed in a view in the Tivoli Enterprise Portal, it is expressed as a date and time in Greenwich Mean Time (GMT). This attribute is saved in the Tivoli Data Warehouse as a number in the format *CYYMMDDhhmmssnnn*. For example, a value for October 18, 2008 at 19:45:17:010 GMT is stored as *1081018194517010*, where:

C = the century marker
YY = the year
MM = the month
DD = the day of the month
hh = the hour portion of the time
mm = the minutes portion of the time
ss = the seconds portion of the time
nnn = the milliseconds portion of the time

Interval Length The length of the interval in minutes. The valid format is a 4-byte integer.

Average Message Length The average message length, in bytes, observed during this interval (including headers when possible). The value of this attribute is set to -1 if no messages were observed for the operation during the monitoring interval. The valid format is a 4-byte integer.

Message Count The number of messages observed during this interval. The valid format is a 4-byte integer.

Elapsed Time Message Count The number of messages observed during this interval that contain an elapsed time value. The valid format is a 4-byte integer.

Average Elapsed Message Round Trip Time Average elapsed round-trip time in milliseconds. This average does not include values which were set to -1 to indicate entry requests. The value of this attribute is set to -1 when no responses were intercepted during the monitored interval. The valid format is a 4-byte integer.

Fault Count The number of faults observed during this interval. The valid format is a 4-byte integer.

Max Message Length The length of the longest message, in bytes, observed during this monitoring interval. When no messages are observed during this interval, the value of this attribute is set to -1. The valid format is a 4-byte integer.

Max Elapsed Time The longest elapsed time, in milliseconds, of any message observed during this monitoring interval. When no messages are observed during this interval, the value of this attribute is set to -1. The valid format is a 4-byte integer.

Min Message Length The length of the shortest message, in bytes, observed during this monitoring interval. When no messages are observed during the monitoring interval, the value of this attribute is set to -1. The valid format is a 4-byte integer.

Min Elapsed Time The shortest elapsed time, in milliseconds, of any message observed during the monitoring interval. When no messages are observed during this interval, the value of this attribute is set to -1. The valid format is a 4-byte integer.

Message Length Std Dev The standard deviation, in bytes, of all message lengths observed during the monitoring interval. The valid format is a 4-byte integer.

Elapsed Message Round Trip Time Std Dev The standard deviation of all elapsed round-trip times, in milliseconds, observed during this monitoring interval. The value of this attribute is set to -1 when no responses were intercepted during the monitored interval. The valid format is a 4-byte integer.

Interval Status Specifies the status of the monitoring interval. The valid format is a 2-byte integer.

Valid values are:

1=Incomplete
2=Complete

A value of 1 means that the end of the current monitoring interval has not yet been reached, and it is possible that not all messages have been received for the entire interval. The format of this attribute is an integer.

Filler1 This attribute is for internal use only, it is used as a filler to maintain the data on a word boundary. It does not appear in a table view and is not intended to be included in a query based on this attribute group, but it is included in the list of selectable attributes. The valid format is a 2-byte integer.

Port Namespace (Unicode) The namespace of the Web Services Description Language (WSDL) service port monitored by the data collector. The namespace,

combined with the service port name, comprises a fully qualified name for the service port. The valid format is an alphanumeric string, with a maximum of 128 characters.

Port Number The number of the port, 0 to 65535, on which the application server runtime environment that is being monitored is listening. If the port number is unknown, this attribute is set to -1. The valid format is a 4-byte integer. If you configure your IBM WebSphere Application Server environment with security to prevent unauthenticated users from connecting to your JMX server, the data collector cannot obtain the port number. As a result, this attribute is set to a value of 0.

Unique Key (Unicode) This attribute is for internal use only, and is used to uniquely identify the combination of service port namespace, service port name, operation namespace, operation name, and service type in the row of data in the table. The key that is generated from this combination is guaranteed to be unique within the table for the specific data collector, but might not be unique across multiple data collectors. This attribute is not displayed in a table view and is not intended to be included in a query based on this attribute group, but it is included in the list of selectable attributes. This attribute is not useful for queries, but might be useful for situation definitions. Use it as the Display Item in the situation definition if you want to differentiate the situation and have IBM Tivoli Monitoring generate individual situation events for each row in the table that meets the situation threshold. The valid format is an alphanumeric string, with a maximum of 32 characters.

TEMA Info (Unicode) Provides information related to the Tivoli Enterprise Monitoring Agent that might be used to link to other related workspaces. The valid format is an alphanumeric string, with a maximum of 64 characters.

DC Info (Unicode) Information related to the data collector used to link to other workspaces related to this data collector. The valid format is an alphanumeric string, with a maximum of 64 characters.

Service Port Name Type The type of service port, such as the Web Services Description Language (WSDL) port name, or the SCA Component name. The valid format is a 2-byte integer.

Valid values are:

- 1=WSDL_Port_Name
- 2=WSDL_Service_Name
- 3=WSDL_Service_URI
- 4=URI
- 5=SCA_Component_Name
- 7=CICS_Web_Service_Name
- 8=Message_Flow_For_SOAP
- 9=Message_Flow_For_XML
- 10=Message_Flow_For_Other
- 11=Service_Integration_Bus
- 12=BPD

Request Count The number of request messages observed during this interval. The valid format is a 4-byte integer.

Interval Response Count The number of response messages observed during this interval where the corresponding request message was observed during the same interval. This count includes both valid responses and fault responses and is less than or equal to the Elapsed Time Message Count attribute. The valid format is a 4-byte integer.

Interval Fault Count The number of fault response messages observed during this interval where the corresponding request message was observed during the same interval. This count is less than or equal to the Fault Count attribute. The valid format is a 4-byte integer.

Missing Response Percentage The percentage of request messages observed during this interval that did not have a corresponding response. If a response request was observed in a previous interval, it is not counted. This attribute is different from the Missing Valid Response Percentage attribute, because a fault is not counted as a valid response. The valid format is a 4-byte integer.

Missing Valid Response Percentage The percentage of request messages observed during this interval that did not have a corresponding valid response. If a response request was observed in a previous interval, it is not counted. This attribute is different from the Missing Response Percentage attribute, because a fault is counted as a response. The valid format is a 4-byte integer.

Valid Response Percentage The percentage of response messages observed during this interval that were not faults. The valid format is a 4-byte integer.

Business Process Events attributes

The Business Process Events attributes represent events associated with BPEL processes, activities, and human tasks.

Origin Node The application environment instance from which the data in this table originated. The valid format is an alphanumeric string, with a maximum of 32 characters.

Environment Code The code for the application server and computer system environment. The valid format is an alphanumeric string, with a maximum of 36 characters.

Event Source ID The event source ID. The valid format is a 2-byte integer.

Event Kind The kind of event. The valid format is a 2-byte integer.

The available values are:

- 1 = Process
- 2 = Task
- 10 = Assign
- 11 = Compensate
- 12 = Empty
- 13 = Flow
- 14 = ForEach
- 15 = Invoke
- 16 = Pick

- 17 = Receive
- 18 = RepeatUntil
- 19 = Reply
- 20 = Rethrow
- 21 = Sequence
- 22 = Scope
- 23 = Script
- 24 = Staff
- 25 = Switch
- 26 = Terminate
- 27 = Throw
- 28 = Wait
- 29 = While
- 50 = Case
- 51 = CompensationHandler
- 52 = Catch
- 53 = CatchAll
- 54 = OnAlarm
- 55 = OnEvent
- 56 = OnMessage
- 57 = ForEachParallel
- 58 = IorinGateway

Event Nature The nature of the event. The valid format is a 2-byte integer.

The available values are:

- 1 = ENTRY
- 2 = EXIT
- 3 = FAILED
- 4 = CUSTOM
- 5 = INVOCATION
- 6 = FAILURE
- 7 = STARTED
- 8 = COMPLETED
- 9 = TERMINATED
- 10 = DELETED
- 11 = CAUGHT
- 12 = THROWN
- 13 = EXPECTED
- 14 = ACTIVE
- 15 = READY
- 16 = RESOURCE_ASSIGNED
- 17 = LOOP_CONDITION_TRUE
- 18 = LOOP_CONDITION_FALSE
- 19 = MULTIPLE_INSTANCES_STARTED
- 20 = ACTIVATED

- 21 = PARALLEL_INSTANCES_STARTED
- 22 = FCOMPLETED
- 23 = JUMPED
- 24 = STOPPED
- 25 = CREATED
- 26 = CONDFALSE
- 27 = CONDTRUE
- 28 = ALLCONDFALSE

Event Timestamp Event date and time timestamp. The number of milliseconds since January 1, 1970 UTC. The valid format is an alphanumeric string, with a maximum of 16 characters.

INSTANCE ID The instance ID. For a process this value is PIID; for a task, this value is TKIID; for an activity, this value is AIID. This column is also known as *ECS_Current_ID*. The valid format is an alphanumeric string, with a maximum of 40 characters.

ECS Parent ID The Parent Instance ID of this instance. If the length of the Parent Instance ID of this instance is greater than 40 characters, the attribute contains an MD5 sum of the Parent Instance ID. The valid format is an alphanumeric string, with a maximum of 40 characters.

Template ID The template ID of this instance. For a process, this value is PTID; for a task, this value is TKTID; for an activity, this value is ATID. The valid format is an alphanumeric string, with a maximum of 40 characters.

State The state of a process, a task, or an activity. The valid format is a 2-byte integer.

For a process, this value is the process execution state:

- 1 = STATE_READY
- 2 = STATE_RUNNING
- 3 = STATE_FINISHED
- 4 = STATE_COMPENSATING
- 5 = STATE_FAILED
- 6 = STATE_TERMINATED
- 7 = STATE_COMPENSATED
- 8 = STATE_TERMINATING
- 9 = STATE_FAILING
- 11 = STATE_SUSPENDED
- 12 = STATE_COMPENSATION_FAILED

For a task, this value is the current state of this task instance:

- 1 = INACTIVE
- 2 = READY
- 3 = RUNNING
- 5 = FINISHED
- 6 = FAILED
- 7 = TERMINATED

- 8 = CLAIMED
- 12 = EXPIRED
- 101 = FORWARDED

For a normal activity, this value is the current state of this activity instance:

- 1 = STATE_INACTIVE
- 2 = STATE_READY
- 3 = STATE_RUNNING
- 4 = STATE_SKIPPED
- 5 = STATE_FINISHED
- 6 = STATE_FAILED
- 7 = STATE_TERMINATED
- 8 = STATE_CLAIMED
- 11 = STATE_WAITING
- 12 = STATE_EXPIRED
- 13 = STATE_STOPPED

For a scope activity, this value is the current state of this scope activity instance:

- 1 = STATE_READY
- 2 = STATE_RUNNING
- 3 = STATE_FINISHED
- 4 = STATE_COMPENSATING
- 5 = STATE_FAILED
- 6 = STATE_TERMINATED
- 7 = STATE_COMPENSATED
- 8 = STATE_COMPENSATION_FAILED
- 9 = STATE_FAILING
- 10 = STATE_SKIPPED
- 11 = STATE_COMPENSATION_FAILING
- 12 = STATE_FAULTHANDLER_FAILING
- 13 = STATE_FINISHING
- 14 = STATE_STOPPED

Associated Instance Identifier For a task event, this value is the associated activity instance ID; for an activity event, this value is the associated task instance ID. The valid format is an alphanumeric string, with a maximum of 40 characters.

Process Instance ID Associated Process instance ID of the BPEL task or activity. The valid format is an alphanumeric string, with a maximum of 40 characters.

Process Template ID Associated Process template ID of the BPEL task or activity. The valid format is an alphanumeric string, with a maximum of 40 characters.

Associated Error Associated error ID. See “BPM Associated Errors attributes” on page 294. The valid format is an alphanumeric string, with a maximum of 36 characters.

Table Version The version of this table definition. The valid format is an alphanumeric string, with a maximum of 8 characters.

Request Type Identifies the type of a query. For internal use only. The valid format is a 2-byte integer.

Bucket Size Indicates the pull bucket size for each SDMS retrieval. The valid format is a 4-byte integer.

BPM Associated Errors attributes

This table provides error information for BPM topology diagram.

Origin Node Used to identify the TEMA instance from which the data in this table originated. The valid format is an alphanumeric string, with a maximum of 32 characters.

Environment Code The code for the application server and computer system environment. The valid format is an alphanumeric string, with a maximum of 36 characters.

Associated Error A key that uniquely identifies this row of data. The valid format is an alphanumeric string, with a maximum of 36 characters.

Component Type Type of the component that this error is from. The valid format is a 2-byte integer.

Valid fixed values are:

- 1 = Process
- 2 = Task
- 3 = SCA.EJB.Export
- 4 = SCA.EIS.Export
- 5 = SCA.JAX-WS.Export
- 6 = EventEmitter
- 7 = MEDIATION.FLOW
- 8 = MEDIATION.FLOW.PRIMITIVE
- 9 = MEDIATION.FLOW.WSRR
- 10 = Assign
- 11 = Compensate
- 12 = Empty
- 13 = Flow
- 14 = ForEach
- 15 = Invoke
- 16 = Pick
- 17 = Receive
- 18 = RepeatUntil
- 19 = Reply
- 20 = Rethrow
- 21 = Sequence
- 22 = Scope
- 23 = Script
- 24 = Staff
- 25 = Switch

- 26 = Terminate
- 27 = Throw
- 28 = Wait
- 29 = While
- 30 = SCA.EJB.Import
- 31 = SCA.EIS.Import
- 32 = SCA.JAX-WS.Import
- 37 = SCA.HTTP.Import
- 38 = SCA.HTTP.Export
- 50 = Case
- 51 = CompensationHandler
- 52 = Catch
- 53 = CatchAll
- 54 = OnAlarm
- 55 = OnEvent
- 56 = OnMessage
- 57 = ForEachParallel
- 58 = IorinGateway

Process Template ID Associated Process template ID of the BPEL task or activity. The valid format is an alphanumeric string, with a maximum of 40 characters.

Mediation Primitive Instance Mediation primitive instance code that identifies the target side of the response. The valid format is an alphanumeric string, with a maximum of 36 characters.

External System External system code that identifies the target side of the response. The valid format is an alphanumeric string, with a maximum of 36 characters.

Fault Namespace The namespace of the fault. The valid format is an alphanumeric string, with a maximum of 128 characters.

Fault Name The name of the fault. The valid format is an alphanumeric string, with a maximum of 128 characters.

Exception The failure exception of task, BPEL process, BPEL activity, or mediation flow. The valid format is an alphanumeric string, with a maximum of 256 characters.

Last Occurrence The date and time the error occurs the last time. The number of milliseconds since January 1, 1970 UTC. The valid format is an alphanumeric string, with a maximum of 16 characters.

Number Occurrences The number of errors observed during one SDMS retrieval interval. The value is set to 0 after SDMS retrieval; the value is increased by one every time the same type of error is received by TEMA. The valid format is a 4-byte integer.

Table Version The version of this table definition. The valid format is an alphanumeric string, with a maximum of 8 characters.

Request Type Identifies the type of a query. For internal use only. The valid format is a 2-byte integer.

Mediation Configuration_610 attributes

This attribute group provides data about the configuration criteria for managed SCA mediation primitives. This attribute group is deprecated since ITCAM for SOA version 7.2.

Hostname (Unicode) The name and port of the host computer on which the managed mediations being configured are running. The format of this attribute is an alphanumeric text string with a maximum of 128 characters. This attribute is deprecated since ITCAM for SOA version 7.2.

Origin Node The node name of the managed system. The format of this attribute is an alphanumeric text string with a maximum length of 32 characters. This attribute is deprecated since ITCAM for SOA version 7.2.

Mediation Container (Unicode) The concatenation of the cell name, node name, and application server name, delimited by semicolon characters, of the application server on which the mediation being configured is running. The format of this attribute is an alphanumeric text string with a maximum length of 192 characters. This attribute is deprecated since ITCAM for SOA version 7.2.

Mediation Qualifier (Unicode) The concatenation of the application name, module name, and primitive name, delimited by semicolon characters, of the mediation being configured. The format of this attribute is an alphanumeric text string with a maximum of 192 characters. This attribute is deprecated since ITCAM for SOA version 7.2.

Application Server Environment The runtime environment in which the data collector is running. The only value that is valid for this attribute is *WebSphere_Application_Server*. This attribute is deprecated since ITCAM for SOA version 7.2.

Primitive Type The type of managed mediation primitive. This attribute is deprecated since ITCAM for SOA version 7.2.

These are the possible values:

- ManagedMessageLogger
- ManagedXSLT
- ManagedMessageFilter
- ManagedFail

Primitive Name (Unicode) The name of the managed mediation primitive to be configured. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters. This attribute is deprecated since ITCAM for SOA version 7.2.

Primitive Property (Unicode) The name of the managed mediation primitive property to be configured. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters. This attribute is deprecated since ITCAM for SOA version 7.2.

Table version (Unicode) The version of this table definition. This value does not represent the version of this product. The format of this attribute is an alphanumeric text string with a maximum length of eight characters. This attribute is deprecated since ITCAM for SOA version 7.2.

Detected A flag indicating if the managed mediation primitive is detected in the mediation runtime environment (1=True, 0=False). This attribute is deprecated since ITCAM for SOA version 7.2.

Internal tables

ITCAM for SOA uses the following internal tables to support topology display. You can access these tables using Tivoli Monitoring tools, for example, the situation editor.

Relationship Request Metrics attributes

This attribute group is for internal use by ITCAM for SOA for providing support for the service-to-service topology display. It contains the provider enter metrics for relationships between operation instances.

The Tivoli Enterprise Monitoring Agent does not save data for this attribute group unless historical data collection is enabled. Because the Tivoli Enterprise Monitoring Agent is not notified when historical data collection is enabled and disabled for the attribute group, it uses the presence or absence of requests for historical data to determine the state of history collection in the following way:

- When the Tivoli Enterprise Monitoring Agent is restarted, the monitoring agent assumes that historical data collection is enabled and starts saving data.
- When the Tivoli Enterprise Monitoring Agent receives a request for historical data for this attribute group, the monitoring agent starts saving data for the attribute group.
- If a request for historical data is not received for this attribute group for 70 minutes, the monitoring agent assumes that historical data collection is disabled, and stops saving data for this attribute group.

Origin Node The application environment instance from which the data in this table originated. The format of this attribute is an alphanumeric text string with a maximum of 32 characters.

From Service Operation Code (Unicode) The service port operation code that identifies the source of the request. This column can be empty for relationships discovered at server enter interception points. The format of this attribute is an alphanumeric text string with a maximum of 36 characters.

To Service Operation Code (Unicode) The service port operation code that identifies the target of the request. This column can be empty for relationships discovered at client response interception points. The format of this attribute is an alphanumeric text string with a maximum of 36 characters.

Row Index A unique index for each row in this table. This value is an increasing number. When the table is created, the first row of the relationship table has an index value of 1. These values might not be contiguous if rows are deleted from the table. The format of this attribute is an integer.

Interval Begin Time The inclusive beginning date and time of the monitoring interval when the message was intercepted.

When this attribute is displayed in a view in the Tivoli Enterprise Portal, it is expressed as a date and time in Greenwich Mean Time (GMT). This attribute is saved in the Tivoli Data Warehouse as a number in the format *CYYMMDDhhmmssnnn*. For example, a value for October 18 2008 at 19:45:17:010 GMT is stored as *1081018194517010*, where:

C = the century marker (for example, *1*)
YY = the year (for example, *08* for 2008)
MM = the month (for example, *10* for October)
DD = the day of the month (for example, *18* for the 18th day)
hh = the hour portion of the time (for example, *19* for 7:00 p.m.)
mm = the minutes portion of the time (for example, *45* for 45 minutes past the hour)
ss = the seconds portion of the time (for example, *17*)
nnn = the milliseconds portion of the time (for example, *010* for 10 milliseconds)

Interval End Time The exclusive end date and time of the monitoring interval when the message was intercepted.

When this attribute is displayed in a view in the Tivoli Enterprise Portal, it is expressed as a date and time in Greenwich Mean Time (GMT). This attribute is saved in the Tivoli Data Warehouse as a number in the format *CYYMMDDhhmmssnnn*. For example, a value for October 18 2008 at 19:45:17:010 GMT is stored as *1081018194517010*, where:

C = the century marker (for example, *1*)
YY = the year (for example, *08* for 2008)
MM = the month (for example, *10* for October)
DD = the day of the month (for example, *18* for the 18th day)
hh = the hour portion of the time (for example, *19* for 7:00 p.m.)
mm = the minutes portion of the time (for example, *45* for 45 minutes past the hour)
ss = the seconds portion of the time (for example, *17*)
nnn = the milliseconds portion of the time (for example, *010* for 10 milliseconds)

Average Message Length The average request message length, in bytes, observed during the current monitoring interval. The format of this attribute is an integer.

Min Message Length The length of the shortest request message, in bytes, observed during the current monitoring interval. The format of this attribute is an integer.

Max Message Length The length of the longest request message, in bytes, observed during the current monitoring interval. The format of this attribute is an integer.

Message Count The number of messages transmitted during the current monitoring interval that have a valid message length. The format of this attribute is an integer.

Table version (Unicode) The version of this table definition. This value does not represent the version of this product. The format of this attribute is an alphanumeric text string with a maximum of eight characters.

Message Interception Location The location where the message was intercepted (2=Server Enter). The format of this attribute is an integer value of 2.

Request Type The type of a query. The format of this attribute is an integer. Any other request type values are assumed to be a normal query request. For example, return requested rows.

Request Return Code The return code for the request identified by the Request Type attribute. The format of this attribute is an integer.

From Mediation Subflow (Unicode) Mediation subflow code that identifies the caller side of the response. The valid format is an alphanumeric string, with a maximum of 36 characters. Available with agent version 7.2 or later.

From Mediation Subflow Instance (Unicode) Mediation subflow instance code that identifies the caller side of the response. The valid format is an alphanumeric string, with a maximum of 36 characters. Available with agent version 7.2 or later.

To Mediation Primitive (Unicode) Mediation primitive code that identifies the target side of the response. The valid format is an alphanumeric string, with a maximum of 36 characters. Available with agent version 7.2 or later.

To Mediation Primitive Instance (Unicode) Mediation primitive instance code that identifies the target side of the response. The valid format is an alphanumeric string, with a maximum of 36 characters. Available with agent version 7.2 or later.

To External System (Unicode) External system code that identifies the target side of the response. The valid format is an alphanumeric string, with a maximum of 36 characters. Available with agent version 7.2 or later.

To Mediation Subflow (Unicode) Mediation subflow code that identifies the target side of the response. The valid format is an alphanumeric string, with a maximum of 36 characters. Available with agent version 7.2 or later.

To Mediation Subflow Instance (Unicode) Mediation subflow instance code that identifies the target side of the response. The valid format is an alphanumeric string, with a maximum of 36 characters. Available with agent version 7.2 or later.

Relationship Response Metrics attributes

This attribute group is for internal use by ITCAM for SOA for providing support for the service-to-service topology display. It contains the provider leave and requester response metrics for relationships between operation instances.

The Tivoli Enterprise Monitoring Agent does not save data for this attribute group unless historical data collection is enabled. Because the Tivoli Enterprise Monitoring Agent is not notified when historical data collection is enabled and disabled for the attribute group, it uses the presence or absence of requests for historical data to determine the state of history collection in the following way:

- When the Tivoli Enterprise Monitoring Agent is restarted, the monitoring agent assumes that historical data collection is enabled and starts saving data.
- When the Tivoli Enterprise Monitoring Agent receives a request for historical data for this attribute group, the monitoring agent starts saving data for the attribute group.
- If a request for historical data is not received for this attribute group for 70 minutes, the monitoring agent assumes that historical data collection is disabled, and stops saving data for this attribute group.

The monitoring agent assumes that the Relationship Response Metrics attribute group *is not* enabled for historical data collection if a request for historical data is not received for 70 minutes. When the monitoring agent determines that historical data collection is disabled, it deletes rows after they are retrieved by the SOA Domain Management Server.

Origin Node The application environment instance from which the data in this table originated. The format of this attribute is an alphanumeric text string with a maximum of 32 characters.

From Service Operation Code (Unicode) The service port operation code that identifies the target of the response. This column can be empty for relationships discovered at server enter interception points. The format of this attribute is an alphanumeric text string with a maximum of 36 characters.

To Service Operation Code (Unicode) The service port operation code that identifies the source of the response. This column can be empty for relationships discovered at client response interception points. The format of this attribute is an alphanumeric text string with a maximum of 36 characters.

Called Service Operation Code (Unicode) The service port operation code that identifies the target side of the relationship using data available to the caller. The format of this attribute is an alphanumeric text string with a maximum of 36 characters.

Row Index A unique index for each row in this table. This value is an increasing number. When the table is created, the first row of the relationship table has an index value of 1. These values might not be contiguous, if rows are deleted from the table. The format of this attribute is an integer.

Interval Begin Time The inclusive beginning date and time of the monitoring interval when the message was intercepted.

When this attribute is displayed in a view in the Tivoli Enterprise Portal, it is expressed as a date and time in Greenwich Mean Time (GMT). This attribute is saved in the Tivoli Data Warehouse as a number in the format *CYYMMDDhhmmssnnn*. For example, a value for October 18 2008 at 19:45:17:010 GMT is stored as *1081018194517010*, where:

C = the century marker (for example, *1*)
YY = the year (for example, *08* for 2008)
MM = the month (for example, *10* for October)
DD = the day of the month (for example, *18* for the 18th day)
hh = the hour portion of the time (for example, *19* for 7:00 p.m.)
mm = the minutes portion of the time (for example, *45* for 45 minutes past the hour)
ss = the seconds portion of the time (for example, *17*)
nnn = the milliseconds portion of the time (for example, *010* for 10 milliseconds)

Interval End Time The exclusive end date and time of the monitoring interval when the message was intercepted.

When this attribute is displayed in a view in the Tivoli Enterprise Portal, it is expressed as a date and time in Greenwich Mean Time (GMT). This attribute is saved in the Tivoli Data Warehouse as a number in the format *CYYMMDDhhmmssnnn*. For example, a value for October 18 2008 at 19:45:17:010 GMT is stored as *1081018194517010*, where:

C = the century marker (for example, 1)
YY = the year (for example, 08 for 2008)
MM = the month (for example, 10 for October)
DD = the day of the month (for example, 18 for the 18th day)
hh = the hour portion of the time (for example, 19 for 7:00 p.m.)
mm = the minutes portion of the time (for example, 45 for 45 minutes past the hour)
ss = the seconds portion of the time (for example, 17)
nnn = the milliseconds portion of the time (for example, 010 for 10 milliseconds)

Average Message Length The average response message length, in bytes, observed during the current monitoring interval. The format of this attribute is an integer.

Min Message Length The length of the shortest response message, in bytes, observed during the current monitoring interval. The format of this attribute is an integer.

Max Message Length The length of the largest response message, in bytes, observed during this interval. The format of this attribute is an integer.

Message Count The total number of messages transmitted during the current monitoring interval that have a valid message length. The format of this attribute is an integer.

Fault Count The number of faults observed during the current monitoring interval. The format of this attribute is an integer.

Average Elapsed Message Round Trip Time The average elapsed round-trip time in milliseconds. This does not include values that were set to -1 to indicate entry requests. The format of this attribute is an integer.

Min Elapsed Time The shortest elapsed time, in milliseconds, of any response message observed during the current monitoring interval. The format of this attribute is an integer.

Max Elapsed Time The longest elapsed time, in milliseconds, of any response message observed during the current monitoring interval. The format of this attribute is an integer.

Elapsed Time Message Count The number of response messages transmitted during the current monitoring interval that contain a valid elapsed round-trip time value. The format of this attribute is an integer.

Table version (Unicode) The version of this table definition. This value does not represent the version of this product. The format of this attribute is an alphanumeric text string with a maximum of eight characters.

Message Interception Location The location where the message was intercepted (1= Client Response, 3= Server Leave). The format of this attribute is an integer.

Request Type The type of a query. The format of this attribute is an integer.

Request Return Code The return code for the request identified by the Request Type attribute. The format of this attribute is an integer.

From Mediation Subflow (Unicode) Mediation subflow code that identifies the caller side of the response. The valid format is an alphanumeric string, with a maximum of 36 characters. Available with agent version 7.2 or later.

From Mediation Subflow Instance (Unicode) Mediation subflow instance code that identifies the caller side of the response. The valid format is an alphanumeric string, with a maximum of 36 characters. Available with agent version 7.2 or later.

To Mediation Primitive (Unicode) Mediation primitive code that identifies the target side of the response. The valid format is an alphanumeric string, with a maximum of 36 characters. Available with agent version 7.2 or later.

To Mediation Primitive Instance (Unicode) Mediation primitive instance code that identifies the target side of the response. The valid format is an alphanumeric string, with a maximum of 36 characters. Available with agent version 7.2 or later.

To External System (Unicode) External system code that identifies the target side of the response. The valid format is an alphanumeric string, with a maximum of 36 characters. Available with agent version 7.2 or later.

To Mediation Subflow (Unicode) Mediation subflow code that identifies the target side of the response. The valid format is an alphanumeric string, with a maximum of 36 characters. Available with agent version 7.2 or later.

To Mediation Subflow Instance (Unicode) Mediation subflow instance code that identifies the target side of the response. The valid format is an alphanumeric string, with a maximum of 36 characters. Available with agent version 7.2 or later.

Relationships attributes

This attribute group is for internal use by ITCAM for SOA for capturing call relationships and displaying service-to-service topology. When IBM Web Services Navigator is configured to retrieve data from the Tivoli Data Warehouse, it uses the attribute group for displaying topology data. This attribute group is available for historical data collection.

Origin Node Identifies the subnode from which the data in this table originated. The format of this attribute is an alphanumeric text string with a maximum of 32 characters.

Relationship UID The unique ID for the relationship identified in this row of the table. The format of this attribute is an integer.

From Service Operation Code (Unicode) The service port operation code that identifies the caller side of the call relationship. This column might be empty for relationships discovered at server enter interception points. The format of this attribute is an alphanumeric text string with a maximum of 36 characters.

To Service Operation Code (Unicode) The service port operation code that identifies the target side of the call relationship. This column might be empty for relationships discovered at client response interception points. The format of this attribute is an alphanumeric text string with a maximum of 36 characters.
To Service Operation Code (Unicode) The service port operation code that identifies the target side of the call relationship. This column might be empty for relationships discovered at client response interception points. The format of this attribute is an alphanumeric text string with a maximum of 36 characters.

Called Service Operation Code (Unicode) The service port operation code that identifies the target side of the call relationship using data available to the caller. The format of this attribute is an alphanumeric text string with a maximum of 36 characters.

Last Modified Time This time stamp identifies the time, in seconds, when the row was created or the last time it was modified. The value is expressed in seconds since January 1 1970. The format of this attribute is in Coordinated Universal Time (UTC).

Row Index A unique index for each row in this table. The format of this attribute is an integer.

Table version (Unicode) The version of this table definition. This value does not represent the version of this product. The format of this attribute is an alphanumeric text string with a maximum of eight characters. This attribute is for internal use only and is not intended for inclusion in a query or to be displayed in a table view.

Relationship Type Specifies the relationship type (1=Client Request, 2=Client Response, 4=Provider). The valid format is a 2-byte integer. The *Provider* value is used for a relationship detected at the server enter or server leave interception points.

Request Type The type of query. The format of this attribute is an integer. Any other request type values are assumed to be a normal query request. For example, return requested rows.

Request Return Code The return code for the request identified by the Request Type attribute. The format of this attribute is an integer.

From Mediation Subflow (Unicode) Mediation subflow code that identifies the caller side of the response. The valid format is an alphanumeric string, with a maximum of 36 characters. Available with agent version 7.2 or later.

From Mediation Subflow Instance (Unicode) Mediation subflow instance code that identifies the caller side of the response. The valid format is an alphanumeric string, with a maximum of 36 characters. Available with agent version 7.2 or later.

To Mediation Primitive (Unicode) Mediation primitive code that identifies the target side of the response. The valid format is an alphanumeric string, with a maximum of 36 characters. Available with agent version 7.2 or later.

To Mediation Primitive Instance (Unicode) Mediation primitive instance code that identifies the target side of the response. The valid format is an alphanumeric string, with a maximum of 36 characters. Available with agent version 7.2 or later.

To External System (Unicode) External system code that identifies the target side of the response. The valid format is an alphanumeric string, with a maximum of 36 characters. Available with agent version 7.2 or later.

To Mediation Subflow (Unicode) Mediation subflow code that identifies the target side of the response. The valid format is an alphanumeric string, with a maximum of 36 characters. Available with agent version 7.2 or later.

To Mediation Subflow Instance (Unicode) Mediation subflow instance code that identifies the target side of the response. The valid format is an alphanumeric string, with a maximum of 36 characters. Available with agent version 7.2 or later.

BPM Relationship Type Specifies if an import or export operation is called (-1=Not import related, 7=Import called, 8=Export called, 50=BPD calls SCA). The valid format is a 2-byte integer.

Service Flow Metrics attributes

This attribute group is for internal use by ITCAM for SOA for providing support for the topology display in the IBM Web Services Navigator when it is configured to retrieve data from the Tivoli Data Warehouse.

In this release, the IBM Web Services Navigator uses these attributes to discover service transaction message flows and their metrics. The rows in the Service Flow Metrics attribute group table contain a reference to the relationship table row that contains the **From Service Operation Code (Unicode)** and **To Service Operation Code (Unicode)** attributes. These attributes are used to derive the information about the caller and target operation and its application server environment.

The Tivoli Enterprise Monitoring Agent does not save data for this attribute group unless historical data collection is enabled. Because the Tivoli Enterprise Monitoring Agent is not notified when historical data collection is enabled and disabled for the Service Flow Metrics attribute group, it uses the presence or absence of requests for historical data to determine the state of history collection in the following way:

- When the Tivoli Enterprise Monitoring Agent is restarted, the monitoring agent assumes that historical data collection is disabled and does not save any data for this attribute group.
- When the Tivoli Enterprise Monitoring Agent receives a request for historical data for this attribute group, the monitoring agent starts saving data for the attribute group.
- If a request for historical data is not received for this attribute group for 70 minutes, the monitoring agent assumes that historical data collection is disabled, and stops saving data for this attribute group.

Origin Node The application environment instance from which the data in this table originated. The format of this attribute is an alphanumeric text string with a maximum of 32 characters.

Service Flow ID (Unicode) A unique identifier for the service transaction flow. The format of this attribute is an alphanumeric text string with a maximum of 36 characters.

Hop Count A number that represents the relative position within a service transaction flow where the message interception occurred. The format of this attribute is an integer.

Relationship UID The unique ID for the relationship formed by the source and destination of the message. The format of this attribute is an integer.

Message Interception Time The date and time when the message was intercepted.

When this attribute is displayed in a view in the Tivoli Enterprise Portal, it is expressed as a date and time in Greenwich Mean Time (GMT). This attribute is

saved in the Tivoli Data Warehouse as a number in the format *CYYMMDDhhmmssnnn*. For example, a value for October 18 2008 at 19:45:17:010 GMT is stored as *1081018194517010*, where:

C = the century marker (for example, *1*)
YY = the year (for example, *08* for 2008)
MM = the month (for example, *10* for October)
DD = the day of the month (for example, *18* for the 18th day)
hh = the hour portion of the time (for example, *19* for 7:00 p.m.)
mm = the minutes portion of the time (for example, *45* for 45 minutes past the hour)
ss = the seconds portion of the time (for example, *17*)
nnn = the milliseconds portion of the time (for example, *010* for 10 milliseconds)

Elapsed Message Round Trip Time The elapsed round-trip time, in milliseconds, calculated at the data collector. The value is set to -1 for entry requests. The format of this attribute is an integer.

Message Length The length of the message in bytes. The format of this attribute is an integer.

Table version (Unicode) The version of this table definition. This value does not represent the version of this product. The format of this attribute is an alphanumeric text string with a maximum of 8 characters.

Message Interception Location The location where the message was intercepted (0= Client_Request, 1= Client_Response, 2= Server_Enter, 3= Server_Leave). The format of this attribute is an integer.

One Way Message This value indicates whether a message is one way (1) or two way (0). The format of this attribute is an integer.

Fault Response Indication This attribute indicates whether a message contains a fault response (1 =Yes, 0 =No). The format of this attribute is an integer.

Elapsed Time Valid This attribute indicates whether the value of the elapsed time attribute is valid. The format of this attribute is an integer.

Message Length Valid The indication whether the value of the message length attribute is valid. The format of this attribute is an integer.

Service Port Operation Mapping attributes

This attribute group maps the service port operation code to the data that is used to calculate the code. This attribute group is for internal use by ITCAM for SOA for providing support for the service-to-service topology display, and for displaying service topology in the IBM Web Services Navigator when it is configured to retrieve data from the Tivoli Data Warehouse. The attributes for this group can be configured for historical data collection. The Tivoli Enterprise Management Agent sends only the information that is new for every history request.

Origin Node The application environment instance from which the data in this table originated. The format of this attribute is an alphanumeric text string with a maximum of 32 characters.

Service Port Operation Code (Unicode) The service port operation code. The format of this attribute is an alphanumeric text string with a maximum of 36 characters.

Environment Code (Unicode) The unique code for the application server and computer system environment. The format of this attribute is an alphanumeric text string with a maximum of 36 characters.

Service Port Namespace (Unicode) The namespace used to fully qualify the service port name. The format of this attribute is an alphanumeric text string with a maximum length of 128 characters.

Service Port Name (Unicode) The name of the service port where the message was intercepted. The type of service port is specified in the Service Port Name Type column. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters.

Operation Namespace (Unicode) The namespace used to fully qualify the operation name. The format of this attribute is an alphanumeric text string with a maximum length of 128 characters.

Operation Name (Unicode) The name of the operation that is represented by the service operation code. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters.

Row Index A unique index for each row in this table. This value is an increasing number. When the table is created, the first row of the relationship table has an index value of 1. These values might not be contiguous if rows are deleted from the table. The format of this attribute is an integer.

Table version (Unicode) The version of this table definition. This value does not represent the version of this product. The format of this attribute is an alphanumeric text string with a maximum of 8 characters.

Insert Time The date and time that the row was inserted in this table, expressed in seconds. The value is expressed in elapsed seconds since January 1 1970. The format of this attribute is in Coordinated Universal Time (UTC).

Service Port Name Type The specific type of entity (for example, the Web Services Description Language (WSDL) port name or SCA component name) that identifies the service port name. The format of this attribute is an integer.

The possible values are:

- 1= WSDL_Port_Name
- 2= WSDL_Service_Name
- 3= WSDL_Service_URI
- 4= URI
- 5= SCA_Component_Name
- 7= CICS_Web_Service_Name
- 8= Message_Flow_For_SOAP
- 9=Message_Flow_For_XML
- 10=Message_Flow_For_Other
- 11=Service_Integration_Bus

Mediation Flag This attribute indicates whether the service port and operation represent a mediation (1) or not (0). The format of this attribute is an integer.

Service Type The type of service identified by the service port and operation (1=Web_Services, 2=SCA). The format of this attribute is an integer.

Operation Mode This attribute specifies the mode of the operation (1=Stand-alone Client, 2=Non-standalone Client). The format of this attribute is an integer.

Request Type The type of a query. The format of this attribute is an integer.

Request Return Code The return code for the request identified by the Request Type attribute. The format of this attribute is an integer.

Environment Mapping attributes

This attribute group is for internal use by ITCAM for SOA, providing support for displaying service-to-service topology. This attribute group also provides support for displaying service topology in the IBM Web Services Navigator when it is configured to retrieve data from the data warehouse.

The Environment Mapping attribute group contains the environment code mapping to machine information, application server environment data, and network information. The attribute group does not contain the subnode that the application server environment is associated with because there is not a one-to-one mapping in all cases. The Subnode Environment Mapping attribute group provides this mapping.

The attributes in this group can be configured for historical data collection. The Tivoli Enterprise Monitoring Agent sends only the information that is new for every historical data request.

Origin Node Identifies the Tivoli Enterprise Monitoring Agent instance from which the data in this table originated. The format of this attribute is an alphanumeric text string with a maximum of 32 characters.

Environment Code (Unicode) The unique code for the application server and computer system environment. The format of this attribute is an alphanumeric text string with a maximum of 36 characters.

Application Server Name (Unicode) The name of the application server that corresponds to the environment. The format of this attribute is an alphanumeric text string with a maximum length of 64 characters.

In the DataPower application server environment, the application server name in the monitoring metric log file identifies either the DataPower appliance host name or the name of a DataPower display group. However, because a row in this table represents a DataPower appliance domain, and a domain can be mapped to multiple DataPower display groups, this column is null for DataPower application server environments.

Port Number The port number, 0 - 65535, on which the application server environment that is being monitored is listening. If the port number value is unknown, it is set to 0. The format of this attribute is a 4-byte integer.

Application Server Node Name (Unicode) The node name of the application server. The format of this attribute is an alphanumeric text string with a maximum of 64 characters.

In the DataPower application server environment, this attribute contains the DataPower host name, if the DataPower appliance is not explicitly assigned to a display group; otherwise, it is set to null.

Application Server Cell Name (Unicode) The name of the application server cell. The format of this attribute is an alphanumeric text string with a maximum of 64 characters.

Application Server Cluster Name (Unicode) The name of the application server cluster. The format of this attribute is an alphanumeric text string with a maximum of 64 characters.

Application Server Domain Name (Unicode) The name of the DataPower appliance being monitored in the DataPower environment. The format of this attribute is an alphanumeric text string with a maximum of 80 characters.

Application Server Hostname (Unicode) The host name of the application server being monitored. The format of this attribute is an alphanumeric text string with a maximum of 64 characters.

Application Server IP Address (Unicode) The IP address of the application server being monitored. The format of this attribute is an alphanumeric text string with a maximum of 64 characters.

Local Machine ID (Unicode) The Universally Unique ID (UUID) of the computer that is being monitored. The format of this attribute is an alphanumeric text string with a maximum of 36 characters.

Agent Hostname (Unicode) The host name of the computer where the agent is running. The format of this attribute is an alphanumeric text string with a maximum of 64 characters.

Agent IP Address (Unicode) The IP address of the computer where the agent is running. The format of this attribute is an alphanumeric text string with a maximum of 64 characters.

Managed System Name The name of the managed system. The format of this attribute is an alphanumeric text string with a maximum of 64 characters.

Insert Time The date and time that the row was inserted in this table, in elapsed seconds. The value is expressed in elapsed seconds since January 1 1970 Coordinated Universal Time (UTC). The format of this attribute is a 4-byte integer.

Table version (Unicode) The version of this table definition. This value does not represent the version of this product. The format of this attribute is an alphanumeric text string with a maximum of eight characters.

Row Index A unique index for each row in this table. This value is an increasing number. When the table is created, row one of the relationship table has an index value of 1. These values might not be immediately adjacent, if rows are deleted from the table. The format of this attribute is a 4-byte integer.

DC Info (Unicode) The data collector information used to link to other workspaces that are related to this environment. This data is discovered by the data collector. The format of this attribute is an alphanumeric text string with a maximum of 64 characters.

Tivoli Enterprise Monitoring Agent Info (Unicode) The information that is related to the Tivoli Enterprise Monitoring Agent and might be used to link to workspaces that are related to this environment. This data is discovered by the agent. The format of this attribute is an alphanumeric text string with a maximum of 64 characters.

Application Server Environment The type of application server runtime environment. The format of this attribute is a 2-byte integer.

The valid values are:

- 1=WebSphere_Application_Server
- 2=.NET
- 3=WebLogic_Server
- 4=JBoss
- 5=CICS
- 6=SAP
- 7=WebSphere_Community_Edition
- 8=DataPower
- 9=Service_Component_Architecture
- 10=WebSphere_Message_Broker
- 11=WebSphere_MQ
- 12=Oracle_Application_Server
- 13=Generic_Application_Server

Subnode Environment Mapping attributes

This attribute group is for internal use by ITCAM for SOA and provides the one-to-one mapping of an application server runtime environment to a subnode.

In the DataPower environment, the application server runtime environment corresponds to a domain on a DataPower appliance. Here are the following supported relationship types:

- A DataPower application server runtime environment is mapped to multiple subnodes, if the domain is assigned to multiple DataPower display groups.
- A DataPower subnode is associated with multiple DataPower application server runtime environments, if the subnode is being used to monitor multiple DataPower appliance domains.

The attributes for this attribute group *are not* enabled for history collection.

Origin Node The Tivoli Enterprise Monitoring Agent instance from which the data in this table originated. The format of this attribute is an alphanumeric text string with a maximum of 32 characters.

Subnode The application environment subnode instance from which the data in this table originated. The format of this attribute is an alphanumeric text string with a maximum of 32 characters.

Environment Code (Unicode) The unique environment code that identifies an application server runtime environment and the computer system on which the application server is running. The format of this attribute is an alphanumeric text string with a maximum of 36 characters.

Display Group Name (Unicode) The display group name that is specified in the DataPower data collector configuration for data aggregated under this subnode. The format of this attribute is an alphanumeric text string with a maximum length of 16 characters.

Insert Time The date and time that the row was inserted in this table, expressed in elapsed seconds. The value is expressed in elapsed seconds since January 1 1970. The format of this attribute is in Coordinated Universal Time (UTC).

Table version (Unicode) The version of this table definition. This value does not represent the version of this product. The format of this attribute is an alphanumeric text string with a maximum of eight characters.

Application Server Environment The type of application server runtime environment in which the data collector is running.

The valid values for this attribute are:

- 1 = WebSphere Application Server
- 2 = .NET
- 3 = WebLogic_Server
- 4 = JBoss
- 5 = CICS
- 6 = SAP
- 7 = WebSphere_Community_Edition
- 8 = DataPower
- 9 = Service_Component_Architecture
- 10 = WebSphere_Message_Broker
- 11 = WebSphere_MQ
- 12 = Oracle_Application_Server
- 13 = Generic_Application_Server

Adapter_Binding_Metrics attributes

This table is for internal use by ITCAM for SOA for providing support for service-to-service topology display. It contains the request metrics for service-to-service relationships.

Origin Node Used to identify the application environment instance from which the data in this table originated. The valid format is an alphanumeric string, with a maximum of 32 characters.

Adapter Binding Code Binding code of the adapter for which metrics are stored. The valid format is an alphanumeric string, with a maximum of 36 characters.

Row Index A unique index for each row in this table. The valid format is a 4-byte integer.

Interval Begin Time The inclusive beginning date and time of the monitoring interval expressed as the number of seconds since January 1, 1970 UTC. The valid format is a 4-byte integer.

Interval End Time The exclusive ending date and time of the monitoring interval expressed as the number of seconds since January 1, 1970 UTC. The valid format is a 4-byte integer.

Message Count The number of messages observed during this interval. The valid format is a 4-byte integer.

Table Version (Unicode) The version of this table definition. The valid format is an alphanumeric string, with a maximum of 8 characters.

Request Type This field is used to identify the type of a query. The valid format is a 2-byte integer.

Request Return Code The request return code. The valid format is a 2-byte integer.

BPM_Dynamic_Data attributes

This table is for internal use by ITCAM for SOA for providing support for topology display. It contains BPM information of a service port operation.

Origin Node The application environment instance from which the data in this table originated. The format of this attribute is an alphanumeric text string with a maximum of 32 characters.

Mediation Primitive (Unicode) Mediation primitive code that identifies the mediation primitive of the dynamic data. The valid format is an alphanumeric string, with a maximum of 36 characters.

Mediation Primitive Instance (Unicode) Mediation primitive instance code that identifies the mediation primitive of the dynamic data. The valid format is an alphanumeric string, with a maximum of 36 characters.

External System (Unicode) External system code that identifies the external system of the dynamic data. The valid format is an alphanumeric string, with a maximum of 36 characters.

Database Table Name (Unicode) Database table name of a database lookup primitive. The valid format is an alphanumeric string, with a maximum of 128 characters.

Database Column Name (Unicode) Database column name of a database lookup primitive. The valid format is an alphanumeric string, with a maximum of 128 characters.

Registry Name (Unicode) Registry name of a WebSphere Service Registry and Repository lookup primitive. The valid format is an alphanumeric string, with a maximum of 256 characters.

First Occurrence The date and time the dynamic data occurs the first time as the number of seconds since January 1, 1970 UTC. The valid format is a 4-byte integer.

Table Version (Unicode) The version of this table definition. The valid format is an alphanumeric string, with a maximum of 8 characters.

BPM_Static_Data attributes

This table is for internal use by ITCAM for SOA for providing support for topology display. It contains BPM information of a service port operation.

Origin Node The application environment instance from which the data in this table originated. The format of this attribute is an alphanumeric text string with a maximum of 32 characters.

Environment Code (Unicode) The code for the application server and computer system environment. The valid format is an alphanumeric string, with a maximum of 36 characters.

Application Name (Unicode) Only used for an SCA component. The name of the application. The valid format is an alphanumeric string, with a maximum of 220 characters.

Event Type Only used for an SCA component. The valid format is a 2-byte integer. Valid fixed values are: Uninstall=0, Install=1, AlreadyInstall=2

Segment Number Segment number of current row. The valid format is a 2-byte integer.

Segment Count Total number of segments of the current BPM application data. The valid format is a 2-byte integer.

Insert Time The date and time the row was inserted in this table expressed as the number of seconds since Jan 1, 1970 UTC. The valid format is a 4-byte integer.

Table Version (Unicode) The version of this table definition. The valid format is an alphanumeric string, with a maximum of 8 characters.

BPM Static Data (Unicode) Only used for an SCA component. The name of the module. The valid format is an alphanumeric string, with a maximum of 1536 characters.

BPD_Activity_Events attributes

This table is for internal use by ITCAM for SOA for providing support for topology display. It contains activity events for business processes.

Origin Node A value identifying the application environment instance from which the data in this table originated. The valid format is an alphanumeric string, with a maximum of 32 characters.

Unique Key A key that uniquely identifies the business process activity event. The valid format is an alphanumeric string, with a maximum of 36 characters.

Environment Code The code for the application server where the business process was deployed. The valid format is an alphanumeric string, with a maximum of 36 characters.

Process Description Code The code for the description of the business process that contains this activity. The valid format is an alphanumeric string, with a maximum of 36 characters.

Event Nature The nature of the business process activity event. The valid format is a 2-byte integer.

Valid fixed values are:

Table 52. Values for Event Nature

Meaning	Value
Entry	1
Exit	2
Failed	3
Custom	4
Invocation	5
Failure	6
Started	7
Completed	8
Terminated	9
Deleted	10
Caught	11
Thrown	12
Expected	13
Active	14
Ready	15
Resource	16
Loop_Condition_True	17
Loop_Condition_False	18
Multiple_Instances_Started	19
Activated	20
Parallel_Instances_Started	21

Timestamp The date and time of the business process activity event expressed in seconds since January 1, 1970 UTC. The valid format is a 4-byte integer.

Process Instance Identifier A unique identifier for the business process instance. The valid format is an alphanumeric string, with a maximum of 36 characters.

Process State The execution state of the business process. The valid format is an alphanumeric string, with a maximum of 16 characters.

Current Correlator The current correlator for the business process activity event. The valid format is an alphanumeric string, with a maximum of 128 characters.

Parent Correlator The parent correlator for the business process activity event. The valid format is an alphanumeric string, with a maximum of 128 characters.

Root Correlator The root correlator for the business process activity event. The valid format is an alphanumeric string, with a maximum of 128 characters.

Activity Identifier The unique identifier for the business process activity. The valid format is an alphanumeric string, with a maximum of 64 characters.

Activity Type The business process activity type. The valid format is an alphanumeric string, with a maximum of 32 characters.

Activity Name The business process activity name. The valid format is an alphanumeric string, with a maximum of 64 characters.

Activity Instance Identifier A numeric identifier for the business process activity instance. The valid format is an alphanumeric string, with a maximum of 36 characters.

Activity Role Count The role count for the business process activity. The valid format is a 4-byte integer.

Role Identifier The role identifier for the business process activity. The valid format is an alphanumeric string, with a maximum of 32 characters.

Activity Role Resource Count The role resource count for the business process activity. The valid format is a 4-byte integer.

Role Resource Identifier The role resource identifier for the business process activity. The valid format is an alphanumeric string, with a maximum of 128 characters.

Role Resource Name The role resource name for the business process activity. The valid format is an alphanumeric string, with a maximum of 64 characters.

Discovery Mode Specifies whether the business process was discovered from static configuration files or dynamic event data. The valid format is a 2-byte integer.

Valid fixed values are:

Table 53. Values for Discovery Mode

Meaning	Value
Static	1
Dynamic	2

Maximum Requestable Rows The maximum number of rows from this table that can be returned in a single ODI request. The valid format is a 2-byte integer.

Process Identifier A key that uniquely identifies the business process. The valid format is an alphanumeric string, with a maximum of 64 characters.

Business_Process_Description attributes

This table is for internal use by ITCAM for SOA for providing support for topology display. It contains business process description data.

Origin Node A value identifying the application environment instance from which the data in this table originated. The valid format is an alphanumeric string, with a maximum of 32 characters.

Unique Key A key that uniquely identifies the business process description. The valid format is an alphanumeric string, with a maximum of 36 characters.

Process Application Identifier A key that uniquely identifies the process application. The valid format is an alphanumeric string, with a maximum of 64 characters.

Process Application Name The process application name. The valid format is an alphanumeric string, with a maximum of 64 characters.

Process Application Description The process application description. The valid format is an alphanumeric string, with a maximum of 64 characters.

Process Application Version The process application version. The valid format is an alphanumeric string, with a maximum of 64 characters.

Process Identifier A key that uniquely identifies the business process. The valid format is an alphanumeric string, with a maximum of 64 characters.

Process Name The business process name. The valid format is an alphanumeric string, with a maximum of 64 characters.

Process Description The business process description. The valid format is an alphanumeric string, with a maximum of 64 characters.

Process Version The business process version. The valid format is an alphanumeric string, with a maximum of 64 characters.

Process Namespace The business process namespace. The valid format is an alphanumeric string, with a maximum of 64 characters.

Snapshot Name The snapshot name. The valid format is an alphanumeric string, with a maximum of 64 characters.

Acronym The acronym. The valid format is an alphanumeric string, with a maximum of 8 characters.

Discovery Mode Specifies whether the business process was discovered from static configuration files or dynamic event data. The valid format is a 2-byte integer.

Valid fixed values are:

Table 54. Values for Discovery Mode

Meaning	Value
Static	1
Dynamic	2

Maximum Requestable Rows The maximum number of rows from this table that can be returned in a single ODI request. The valid format is a 2-byte integer.

BPD_Application_Environment attributes

This table is for internal use by ITCAM for SOA for providing support for topology display. It contains application environment information for BPD deployments.

Origin Node A value identifying the application environment instance from which the data in this table originated. The valid format is an alphanumeric string, with a maximum of 32 characters.

Environment Code The code for the application server and computer system environment. The valid format is an alphanumeric string, with a maximum of 36 characters.

Application Server Node Name The node name of the application server where the BPD was deployed. The valid format is an alphanumeric string, with a maximum of 64 characters.

Application Server Cell Name The name of the application server cell where the BPD was deployed. The valid format is an alphanumeric string, with a maximum of 64 characters.

Application Server Name The name of the application server where the BPD was deployed. The valid format is an alphanumeric string, with a maximum of 64 characters.

Application Server Cluster Name The name of the application server cluster where the BPD was deployed. The valid format is an alphanumeric string, with a maximum of 64 characters.

Local Hostname The hostname of the machine where the BPD was deployed. The valid format is an alphanumeric string, with a maximum of 64 characters.

Local IP Address The IP address of the machine where the BPD was deployed. The valid format is an alphanumeric string, with a maximum of 64 characters.

Local Machine ID The Universally Unique ID (UUID) of the machine where the BPD was deployed. The valid format is an alphanumeric string, with a maximum of 36 characters.

Maximum Requestable Rows The maximum number of rows from this table that can be returned in a single ODI request. The valid format is a 2-byte integer.

BPD_Gateway_Events attributes

This table is for internal use by ITCAM for SOA for providing support for topology display. It contains gateway events for business processes.

Origin Node A value identifying the application environment instance from which the data in this table originated. The valid format is an alphanumeric string, with a maximum of 32 characters.

Unique Key A key that uniquely identifies the business process gateway event. The valid format is an alphanumeric string, with a maximum of 36 characters.

Environment Code The code for the application server where the business process was deployed. The valid format is an alphanumeric string, with a maximum of 36 characters.

Process Description Code The code for the description of the business process that contains this gateway. The valid format is an alphanumeric string, with a maximum of 36 characters.

Event Nature The nature of the business process gateway event. The valid format is a 2-byte integer.

Valid fixed values are:

Table 55. Values for Event Nature

Meaning	Value
Entry	1
Exit	2
Failed	3
Custom	4
Invocation	5
Failure	6
Started	7
Completed	8
Terminated	9
Deleted	10
Caught	11
Thrown	12
Expected	13
Active	14
Ready	15
Resource	16
Loop_Condition_True	17
Loop_Condition_False	18
Multiple_Instances_Started	19
Activated	20
Parallel_Instances_Started	21

Timestamp The date and time of the business process gateway event expressed in seconds since January 1, 1970 UTC. The valid format is a 4-byte integer.

Process Instance Identifier A unique identifier for the business process instance. The valid format is an alphanumeric string, with a maximum of 36 characters.

Process State The execution state of the business process. The valid format is an alphanumeric string, with a maximum of 16 characters.

Current Correlator The current correlator for the business process gateway event. The valid format is an alphanumeric string, with a maximum of 128 characters.

Parent Correlator The parent correlator for the business process gateway event. The valid format is an alphanumeric string, with a maximum of 128 characters.

Root Correlator The root correlator for the business process gateway event. The valid format is an alphanumeric string, with a maximum of 128 characters.

Gateway Identifier The unique identifier for the business process gateway. The valid format is an alphanumeric string, with a maximum of 64 characters.

Gateway Type The business process gateway type. The valid format is an alphanumeric string, with a maximum of 32 characters.

Gateway Name The business process gateway name. The valid format is an alphanumeric string, with a maximum of 64 characters.

Gateway Instance Identifier A numeric identifier for the business process gateway instance. The valid format is an alphanumeric string, with a maximum of 36 characters.

Gateway Role Count The role count for the business process gateway. The valid format is a 4-byte integer.

Discovery Mode Specifies whether the business process was discovered from static configuration files or dynamic event data. The valid format is a 2-byte integer.

Valid fixed values are:

Table 56. Values for Discovery Mode

Meaning	Value
Static	1
Dynamic	2

Maximum Requestable Rows The maximum number of rows from this table that can be returned in a single ODI request. The valid format is a 2-byte integer.

Process Identifier A key that uniquely identifies the business process. The valid format is an alphanumeric string, with a maximum of 64 characters.

BPD_Notification_Events attributes

This table is for internal use by ITCAM for SOA for providing support for topology display. It contains notification events for business processes.

Origin Node A value identifying the application environment instance from which the data in this table originated. The valid format is an alphanumeric string, with a maximum of 32 characters.

Unique Key A key that uniquely identifies the business process notification event. The valid format is an alphanumeric string, with a maximum of 36 characters.

Environment Code The code for the application server where the business process was deployed. The valid format is an alphanumeric string, with a maximum of 36 characters.

Process Description Code The code for the description of the business process that received this notification. The valid format is an alphanumeric string, with a maximum of 36 characters.

Event Nature The nature of the business process notification event. The valid format is a 2-byte integer.

Valid fixed values are:

Table 57. Values for Event Nature

Meaning	Value
Entry	1
Exit	2
Failed	3
Custom	4
Invocation	5
Failure	6
Started	7
Completed	8
Terminated	9
Deleted	10
Caught	11
Thrown	12
Expected	13
Active	14
Ready	15
Resource	16
Loop_Condition_True	17
Loop_Condition_False	18
Multiple_Instances_Started	19
Activated	20
Parallel_Instances_Started	21

Timestamp The date and time of the business process notification event expressed in seconds since January 1, 1970 UTC. The valid format is a 4-byte integer.

Process Instance Identifier A unique identifier for the business process instance. The valid format is an alphanumeric string, with a maximum of 36 characters.

Process State The execution state of the business process. The valid format is an alphanumeric string, with a maximum of 16 characters.

Current Correlator The current correlator for the business process notification event. The valid format is an alphanumeric string, with a maximum of 128 characters.

Parent Correlator The parent correlator for the business process notification event. The valid format is an alphanumeric string, with a maximum of 128 characters.

Root Correlator The root correlator for the business process notification event. The valid format is an alphanumeric string, with a maximum of 128 characters.

Event Identifier The unique identifier for the business process notification event. The valid format is an alphanumeric string, with a maximum of 64 characters.

Event Type The business process notification event type. The valid format is an alphanumeric string, with a maximum of 32 characters.

Event Name The business process notification event name. The valid format is an alphanumeric string, with a maximum of 64 characters.

Event Instance Identifier A numeric identifier for the business process notification event. The valid format is an alphanumeric string, with a maximum of 36 characters.

Event Role Count The role count for the business process notification event. The valid format is a 4-byte integer.

Discovery Mode Specifies whether the business process was discovered from static configuration files or dynamic event data. The valid format is a 2-byte integer.

Valid fixed values are:

Table 58. Values for Discovery Mode

Meaning	Value
Static	1
Dynamic	2

Maximum Requestable Rows The maximum number of rows from this table that can be returned in a single ODI request. The valid format is a 2-byte integer.

Process Identifier A key that uniquely identifies the business process. The valid format is an alphanumeric string, with a maximum of 64 characters.

BPD_Process_Execution_Events attributes

This table is for internal use by ITCAM for SOA for providing support for topology display. It contains execution events for business processes.

Origin Node A value identifying the application environment instance from which the data in this table originated. The valid format is an alphanumeric string, with a maximum of 32 characters.

Unique Key A key that uniquely identifies the business process execution event. The valid format is an alphanumeric string, with a maximum of 36 characters.

Environment Code The code for the application server where the business process was deployed. The valid format is an alphanumeric string, with a maximum of 36 characters.

Process Description Code The code for the description of the business process which was executed. The valid format is an alphanumeric string, with a maximum of 36 characters.

Event Nature The nature of the business process execution event. The valid format is a 2-byte integer.

Valid fixed values are:

Table 59. Values for Event Nature

Meaning	Value
Entry	1
Exit	2
Failed	3
Custom	4
Invocation	5
Failure	6
Started	7
Completed	8
Terminated	9
Deleted	10
Caught	11
Thrown	12
Expected	13
Active	14
Ready	15
Resource	16
Loop_Condition_True	17
Loop_Condition_False	18
Multiple_Instances_Started	19
Activated	20
Parallel_Instances_Started	21

Timestamp The date and time of the business process execution event expressed in seconds since January 1, 1970 UTC. The valid format is a 4-byte integer.

Process Instance Identifier A unique identifier for the business process instance. The valid format is an alphanumeric string, with a maximum of 36 characters.

Process State The execution state of the business process. The valid format is an alphanumeric string, with a maximum of 16 characters.

Current Correlator The current correlator for the business process execution event. The valid format is an alphanumeric string, with a maximum of 128 characters.

Parent Correlator The parent correlator for the business process execution event. The valid format is an alphanumeric string, with a maximum of 128 characters.

Root Correlator The root correlator for the business process execution event. The valid format is an alphanumeric string, with a maximum of 128 characters.

Fault Code The code for the fault (if any) that happened during the business process execution event. This code is the UNIQUEKEY value in the table "BPD_Faults attributes." The valid format is an alphanumeric string, with a maximum of 36 characters.

From Service Operation Code The service port operation code for the caller of the business process. The valid format is an alphanumeric string, with a maximum of 36 characters.

Discovery Mode Specifies whether the business process was discovered from static configuration files or dynamic event data. The valid format is a 2-byte integer.

Valid fixed values are:

Table 60. Values for Discovery Mode

Meaning	Value
Static	1
Dynamic	2

Maximum Requestable Rows The maximum number of rows from this table that can be returned in a single ODI request. The valid format is a 2-byte integer.

Process Identifier A key that uniquely identifies the business process. The valid format is an alphanumeric string, with a maximum of 64 characters.

BPD_Faults attributes

This table is for internal use by ITCAM for SOA for providing support for topology display. It contains fault messages reported by business processes.

Origin Node The application environment instance from which the data in this table originated. The format of this attribute is an alphanumeric text string with a maximum of 32 characters.

Unique Key A key that uniquely identifies the business process fault. The valid format is an alphanumeric string, with a maximum of 36 characters.

Fault Name The name of the fault reported by the business process. The valid format is an alphanumeric string, with a maximum of 512 characters.

Fault Message The error message for the fault reported by the business process. The valid format is an alphanumeric string, with a maximum of 2048 characters.

Discovery Mode Specifies whether the reported fault was discovered from static configuration files or dynamic event data. The valid format is a 2-byte integer. Valid values are: Static=1, Dynamic=2.

Maximum Requestable Rows The maximum number of rows from this table that can be returned in a single ODI request. The valid format is a 2-byte integer.

Business_Process_Situation_Data attributes

This table is for internal use by ITCAM for SOA for providing support for Tivoli Monitoring situations.

Origin Node The application environment instance from which the data in this table originated. The format of this attribute is an alphanumeric text string with a maximum of 32 characters.

Data Collector Node Used to identify the data collector instance from which the data in this table originated. The valid format is an alphanumeric string, with a maximum of 32 characters.

Environment Code The code for the application server where the business process was deployed. The valid format is an alphanumeric string, with a maximum of 36 characters.

Unique Key A key that uniquely identifies the event. The valid format is an alphanumeric string, with a maximum of 36 characters.

Event Type The type of the business process event. The valid format is a 2-byte integer. Valid fixed values are: Process=51, Activity=52, Gateway=53, Notifier=54

Event Nature The nature of the business process event. The valid format is a 2-byte integer.

The available values are:

- 1 = ENTRY
- 2 = EXIT
- 3 = FAILED
- 4 = CUSTOM
- 5 = INVOCATION
- 6 = FAILURE
- 7 = STARTED
- 8 = COMPLETED
- 9 = TERMINATED
- 10 = DELETED
- 11 = CAUGHT
- 12 = THROWN
- 13 = EXPECTED
- 14 = ACTIVE
- 15 = READY
- 16 = RESOURCE
- 17 = LOOP_CONDITION_TRUE
- 18 = LOOP_CONDITION_FALSE
- 19 = MULTIPLE_INSTANCES_STARTED
- 20 = ACTIVATED
- 21 = PARALLEL_INSTANCES_STARTED

Event Time The date and time when the event occurred. The valid format is a 16-character timestamp.

Process Application Name The process application name for the event. The valid format is an alphanumeric string, with a maximum of 64 characters.

Process Application Version The process application version for the event. The valid format is an alphanumeric string, with a maximum of 128 characters.

Process Name The business process name for the event. The valid format is an alphanumeric string, with a maximum of 64 characters.

Item Name The name of the activity, notifier, or gateway (if any) for the event. The valid format is an alphanumeric string, with a maximum of 64 characters.

Fault Name The name of the fault (if any) reported by the event. The valid format is an alphanumeric string, with a maximum of 512 characters.

Fault Message The error message for the fault reported by the business process. The valid format is an alphanumeric string, with a maximum of 2048 characters.

Discovery Mode Specifies whether the business process was discovered from static configuration files or dynamic event data. The valid format is a 2-byte integer. Valid values are: Static=1, Dynamic=2.

Maximum Requestable Rows The maximum number of rows from this table that can be returned in a single ODI request. The valid format is a 2-byte integer.

Chapter 13. Workflow policies

Use the Tivoli Enterprise Portal to automate a situation by including a Take Action command that runs when the situation becomes true. Tivoli Enterprise Portal provides a *workflow editor* for designing and managing policies. A *policy* is a collection of activities that you can assemble to automate responses to events or routine operator tasks.

This method combines complex automated system processes with operator intervention and decision-making capability, and has the following advantages:

- You can implement more complex workflow strategies than you can create by adding Take Action commands to a situation.
- You can monitor multiple conditions simultaneously on any number of systems.
- You can have selected activities take place when conditions occur.
- You can notify appropriate user groups and offer them activity choices at specified points in the automated process.

After an activity completes, Tivoli Enterprise Portal receives feedback by way of return codes. Advanced automation logic responds with subsequent activities depending on the value of the return codes.

This example illustrates how workflows can be built based on the message flow. By default, this workflow is defined to reject messages to all combinations of service port name and operation name. Modify this workflow before you use it, to customize it for specific combinations of service port name and operation name. For more information about using the workflow editor to create your own workflows, see the IBM Tivoli Monitoring documentation and online help.

Creating policies

Policies are created using the Workflow Editor. Before you create a policy, decide what you want it to do and where you want it to operate. Then, define the situation that triggers the first action. When you create a policy, assign it to one or more managed systems in your network. For additional information about creating and modifying policies, see the Tivoli Enterprise Portal online help.

Predefined policies

ITCAM for SOA provides one predefined policy that you can use as designed, or you can modify for your environment.

The MessageArrival_610 policy

The MessageArrival_610 policy is a predefined policy that is based on the MessageArrivalCritical_610 and MessageArrivalClearing_610 situations. When the MessageArrivalCritical_610 situation becomes true, the AddFltrCntrl_610 Take Action command is automatically run to create a filter control. The filter control rejects messages based on the situation values. This sample workflow is shown in the Workflow editor in Figure 80 on page 326

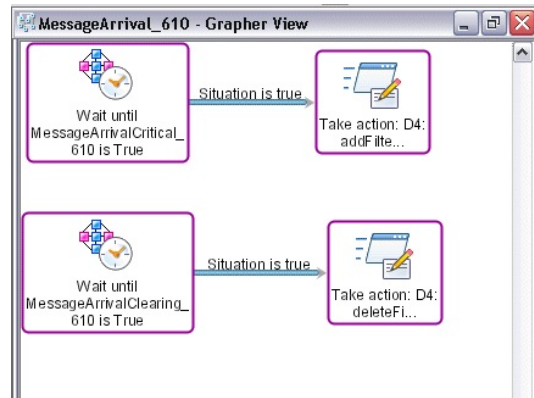


Figure 80. The sample MessageArrival_610 workflow

When the MessageArrivalClearing_610 situation becomes true, the filter control that is created by the MessageArrivalCritical_610 situation is deleted from the system by automatically running the DelFtrCntrl_610 Take Action command.

Important: This policy is not enabled by default. You must update this policy before you use it. In its initial form, the policy uses the default MessageArrivalCritical_610 and MessageArrivalClearing_610 situations, which are initially designed to observe message traffic from *all* service and operation combinations. Typically, you observe traffic from only a certain combination of services and operations. In addition, the same set of service and operation combinations are used as input parameters to the Take Actions that result from the triggering of these situations, so that messages from all combinations of services and operations are rejected. Be sure to configure both the situations and the Take Action commands that are used in this policy before distributing it for general use. You must first distribute the policy to the managed systems on which you want it to run, then start the policy on those managed systems.

For details on using and editing policies, see the online help for Tivoli Enterprise Portal.

Appendix A. Integrating with other products

Integration with IBM Tivoli License Compliance Manager

IBM Tivoli License Compliance Manager version 2.1 or later can be used to complete inventory and usage tracking of the ITCAM for SOA monitoring agent.

Integration with IBM Tivoli Business Service Manager

ITCAM for SOA provides integration with Tivoli Business Service Manager version 4.1 and later, through both discovery and event support.

ITCAM for SOA provides Discovery Library Adapters that discover service definitions in WebSphere Service Registry and Repository, service deployments (which application servers services are deployed on), and business processes defined using the Business Process Execution Language and the web services that are started by the business processes. The discovery library adapters create books that contain the discovered service data. For more details, see the *IBM Tivoli Composite Application Manager for SOA Discovery Library Adapters* guide. These books can be imported into Tivoli Business Service Manager or into the Tivoli Application Dependency Discovery Manager to which Tivoli Business Service Manager connects. Tivoli Business Service Manager can then map the discovered service data to templates that you use to create business service instances.

Situation events for services discovered by ITCAM for SOA monitoring agents can also be forwarded to Tivoli Business Service Manager to update the status of business service instances created from the discovered service data.

Through the use of this integration, the users visualizing the services in Tivoli Business Service Manager can then link back into the Tivoli Enterprise Portal console to view the data within ITCAM for SOA for more detailed examination and analysis.

For more information about how ITCAM for SOA integrates with IBM Tivoli Business Service Manager, see the following web link: <http://www-01.ibm.com/software/brandcatalog/portal/opal/details?NavCode=1TW10BM03>

Integration using Workspace Links

IBM Tivoli Composite Application Manager for SOA can integrate with other products through workspace links. For details about using the list of predefined workspace links that are provided with this release of ITCAM for SOA, see “Accessing workspaces by using links” on page 27.

Integration with IBM WebSphere Service Registry and Repository

IBM WebSphere Service Registry and Repository is a system for storing, accessing, and managing information, referred to as service metadata. WebSphere Service Registry and Repository is where you store information about services in your systems, or in systems for other organizations, that you already use, that you plan to use, or of which you want to be aware.

ITCAM for SOA can obtain the list of registered services from WebSphere Service Registry and Repository and display them in the topology views that are described in Chapter 6, “Workspace for service registry integration,” on page 53.

For more details on how ITCAM for SOA obtains the data from WebSphere Service Registry and Repository, see the *IBM Tivoli Composite Application Manager for SOA Discovery Library Adapters* guide.

ITCAM for SOA can also forward events to WebSphere Service Registry and Repository to update metadata properties for service ports affected by the event. For more details, see the *IBM Tivoli Composite Application Manager for SOA WSRR Integration Guide*.

Integration with IBM WebSphere Business Modeler

IBM WebSphere Business Modeler helps you visualize, understand, and document your business processes. ITCAM for SOA can discover business processes that are defined using IBM WebSphere Business Modeler and show which web service operations are used by the business processes. For more information, see these places in this User's Guide:

- Chapter 7, “Workspaces for service-to-service topology,” on page 71
- Chapter 6, “Workspace for service registry integration,” on page 53

For more information about the Business Process Execution Language Discovery Library Adapter, see the *IBM Tivoli Composite Application Manager for SOA Discovery Library Adapters* guide.

If you define your business processes using IBM WebSphere Integration Developer instead of IBM WebSphere Business Modeler, the same integration capabilities are provided.

Integration with Tivoli Change and Configuration Management Database and Tivoli Application Dependency Discovery Manager

IBM Tivoli Change and Configuration Management Database is an integrated productivity tool and database that helps you manage, audit, and coordinate the change and configuration management processes using user interfaces and workflows. It includes a Configuration Management Database that stores information about configurations items for use in configuration management processes.

IBM Tivoli Application Dependency Discovery Manager is an application mapping and discovery tool that automatically gathers an inventory of all applications and dependencies, helps you to understand configurations and helps to prove compliance, with detailed reports and auditing tools. It also includes a Configuration Management Database that stores information about configurations items.

The discovery library adapters provided with ITCAM for SOA create books that define configuration items for service definitions, deployed services, and business processes. These books can be imported into these versions of Configuration Management Database of CCMDB and TADDM:

- Change and Configuration Management Database version 1.1.1 fix pack 3 or later fix pack

- Change and Configuration Management Database version 7.1
- Tivoli Application Dependency Discovery Manager version 5.1.1 fix pack 3 or later fix pack
- Tivoli Application Dependency Discovery Manager version 7.1 and version 7.1.2 and later fix packs
- Tivoli Application Dependency Discovery Manager version 7.2 and later fix packs

Appendix B. Determining status for operation instances, operation aggregates, and groups

This version of IBM Tivoli Composite Application Manager for SOA introduces the concept of a *group* to represent a set of related services that collectively represent some business function. Because understanding status is important for managing a business function, this appendix describes the model for how the status of operation instances, operation aggregates, and groups are determined and displayed.

The status model adheres to the following key characteristics:

- The model is consistent with the way IBM Tivoli Monitoring version 6.2 represents the status of open situations in Tivoli Enterprise Portal workspaces and views.
- It supports the ability to assess status at all levels of the SOA, from operation instances to the operation aggregates and on up to groups.
- The status model reflects, where applicable, the structure of service flows and can characterize the directed call relationships among a pair of services, as well as more complex flows.
- At higher levels of the SOA, for example, groups whose operation aggregate members might represent a wide variety of different status values, the tendency is to bias the overall group status towards a more *pessimistic* state, to more readily call your attention to problems across the entire managed SOA.
- The status model does not require you to configure or maintain it. The model uses a set of predefined rules and algorithms to determine the overall status at each level of the SOA.

Status values

The basis for the overall status model is the state of situations that are associated with specific operation instances. For IBM Tivoli Monitoring version 6.2, the various levels of status for situations is shown in Table 61.

Table 61. Possible states for situations in IBM Tivoli Monitoring version 6.2








Symbol	State
	Fatal
	Critical
	Minor
	Warning
	Harmless
	Informational

Table 61. Possible states for situations in IBM Tivoli Monitoring version 6.2 (continued)

Symbol	State
	Unknown
(No state symbol is used when there are no situations opened.)	(Normal)

Enabling status to fit your needs: To obtain the most effective status representation in your Tivoli Enterprise Portal workspaces and views, you must ensure that situations are defined and deployed in way that makes the most sense for your SOA. At higher levels, create and maintain your groups in a way that also makes the most sense for your SOA. To the extent that both the situations and the groups are defined correctly for your environment, the resulting status must also be correct and representative.

Determining status for operation instances

Operation instances continue to reflect the status of the situations associated with them. The status that is reported for an operation instance continues to be the most severe status condition among the set of open situations that are associated with that operation instance.

The various status levels that an operation instance can have is reflected in Table 61 on page 331. If the operation instance does not have any open situations associated with it, then the status for the operation instance is *Normal*, and no status icon is displayed with the operation instance.

Offline: If the monitoring agent for an operation instance is offline, the operation instance is displayed as *offline* in the Interaction Detail portion of the Operation Flow view.

If the subnode for an operation instance is offline, no status icon is displayed for the instance in the Interaction Detail portion of the view. This is because the list of situations that were open before the subnode went offline are no longer available, and their current state is not known.

Unmanaged: Status indicators are not displayed for operation instances that are unmanaged clients or unmanaged operations. Their implied status is *Normal*.

Determining status for operation aggregates

For IBM Tivoli Composite Application Manager for SOA version 7.1.0, operation aggregates displayed a purple diamond shaped *Abnormal* status icon, similar to the following example:















In this context, this icon indicated that one or more of its associated operation instances had at least one open situation.

For this release, the indication of aggregate status is further refined. In addition to the purple diamond shaped status icon, the summarized status for operation

instances at the operation aggregate level are represented by a condensed version of the status indicators available for operation instances, as shown in Table 62.

Table 62. Mapping of operation instance status to operation aggregate level status

Operation instance level status		Operation aggregate level status	
	Fatal		Fatal
	Critical		Critical
	Minor		Warning
	Warning		
	Harmless	(No state symbol is used when there are no situations opened)	(Normal) If the status for an operation aggregate is assumed to be <i>Normal</i> for purposes of calculating the group status, but there is at least one operation instance with an open situation or an offline subnode, the purple diamond icon is displayed in the service-to-service topology views, similar to the following example:  This icon indicates an <i>Abnormal</i> status for the operation aggregate.
	Informational		
(No state symbol is used when there are no situations opened)	(Normal)		
	Unknown		Unknown

Offline: Operation instances whose monitoring agents are offline are classified as *Unknown* for the purposes of determining the operation aggregate status. Any situation status that is associated with these operation instances is ignored.

Unmanaged: Status indicators are not displayed for unmanaged client aggregates or unmanaged operation aggregates.

Rules for calculating operation aggregate status

The rules to calculate the operation aggregate status focus on the following concepts:

Plurality

The largest quantity of instances associated with a status (with ties going to the most severe status).

Majority

A simple majority, or more than half, of the number of operation instances in an aggregate that have a status.

Pessimistic bias

When attempting to characterize the status of operation aggregates and groups, the correct tendency is toward assigning the more serious, or critical, state. This tendency is referred to as the *pessimistic bias*. As the operation aggregate or group is being evaluated, with each of the various status levels being counted, this bias is a way of determining whether to use the status level as the aggregate status, or to lean toward a more serious status level.

The algorithm used to determine the aggregate status level performs these general steps:

1. The status for each operation instance is mapped to its equivalent operation aggregate status (as described in Table 62 on page 333).
2. The number of operation instances at each unique aggregate level status is counted (that is, the number of instances with *Unknown* state, the number with *Warning* state, the number of instances with *Critical* state, and so on).
3. A preliminary evaluation is performed on the total number of operation instances at each level of status. If more than half of all of the operation instances have a mapped status level of *Warning*, *Critical*, or *Fatal*, then the initial status for the aggregate is set to *Warning*, otherwise the initial status for the aggregate is set to *Unknown*.
4. A simple evaluation is made on the total count of operation instances at each unique status level, starting from the initial aggregate status (*Warning* or *Unknown* as determined in the previous step) through the more severe status levels up to *Fatal*, taking into account the pessimistic bias factor. This result determines whether to keep the current aggregate status or to move up to the next most severe status.
5. In certain circumstances, it might be acceptable to characterize the overall status of the aggregate as *Normal*, even if at least one operation instance in the aggregate has a non-normal status. For example, you might have an operation aggregate consisting of 100 operation instances, but only one has a status of *Warning*, while the rest have normal status. In this case, you can tolerate a small percentage of operation instances at a non-normal status level, and consider the overall aggregate status as *Normal* when calculating the overall status for a group. In service-to-service topology views, however, this operation aggregate is still displayed with the purple diamond status icon, indicating an *Abnormal* status associated with at least one of the operation instances for that aggregate.

The following examples illustrate how aggregate status is calculated:

- **Example 1:** An operation aggregate has nine operation instances. One instance has an open *Fatal* situation, two instances have a status of *Critical*, two instances have a status of *Minor*, one instance has a status of *Informational*, two instances have a status of *Unknown*, and the remaining instance is *Normal*.
 1. Using the mapping that is described in Table 62 on page 333, each operation instance status is mapped to its equivalent operation aggregate status:
 - The *Fatal* operation instance status is mapped to its equivalent operation aggregate status of *Fatal*.
 - The *Critical* operation instance status is mapped to its equivalent operation aggregate status of *Critical*.

- The *Minor* operation instance status is mapped to its equivalent operation aggregate status of *Warning*.
 - The *Informational* operation instance status is mapped to its equivalent operation aggregate status of *Normal*.
 - The *Unknown* operation instance status is mapped to its equivalent operation aggregate status of *Unknown*.
 - The remaining operation instance with a *Normal* status is mapped to its equivalent operation aggregate status of *Normal*.
2. The number of operation instances at each unique (mapped) status level are counted:
 - Fatal: 1
 - Critical: 2
 - Warning: 2
 - Unknown: 2
 - Normal: 2
 3. A preliminary evaluation of these totals displays that 5 out of 9 operation instances are at one of the more severe (*Warning*, *Critical*, or *Fatal*) status levels. Because at least one half of the operating instances are at one of these severe status levels, the initial status level for the operation aggregate is set to *Warning*.
 4. Starting with the *Unknown* status level and progressing through each more severe status level up to *Fatal*, the evaluation of pessimistic bias is applied. As a result, the initial aggregate status of *Warning* is eventually increased to *Fatal*.
 5. Finally, in this example, enough operation instances have a non-normal status that the algorithm cannot consider the overall aggregate status to be *Abnormal*. This operation aggregate has a significant number of situations at one of the more severe status levels, so the algorithm characterizes the overall status of this operation aggregate as *Fatal*.

For this example, the operation aggregate is displayed in service-to-service topology views with the *Fatal* status indicator. If this operation aggregate is also part of a group, its status is considered to be *Fatal* when the overall status of the group is calculated.

- **Example 2:** An operation aggregate has 10 operation instances. One instance has a status of *Critical*, and a second instance has a status of *Minor*, because of open situations associated with each operation instance. Assume that the remaining operation instances have a status of *Normal*.
 1. Using the mapping that is described in Table 62 on page 333, each operation instance status is mapped to its equivalent operation aggregate status:
 - The *Critical* operation instance status is mapped to its equivalent operation aggregate status of *Critical*.
 - The *Minor* operation instance status is mapped to its equivalent operation aggregate status of *Warning*.
 - The remaining eight operation instances with a *Normal* status are mapped to their equivalent operation aggregate status of *Normal*.
 2. The number of operation instances at each unique (mapped) status level are counted:
 - Fatal: 0
 - Critical: 1
 - Warning: 1

- Unknown: 0
 - Normal: 8
3. A preliminary evaluation of these totals displays that 2 out of 10 operation instances are at one of the more severe (*Warning*, *Critical*, or *Fatal*) status levels. Because less than one half of the operating instances are at one of these severe status levels, the initial status level for the operation aggregate is set to *Unknown*.
 4. Starting with the *Unknown* status level and progressing through each more severe status level up to *Fatal*, the evaluation of pessimistic bias is applied. As a result, the initial aggregate status of *Unknown* is increased to *Critical*.
 5. Finally, even though two of the operation instances are not at a normal status, a large majority of the instances are normal, indicating that most of the operations associated with this aggregate are functioning properly. Because only a small percentage of the operation instances is not in a normal state, the algorithm characterizes this overall status as merely *Abnormal*, instead of keeping the more severe *Critical* status.

For this example, the operation aggregate is displayed in service-to-service topology views with the purple diamond *Abnormal* status indicator. If this operation aggregate is also part of a group, however, its status is considered to be *Normal* when the overall status of the group is calculated.

Determining status for dependent pairs

In previous sections, the status model can determine the status of individual operation instances and an overall status for an operation aggregate.

At higher levels of the SOA, such as at the group level, status is determined not only by groups of operation aggregates, but also by service flows, or parts of service flows, that are included in the group. Before you understand how to determine the overall status for a group, you must understand how the status is determined for dependent pairs of operation aggregates and the relationships between them. When that is understood, then this dependent pair status can be used to determine the status of more complex composite relationships across a service flow.

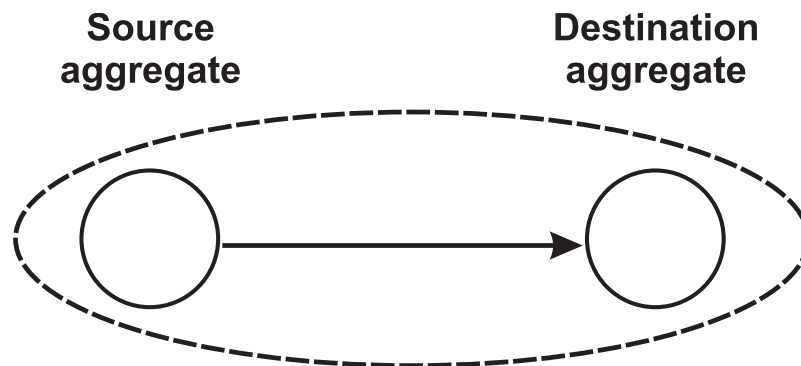


Figure 81. A dependent pair of aggregates with a directed relationship

Given a directional relationship between a pair of operation aggregates, shown graphically in Figure 81, the derived status is to be a function of the status of the source and the status of the destination. The status for the pair of aggregates must also reflect the fact that the status of the source has an implicitly greater weight on the overall status. For example, if the source operation aggregate has a status of

Normal but the status of the destination operation aggregate is *Fatal*, then the combined status, although not *Normal*, is not *Fatal*, because the client service seems to be functioning normally based on its situation status.

The algorithm that is used to determine the status of a dependent pair does these general steps:

1. Set the initial status of the dependent pair equal to the status of the source aggregate.
2. Determine if the status of the destination aggregate is more severe than the status of the source aggregate.
3. If the status of the destination aggregate is more severe than the status of the source aggregate, then increment the status of the dependent pair to the next more severe status level (*Normal*, *Warning*, *Critical*, *Fatal*, skipping the *Unknown* status).

Here are some examples of evaluating the status for a dependent pair of aggregates:

Table 63. Examples of status for a dependent pair of aggregates

Source aggregate status	Destination aggregate status	Dependent pair status
Normal	Normal	Normal
Normal	Critical	Warning
Warning	Normal	Warning
Normal	Fatal	Warning
Critical	Fatal	Fatal

Determining status for composite relationships

The previous section described the basic method for determining the combined status for a dependent pair of operation aggregates. Building on this technique, this section describes how to determine the overall status of a service flow.

Start at the end of the service flow, iterating backwards through the flow toward the beginning of the flow, and cascading the combined status of each dependent pair into the next preceding pair.

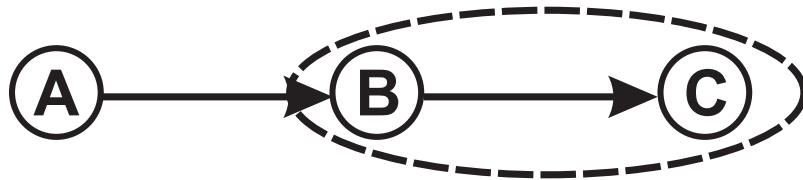


Figure 82. Determining the overall status for a simple service flow

The status is determined by first determining the overall status of the rightmost pair (aggregates *B* and *C*), shown enveloped in the dashed-line circle in Figure 82), and then using that resulting status as the new aggregate destination status for computing the overall status of the leftmost pair (aggregate *A* and the combined status of *BC*).

For example, suppose that in Figure 82, aggregate *A* has an overall status of *Normal*, aggregate *B* has a status of *Warning*, and aggregate *C* has a status of *Fatal*.

First, the status for the *B* and *C* dependent pair are determined:

- The initial status of *BC* is set to the same status as *B*, or *Warning*.
- The status for the destination aggregate *C* is more severe than the status for *B*, so the overall status for *BC* is incremented to the next higher state, or *Critical*.

Now, the status for the dependent pair, source *A* and destination *BC* is determined:

- The initial status of *ABC* is set to the same status as *A*, or *Normal*.
- The status for the destination aggregate *BC* is *Critical*, and is more severe than the status for *A*, so the overall status for *ABC* is incremented to the next higher state, or *Warning*.

This same technique also applies to other varieties of composite relationships. For example, Figure 83 displays a single source aggregate that has two parallel relationships with two different destination aggregates.

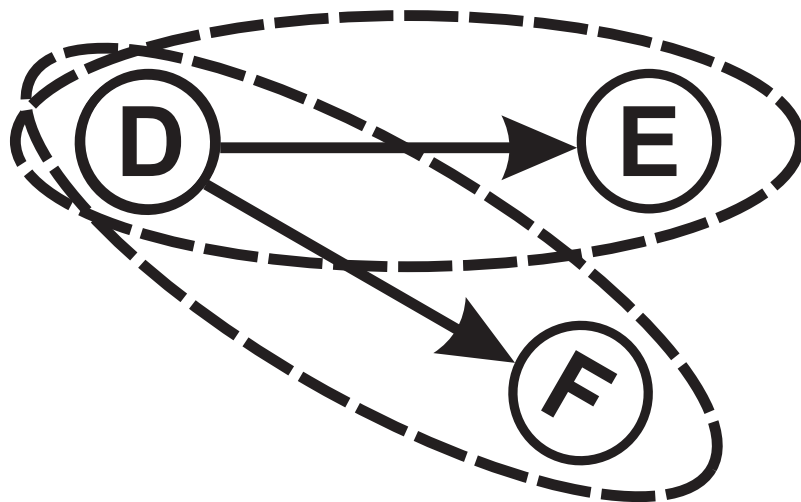


Figure 83. A dependent pair of aggregates with a directed relationship

The status of each dependent pair (*DE* and *DF*) is determined separately using the technique described in the previous section. This results in two different status values, one for each dependent pair. These two values of status are then summarized into an overall composite status using the same technique that is used to determine operation aggregate status with multiple operation instances, described in “Rules for calculating operation aggregate status” on page 333.

For example, suppose that in Figure 83, aggregate *D* has an overall status of *Normal*, aggregate *E* has a status of *Warning*, and aggregate *F* has a status of *Fatal*.

First, the status for the *D* and *E* dependent pair are determined:

- The initial status of *DE* is set to the same status as *D*, or *Normal*.
- The status for the destination aggregate *E* is more severe than the status for *D*, so the overall status for *DE* is incremented to the next higher state, or *Warning*.

Now the status for the dependent pair, source *D* and destination *F*, is determined:

- The initial status of *DF* is set to the same status as *D*, or *Normal*.
- The status for the destination aggregate *F* is *Fatal*, and is more severe than the status for *D*. Therefore, the overall status for *DF* is incremented to the next higher state, or *Warning*.

Finally, the *Warning* status for *DE* and the *Warning* status for *DF* are summarized to an overall status of *Warning* for *DEF*, using the rules for calculating operation aggregate status.

This strategy applies regardless of the number of destination aggregates the source has a relationship with. Similarly, this technique also works if the service flow is reversed, with multiple clients calling into a single common service.

Front-end service: The overall algorithm determines the status of the *front-end service* for a group by analyzing the full path of the flow, starting from the front-end service and following all call paths that go through other members of the group. The resulting status is valid whether or not the *front end* for a group is really the front end for a particular service flow.

Cyclic flows

It is possible to have cycles or loops occurring in service flows. In this situation, the status of the cycle is determined by applying the procedure that is described in “Rules for calculating operation aggregate status” on page 333 to the aggregates in the cycle, disregarding any relationships among them. Then, the overall status for the cycle is considered, along with the status of the parts of the service flow that occur before and after the cycle, to determine the status for the entire service flow.

Figure 84 displays a simple example of a cyclic flow. The cycle occurs where *B* calls *C* and *C*, in turn, calls *B*.



Figure 84. An example of a service flow with a cyclic flow

To illustrate how the overall status for this flow is determined, consider the following example:

1. Suppose these aggregates each have the following aggregate level status:
 - The status for *A* is *Normal*.
 - The status for *B* is *Warning*.
 - The status for *C* is *Critical*.
 - The status for *D* is *Fatal*.
2. Starting at the end of the flow, the status for the rightmost dependent pair, *CD*, is determined to be *Fatal*.
3. Using the rules for calculating operation aggregate status, and ignoring the cyclical directed relationships between *C* and *D*, the combined status for *B* and *CD* is determined to be *Fatal*.
4. Finally, the status for the dependent pair *A* and *BCD* is determined to be *Warning*.

It is possible for a cyclic flow to occur at the beginning of the service flow. In this case, there is no clear front-end service. However, the algorithm for determining status within a cycle still applies. Using the example flow in Figure 84, assume *B*, *C*, and *D* are in the service group but *A* is not. The status of the composite relationships for *B*, *C*, and *D* are determined as follows:

1. Determine the status of the dependent pair, *CD*.
2. Determine the combined status of *B* and *CD*.

Determining status for groups

Groups are collections of operation aggregates that you define. They can encompass one or more service flows, or subsets of flows, or they can be any selection of individual operation aggregates. Because the status of these collections of operation aggregates can be determined using the methods described in the previous sections, the overall status for a group is determined using the same set of rules and procedures.

The approach is to treat the group as a collection of discrete elements:

- Operation aggregates that have no relationships to other aggregates in the group. Each of these operation aggregates is considered a front-end service.
- Service flows with one or more front-end services.
- Subsets of service flows (that is, a specific set of aggregates, implicitly reflecting the relationships among them) with one or more front-end services.

The previous sections describe how to determine the overall status for each of these different kinds of elements that might comprise a group. Summarizing the status of each of these elements into an overall group status is accomplished by applying the same procedures:

1. For each front-end service, determine the status using *either* of the following methods:
 - Use the operation aggregate status for the front-end service if it does not have a relationship to other operation aggregates in the group.
 - Use the composite relationship status for the flow starting at the front-end service.
2. If a group has more than one front-end service, then apply the procedure that is described in “Rules for calculating operation aggregate status” on page 333 using the status of each front-end service determined in step 1.

For groups this results in one of the same status values used for operation aggregates (see Table 62 on page 333).

Empty group: Groups are initially empty when first created, and a particular group might become empty again later as you remove operation aggregates using the Groups page. It is also possible for a group to become empty again if all of the operation instances in all of the operation aggregates in a group are deleted. In any case, an empty group has a status of *Normal*.

Appendix C. Platform tuning

The following sections describe several ways that you can improve the performance of your IBM Tivoli Monitoring environment when working with ITCAM for SOA.

Configuring Monitoring Intervals

If you are using ITCAM for SOA in a non-production environment, such as a demonstration or a proof-of concept, you might find it helpful to not have to wait for the full 5-minute monitoring time interval before you can display updated metrics in ITCAM for SOA workspaces, including the service-to-service topology workspaces.

You can reduce the monitoring interval that is used by the ITCAM for SOA monitoring agent to a value less than the default of 5 minutes by modifying the value of the `kd4.ira.aggDataWindowSizeMinutes` property in the `<ITCAM4SOA_Home>/KD4/config/KD4.dc.properties` file.

Important: You can also set the monitoring interval longer than 5 minutes, if desired.

The monitoring agent uses the `kd4.ira.aggDataWindowSizeMinutes` property to control the monitoring interval for these tables:

- Services Inventory table (KD42IT)
- Services Inventory for Requester ID table (KD42JT)
- Relationship Request Metrics table (KD43RQ)
- Relationship Response Metrics tables (KD43RP)

Performance Impact: Reducing the monitoring interval for the ITCAM for SOA monitoring agent below the default value can affect performance, because of the increased frequency of retrieving data, and the additional memory required to store the collected metrics. Modify this interval only for nonproduction environments. For production environments, do not use a monitoring interval less than the default value of 5 minutes.

Make the same change on all servers: If you change the monitoring interval, make the same change on each server where the ITCAM for SOA monitoring agent is installed, and then restart the monitoring agent for the change to take effect.

Change the SOA Domain Management Server interval too: If you change the monitoring interval for the ITCAM for SOA agent, make the same change for the SOA Domain Management Server interval. For more information, see “SOA Domain Management Server considerations” on page 342.

To modify the value of the `kd4.ira.aggDataWindowSizeMinutes` property on each server where the monitoring agent is installed, complete these steps:

1. Navigate to the `<ITCAM4SOA_Home>/KD4/config` directory and locate the `KD4.dc.properties` file. For information about the value of `<ITCAM4SOA_Home>`, see “Operating system-dependent variables and paths” on page xi.

2. Edit the `KD4.dc.properties` file using your preferred text editor.
3. In the `KD4.dc.properties` file, locate the following line:
`kd4.ira.aggDataWindowSizeMinutes=5`
4. Change the value of this property to another valid value. You can set the value of this property to any of these valid values, expressed in minutes: 60, 30, 20, 15, 10, 5 (the default value), 4, 3, 2, or 1.
 Keep in mind these conditions:
 - If the value for the `kd4.ira.aggDataWindowSizeMinutes` property is missing, the default value of 5 is used.
 - If you specify a value other than one of the valid values listed above, the monitoring interval is *rounded down* to the closest valid value.
 - If you specify a value less than 1 minute, the monitoring interval used is rounded up to 1.
5. Save your changes to the `KD4.dc.properties` file.
6. Restart the monitoring agent for the change to take effect.

Repeat these steps for *all servers* where the ITCAM for SOA monitoring agent is installed. All ITCAM for SOA monitoring agents must be configured to use the *same value* for the `kd4.ira.aggDataWindowSizeMinutes` property.

SOA Domain Management Server considerations

If you change the value of the monitoring interval for your monitoring agents in the `kd4.ira.aggDataWindowSizeMinutes` property, you must also make the *same change* to the interval that the SOA Domain Management Server uses to query the monitoring agents and to calculate the length of the current metric timeframe for data that is displayed in the Operational Flow workspaces and views and the Group Summary view.

The SOA Domain Management Server does not query the monitoring agents as soon as a monitoring interval is complete. Instead, it delays the query by a default value of 2 minutes to ensure that all metrics are reported for the monitoring interval. If you want, you can modify this delay time.

To keep the SOA Domain Management Server time interval synchronized with the monitoring agent time interval, you must configure two additional resource environment properties:

ALLTEMAStartTime

This is the number of seconds after a completed monitoring interval that SOA Domain Management Server waits before querying the monitoring agent. The minimum permitted value is 70 seconds. The default value is 120 seconds (2 minutes). Specifying a time longer than 120 seconds only delays the display of the metric data in the view. If you specify a value less than 70 seconds, the default value of 120 is used instead. Message `KD4DM0030I` is written to the SOA Domain Management Server log file indicating both the configured value that you specified and the actual value that is used by SOA Domain Management Server.

ALLTEMARepeatTime

This property is used to determine the starting time and the length of monitoring intervals. Like the monitoring agent, SOA Domain Management Server assumes that monitoring intervals start on an hourly boundary (:00) and each succeeding monitoring interval occurs at a time incremented by this `ALLTEMARepeatTime` property value.

For example, a value of 3 results in monitoring intervals that start at :00, :03, :06, :09, and so on.

The valid values for this property are: 1, 2, 3, 4, 5, 10, 15, 20, 30, 60 The default value is 5.

If you specify a value other than one of these values, it is *rounded down* to the nearest valid value. If you specify a value less than 1, it is rounded up to 1. This behavior is consistent with how the monitoring agent validates the value of the `kd4.ira.aggDataWindowSizeMinutes` property. Message KD4DM0030I is written to the SOA Domain Management Server log file indicating your specified value and the actual value used by SOA Domain Management Server.

Be careful when changing this value from its default: A shorter repeat time causes SOA Domain Management Server to query the monitoring agents more frequently. The queries can have a performance impact, especially in environments with many monitoring agents. Similar to changing the monitoring interval for the agent, change this property only for demonstrations or proof of concept exercises. For production environments, use the default value of 5.

If you configure the `AllTEMARepeatTime` property to a value less than the default of 5 minutes, set `AllTEMAStartTime` to 70 seconds. This setting ensures that metrics are retrieved as soon as they are available from the monitoring agent. However, if you are monitoring in the DataPower environment and the service transaction response times are longer than 1 minute, continue to use the default monitoring interval (5 minutes) for the monitoring agent and the default settings for `AllTEMAStartTime` (120 seconds) and `AllTEMARepeatTime` (5 minutes).

Updating property values: Use the `kd4UpdateResourceProperty` scripts to set the values for the `AllTEMAStartTime` and `AllTEMARepeatTime` properties, following this general procedure:

1. Navigate to one of the following directories depending on your operating system:

- On Windows operating systems:
`<ITM_Home>\CNPS\Products\KD4\bin`
- On Linux and UNIX operating systems:
`<ITM_Home>/<platform>/cq/Products/KD4/bin`

For information about the values for directory variables, see “Operating system-dependent variables and paths” on page xi.

2. Run these scripts from a command prompt window using this syntax:

- On Windows:
`kd4UpdateResourceProperty <property> <value>`
- On Linux and UNIX operating systems:
`./kd4UpdateResourceProperty.sh <property> <value>`

The following parameters are specified in this syntax:

- `<property>` is either `AllTEMAStartTime` or `AllTEMARepeatTime`
- `<value>` is the new value for the specified property.

3. Restart Tivoli Enterprise Portal Server after running the `kd4UpdateResourceProperty` script for the new value to take effect.

Keep SOA Domain Management Server and the monitoring agent synchronized: Be sure to set the `AllTEMARepeatTime` property for SOA Domain Management

Server and the `kd4.ira.aggDataWindowSizeMinutes` property for each connected monitoring agent to the same value. This setting ensures that SOA Domain Management Server and the monitoring agent remain synchronized with each other.

If SOA Domain Management Server and a connected monitoring agent are configured with different values, incorrect interval data might be displayed for that monitoring agent, or SOA Domain Management Server might not display any data for that agent. This discrepancy in configured intervals might also cause SOA Domain Management Server to display incorrect time intervals in the service-to-service topology views and the Group Summary view. For example, if the monitoring agent is configured for a 5-minute interval and SOA Domain Management Server is configured for a 3-minute interval, SOA Domain Management Server might display a time interval of :02 to :05, which is not a valid interval based on either configured value (because :02 is not divisible by 3 or 5).

Each time SOA Domain Management Server queries an agent for data, it checks the monitoring interval configured for that agent. If SOA Domain Management Server detects that the agent is using an interval length different than its own, message `KD4DM0031E` is logged identifying the agent, the monitoring interval being used by the agent, and the SOA Domain Management Server interval length. This message is logged per agent, per query, when the interval values are not the same.

Examples: The following examples demonstrate the behavior of SOA Domain Management Server given different configuration values for the `AllTEMAStartTime` and `AllTEMARepeatTime` properties.

- Given the following default values:
 - `AllTEMAStartTime` = 120 (seconds)
 - `AllTEMARepeatTime` = 5 (minutes)

Assuming a start time for the SOA Domain Management Server at 9:00, the SOA Domain Management Server queries the monitoring agent at these times:

9:02, 9:07, 9:12, 9:17, and so on.

- Given the following configured values:
 - `AllTEMAStartTime` = 180 (seconds)
 - `AllTEMARepeatTime` = 10 (minutes)

Assuming a start time for the SOA Domain Management Server at 9:31, the SOA Domain Management Server queries the monitoring agent at these times:

9:33, 9:43, 9:53, 10:03, and so on.

In this example, because the monitoring interval starts on the hourly boundary (:00) and increments by 10 minutes, monitoring intervals start at 9:00, 9:10, 9:20, 9:30, 9:40, and so on. Starting at 9:31, the SOA Domain Management Server queries the monitoring agent 180 seconds (3 minutes) after the start of each monitoring interval.

- Given the following configured values:
 - `AllTEMAStartTime` = 70 (seconds)
 - `AllTEMARepeatTime` = 1 (minutes)

Assuming a start time for the SOA Domain Management Server at 9:15, the SOA Domain Management Server queries the monitoring agent at these times:

9:16, 9:17, 9:18, 9:19, and so on.

Verifying the configured monitoring interval

After changing the value of the `kd4.ira.aggDataWindowSizeMinutes` property and restarting the monitoring agent, you can verify what monitoring interval is being used by the agent using either of these methods:

- Edit the `KD4.dc.properties` file in the `<ITCAM4SOA_Home>/KD4/config/` directory and examine the value specified for the `kd4.ira.aggDataWindowSizeMinutes` property. If the specified value is not one of the supported values, it is rounded down to the nearest supported value.
- Use the Tivoli Enterprise Portal to display the Performance Summary workspace for an application server that is being monitored and examine the Interval Length column of the Services Inventory table view.

Memory usage considerations

When you set the `kd4.ira.aggDataWindowSizeMinutes` property to a value less than 5 minutes, the ITCAM for SOA Tivoli Enterprise Monitoring Agent uses additional memory to store the metrics and events. For example, if the monitoring interval is changed from 5 minutes to 1 minute, the Tivoli Enterprise Monitoring Agent requires five times more memory to store the metrics.

Reducing the amount of memory required to store metrics and events

By default, the Tivoli Enterprise Monitoring Agent stores metric data in memory for up to 70 minutes, depending on the values that are configured for these properties in the `KD4.dc.properties` file:

`kd4.ira.maxSvcInvRetainMinutes`

This property specifies the maximum number of minutes that metrics are retained in memory for these tables:

- Services Inventory (KD42IT)
- Services Inventory for Requester ID (KD42JT)

Metrics are stored in memory for this amount of time even if history collection is not enabled. The default value is 70 minutes. Valid values range from 5 to 1440 minutes; lower values are rounded up to 5 minutes, and higher values are rounded down to 1440 minutes.

`kd4.ira.maxRelMetricRetainMinutes`

This property specifies the maximum number of minutes that metrics are retained in memory for these tables:

- Relationship Request Metrics (KD43RQ)
- Relationship Response Metrics (KD43RP)

Metrics are stored in memory for this amount of time only when history collection is enabled. When history collection is not enabled, the metrics are removed from memory after the SOA Domain Management Server retrieves them. The default value is 70 minutes. Valid values range from 5 to 1440 minutes; lower values are rounded up to 5 minutes, and higher values are rounded down to 1440 minutes.

Separate settings control the storage of event records and errors for BPEL process components:

`kd4.ira.maxBpelEventRecords`

This property specifies the maximum number of BPEL event records that are retained in memory for the table Business Process Events (KD46BP).

The default value is 75000 records, using approximately 30 megabytes of memory. Valid values range from 10000 to 1000000.

kd4.ira.maxBpelEventRetainMinutes

This property specifies the maximum number of minutes that events are retained in memory for the table Business Process Events (KD46BP). The default value is 15 minutes. Valid values range from 5 to 15 minutes; lower values are rounded up to 5 minutes, and higher values are rounded down to 15 minutes.

kd4.ira.maxAssociatedErrorRetainMinutes

This property specifies the maximum number of minutes that associated error metrics are retained in memory for the table BPM Associated Errors (KD46AE). The default value is 15 minutes. Valid values range from 5 to 15 minutes; lower values are rounded up to 5 minutes, and higher values are rounded down to 15 minutes.

Important: The agent creates an associated error metric record after observing an error. Each time SDMS collects the information (by default, every 5 minutes), the number of occurrences for the error is reset to zero, but the metric record remains. If the error happens again, the number of occurrences is increased. If the error does not happen, the record with zero occurrences is retained in the table for this number of minutes, and then pruned.

To reduce the amount of memory used by the Tivoli Enterprise Monitoring Agent to store metrics and events, you can reduce the values of these properties so that metric and event data is stored for a shorter amount of time. To reduce the value of these properties, complete the following steps on the server where the ITCAM for SOA Tivoli Enterprise Monitoring Agent is installed:

1. Navigate to the `<ITCAM4SOA_Home>/KD4/config` directory and locate the `KD4.dc.properties` file.
2. Edit the `KD4.dc.properties` file using your preferred text editor.
3. In the `KD4.dc.properties` file, locate the lines that set the properties, for example:

```
kd4.ira.maxSvcInvRetainMinutes=70
kd4.ira.maxRelMetricRetainMinutes=70
kd4.ira.maxBpelEventRetainMinutes=15
kd4.ira.maxAssociatedErrorRetainMinutes=15
```
4. Change the values of these properties to another valid value.
5. Save your changes to the `KD4.dc.properties` file.

You do not have to restart the monitoring agent for the change to take effect. Repeat these steps for all servers where the ITCAM for SOA monitoring agent is installed to ensure that they are all configured with the same value.

Historical data collection: If historical data collection is enabled for the `Services_Inventory_610` or the `Services_Inventory_ReqID_610` attribute groups, ensure that the history collection interval is less than the number of minutes specified by the `kd4.ira.maxSvcInvRetainMinutes` property. If historical data collection is enabled for the `Rel_Resp_Metrics` or the `Rel_Req_Metrics` attribute groups, ensure that the history collection interval is less than the number of minutes specified by the `kd4.ira.maxRelMetricRetainMinutes` property.

Reducing the amount of memory required to store mediation flow request messages

By default, the ITCAM for SOA Tivoli Enterprise Monitoring Agent retains a mediation flow request message in memory until it receives a corresponding response message. The monitoring agent uses the response message to calculate the elapse time for the mediation flow operation.

The maximum numbers of minutes that the mediation flow request messages are held in memory is specified in the `kd4.ira.maxMediationFlowOpenMsgRetainMinutes` property in the `KD4.dc.properties` file. If a corresponding response message is not received within the number of minutes specified by this property, the monitoring agent deletes the request message.

The default value of the `kd4.ira.maxMediationFlowOpenMsgRetainMinutes` property is 15 minutes. The minimum value for this property is 1 minute.

To modify the number of minutes that mediation flow request messages are held in memory, on each server where the monitoring agent is installed, complete the following steps:

1. Navigate to the `ITCAM4SOA_Home/KD4/config` directory and locate the `KD4.dc.properties` file.
2. Edit the `KD4.dc.properties` file using your preferred text editor.
3. In the `KD4.dc.properties` file, locate the line that sets the following property:
`kd4.ira.maxMediationFlowOpenMsgRetainMinutes=15`
4. Change the values of this property to another valid value.
5. Save your changes to the `KD4.dc.properties` file.

You do not have to restart the monitoring agent for the change to take effect.

Important: Retaining mediation flow request messages in memory for a long time might cause the monitoring agent to run out of memory. A long retain period might reduce the processing performance of the monitoring agent. In production environments, we recommend that you use the default value of 15.

Preventing data for the Services Message Metric attributes group from being saved

For ITCAM for SOA version 6.1.0, the data in the Services Message Metric attributes group was used primarily by the IBM Web Services Navigator, provided with ITCAM for SOA, when the Navigator was configured to retrieve data from the data warehouse. For ITCAM for SOA version 7.1.0 and later, the IBM Web Services Navigator no longer uses this attribute group. You should not enable history collection for this attribute group unless you are using it for some custom purpose.

Selectively disabling data collection

ITCAM for SOA provides the facilities to limit which combinations of service ports and operations are monitored. By default, all combinations of service port names and operation names, and their respective namespaces are monitored. For test environments, this setting is probably best. For production environments, limit this to only those combinations of service port names, operation names, and

namespaces that you are interested in managing. Modify what is monitored through the use of the monitor control commands documented in Chapter 11, “Take Action commands,” on page 223.

Configuring the Cleanup Service parameters

ITCAM for SOA stores current performance metrics in the SDMS component. This storage is limited and the Cleanup Service periodically clears it. Historical data remains stored in Tivoli Data Warehouse. By default, the Cleanup Service runs at midnight, every 24 hours. You can change the time when the service starts the next time and the period after which it starts again. You can also configure the Cleanup Service to delete BPEL activity start events for which no matching end is found after a set amount of days.

About this task

You require database administrator privileges to make changes to Cleanup Service configuration.

Metrics for BPEL processes, individual mediation primitives in SCA mediation flows, and BPDs are not stored in the Data Warehouse. When the Cleanup Service runs, these metrics are permanently deleted.

If the start of a BPEL activity has been monitored, but its end has not been monitored for any reason, the start event will not be deleted from the database by default. The event will affect some metrics for the BPEL process. You can configure the Cleanup Service to delete such events after a set amount of days.

Important: If you have deleted a BPD node and the same BPD is to run again, you must ensure the Cleanup Service runs first. Otherwise, the BPD will not be visible in ITCAM for SOA. To ensure the Cleanup Service runs, you can set its startup time to a few minutes after the current time; after it runs, you can configure it again for the usual setting.

Make these changes on the Tivoli Enterprise Portal Server.

Tip: The startup time must be set according to the time zone configured on the Tivoli Enterprise Portal Server host.

Procedure

1. In a command prompt, navigate to the following directory:
 - On a Windows system, *ITM_Home\CNPS\Products\KD4\bin*
 - On a Linux or UNIX system, *ITM_Home/platform/cq/Products/KD4/bin*
2. If you want to change the time when the Cleanup Service will run the next time, calculate *starttime* as the number of milliseconds since midnight. For example, for a 2:00 PM (14:00) start, this value would be $14 \times 3600 \times 1000 = 50400000$. Use the following command:
 - On Windows systems, `kd4UpdateResourceProperty CleanupDbStartTime starttime`
 - On Linux and UNIX systems, `./kd4UpdateResourceProperty.sh CleanupDbStartTime starttime`
3. If you want to change the time period in which Cleanup Service will run again after running the next time, set *repeattime* in hours (for example, 24 to run every day at the same time). Use the following command:

- On Windows systems, `kd4UpdateResourceProperty CleanupDbRepeatTime repeattime`
 - On Linux and UNIX systems, `./kd4UpdateResourceProperty.sh CleanupDbRepeatTime repeattime`
4. If you want to configure the Cleanup Service to delete BPEL start events after a certain amount of days, set *maxage* in days. If you want to configure the service not to delete such events any longer, set *maxage* to 0. Use the following command:
 - On Windows systems, `kd4UpdateResourceProperty CleanupDbDeleteBpelEventsMaxAge maxage`
 - On Linux and UNIX systems, `./kd4UpdateResourceProperty.sh CleanupDbDeleteBpelEventsMaxAge maxage`
 5. If you have configured the Cleanup Service startup time or repeat period, restart the Tivoli enterprise Portal Service. If you have configured only the deletion of BPEL start events, you do not need to restart the server.

Setting data age for BPDs

You can change the monitored data age for Business Process Definitions (BPDs). To do this, use the following resource environment properties on the SOA Domain Management Server (SDMS):

BPDUnmatchedStartEventsMaxAge

A BPD normally includes interactive tasks, and therefore it remains active until the human operator completes it. By default, ITCAM for SOA monitors all active BPDs.

If you set this property to a value (in days), then, if a BPD stays active for this amount of time, ITCAM for SOA considers it abandoned and no longer includes it in monitoring.

You can set this property if, as a part of normal operation in your environment, a BPD can be abandoned, and you do not want such BPDs to be represented in ITCAM for SOA metrics.

BPDInstanceDataMaxAge

By default, SDMS removes information for completed BPDs after one day. If you need to include information for a longer period in BPD metrics, set this property to the period length (in days). This setting might be useful for determining a long-term issue with a BPD. For normal monitoring, use the default value.

Important: If you need to analyze BPD information over a longer period, you cannot use Tivoli Monitoring historical data, as it is not collected for BPD monitoring. Use this property instead.

Updating property values: Use the `kd4UpdateResourceProperty` scripts to set the values for the `AllTEMAStartTime` and `AllTEMARepeatTime` properties, following this general procedure:

1. On the Tivoli Enterprise Portal Server host, navigate to one of the following directories depending on your operating system:
 - On Windows operating systems:
`<ITM_Home>\CNPS\Products\KD4\bin`
 - On Linux and UNIX operating systems:
`<ITM_Home>/<platform>/cq/Products/KD4/bin`

For information about the values for directory variables, see “Operating system-dependent variables and paths” on page xi.

2. Run these scripts from a command prompt window using this syntax:

- On Windows:

```
kd4UpdateResourceProperty <property> <value>
```

- On Linux and UNIX operating systems:

```
./kd4UpdateResourceProperty.sh <property> <value>
```

The following parameters are specified in this syntax:

- *<property>* is either *BPDUnmatchedStartEventsMaxAge* or *BPDInstanceDataMaxAge*
- *<value>* is the new value for the specified property.

3. Restart Tivoli Enterprise Portal Server after running the `kd4UpdateResourceProperty` script for the new value to take effect.

Important: The settings applies to monitoring all servers in your environment.

Excluding faults from the calculation of service metric values

Service inventory metrics are displayed in the `Services_Inventory_610` (KD42IT) and the `Services_Inventory_ReqID_610` (KD42JT) attribute groups. The elapsed time and message length service inventory attributes might be distorted if the monitoring agent includes both fault messages and successful messages in the summarization process.

Table 64 displays the elapsed time and message length service inventory attributes in the `Services_Inventory_610` and the `Services_Inventory_ReqID_610` attribute groups:

Table 64. Elapsed time and message length attributes in the Service Inventory attribute groups

Attribute	Description
Avg_Msg_Length	The average message length, in bytes, observed during this interval (including headers when possible).
Msg_Count	The number of messages that are transmitted during the monitoring interval.
Elapsed_Time_Msg_Count	The number of messages that are transmitted during the monitoring interval that contain an elapsed time value.
Avg_Elapsed_Time	The average elapsed time, in milliseconds, of all messages that are transmitted during the monitoring interval.
Max_Msg_Length	The length of the largest message, in bytes, transmitted during the monitoring interval.
Max_Elapsed_Time	The longest elapsed time, in milliseconds, of any messages that are transmitted during the monitoring interval.
Min_Msg_Length	The length of the shortest message, in bytes, transmitted during the monitoring interval.
Min_Elapsed_Time	The shortest elapsed time, in milliseconds, of any messages that are transmitted during the monitoring interval.
Std_Dev_Msg_Length	The standard deviation, in bytes, of all message lengths that are transmitted during the monitoring interval.
Std_Dev_Elapsed_Time	The standard deviation of elapsed times, in milliseconds, of all messages that are transmitted during this monitoring interval.

Beginning with ITCAM for SOA version 7.2 Fix Pack 1, a new property is added to the `KD4.dc.properties` file to exclude fault messages from the elapsed time and message length metrics. By default, the value is set to 0. When set to 0, fault messages are included in the calculation of the attribute values in Table 64 on page 350. When set to 1, fault messages are excluded from the calculation.

To exclude fault messages from the elapsed time and message length service inventory metrics, complete the following steps on each server where the ITCAM for SOA monitoring agent is installed:

1. Navigate to the `<ITCAM for SOA_Home>/KD4/config` directory and locate the `KD4.dc.properties` file.
2. Edit the `KD4.dc.properties` file using your preferred text editor.
3. In the `KD4.dc.properties` file, locate the line that sets the `kd4.ira.excludefault.controls.enabled` property. For example,
`kd4.ira.excludefault.controls.enabled=0`
4. Set the value to 1 to exclude fault messages from the elapsed time and message length calculations.
5. Save your changes to the `KD4.dc.properties` file.
6. Restart the Tivoli Enterprise Monitoring Agent for the change to take effect.

Important:

- You can configure the ITCAM for SOA monitoring agent to exclude fault messages from the elapsed time and message length calculations. Any message received from a DataPower SOA appliance that includes one of the following combination of variables is considered to be a fault message:
 - `error-code` and `error-subcode`
 - `fault-code` and `fault-message`
 - `error-code`, `error-subcode`, `fault-code`, and `fault-message`
- If historical data collection is enabled for the `Services_Inventory_610` and the `Services_Inventory_ReqID_610` attribute groups, service inventory metrics are stored in the Tivoli Data Warehouse. If you exclude fault messages from the service inventory elapsed time and message length calculations, fault messages are excluded from the elapsed time and message length metrics that are stored in the Tivoli Data Warehouse.

Reducing on-demand user resource usage

On-demand processor usage is defined as processor usage generated when a user requests data, for example, when Tivoli Enterprise Portal Server clients investigate alerts by navigating workspaces. Any such continuous data collection must be tuned. Another example is Tivoli Enterprise Portal server clients that have auto-update enabled on workspaces with graphs or tables. Frequently used workspaces with tabular views showing data from monitoring agents that return large quantities of data are also well worth tuning given these two considerations: auto-refreshed screens always occupy memory and large intermittent reports can cause a spike in memory utilization.

The Tivoli Enterprise Portal client typically returns results quickly, but there are still opportunities to make it faster and reduce processor usage for all IBM Tivoli Monitoring components. As more users use the Tivoli Enterprise Portal client, you can prevent the client and its server from becoming overloaded. Following are

some tips to reduce processor and memory required by the Tivoli Enterprise Portal client and server, which can result in better response time for everyone's workspace fetches and refreshes.

The query assigned to a chart or table view requests data from a particular attribute group when you open or refresh the workspace. You can dramatically reduce the number of data samples and the amount of data retrieved by:

- Customizing the query to filter out unwanted data.
- Reducing the number of rows and columns.
- Applying the same query to multiple views in a workspace.
- Although auto-refresh is not recommended, you can also collect agent data less frequently by adjusting auto-refresh to longer intervals and auto-refreshing on graphic view instead of tabular view.

Calculating and reducing memory required for workspace queries

The entire results set from a query is stored at the Tivoli Enterprise Portal server so that you can page through it. The current page and previous page are stored on the Tivoli Enterprise Portal client. Tivoli Enterprise Portal client help files describe every report and their columns, with some products providing field lengths. Approximate the record length and multiply it by the number of rows to estimate the number of bytes required for each user who displays the data.

Using custom queries to reduce resource usage

Custom queries can reduce network traffic and processing cycles at the agent and at Tivoli Enterprise Monitoring Server. They can also reduce memory consumption at the Tivoli Enterprise Portal Server and its client. Queries reduce resource usage by limiting the number of rows and columns passed from the agent to the Tivoli Enterprise Monitoring Server.

Do not confuse custom queries with view filters, which have no effect on reducing any usage of resources and actually increase the Tivoli Enterprise Portal client processor requirements. Filters are applied by the client on the current page. If more than one page is returned by the query, only a subset of the data is viewed on each page. Increasing the page size is an option available in Tivoli Enterprise Portal. This option typically provides more filtered data on each page, but increases the client memory requirements as now the two pages per query stored at the client are larger. It is more efficient to filter in the queries.

Restricting the number of rows

Within the custom query, if you can filter the number of rows to return, add this setting to the query to reduce the data returned. For example, you might be only interested in rows with response time greater than 2000 milliseconds.

Restricting the number of columns

When building the query, select only those columns that contain data that is required. For example, the current and parent correlator provides little value to a user viewing the data. Therefore, do not include these columns in the result.

Using the same query in a workspace

Having multiple views in a workspace with different data from different tables is acceptable. However, if you have a graph in one view and a table in another view,

they can be generated from the same custom query. If you create two unique queries, it drives the data collection at the agent and increases data collection processor usage, which is greater than any savings that you can achieve anywhere else. Use one query per table in a workspace that is shared by all panes within that workspace. Your custom query must retrieve all of the rows and columns that are required by any pane in the workspace. If you have many tables and charts and they use different queries, the entire result set for each query is stored on the Tivoli Enterprise Portal Server. The 100 rows (default page size) from each query currently being viewed and the previous page of any pane viewed are stored on the Tivoli Enterprise Portal client.

Collecting agent data less frequently

Do not use auto-refreshing. The navigator and graphic views automatically refresh without auto-refreshing being specified, providing instantaneous alerts. You can then navigate to workspaces with real data or alerts. The graphic view, which displays alerts but no data, affects only client memory, not the Tivoli Enterprise Portal Server.

Appendix D. ITCAM for SOA SDMS agent Attributes reference

Attributes are the application properties that are being measured and reported by the ITCAM for SOA SDMS agent.

Attribute groups for the ITCAM for SOA SDMS monitoring agent

The ITCAM for SOA SDMS agent contains the following attribute groups. The table name of the group depends on the maximum table name limits of the target database being used for the Tivoli Data Warehouse. If the maximum name is 30 characters, any warehouse table name longer than 30 characters is shortened to 30 characters.

- Attribute group name: Performance Object Status
 - Table name: KS4POBJST
 - Warehouse table name: KS4_PERFORMANCE_OBJECT_STATUS or KS4POBJST
- Attribute group name: Process Groups
 - Table name: KS4PGS
 - Warehouse table name: KS4_PROCESS_GROUPS or KS4PGS
- Attribute group name: SOA All Groups
 - Table name: KS4SOAALLG
 - Warehouse table name: KS4_SOA_ALL_GROUPS or KS4SOAALLG
- Attribute group name: SOA Group Status
 - Table name: KS4GRPSTAT
 - Warehouse table name: KS4_SOA_GROUP_STATUS or KS4GRPSTAT

Attributes in each attribute group

Attributes in each ITCAM for SOA SDMS agent attribute group collect data that the agent uses for monitoring.

The descriptions of the attribute groups contain the following information:

Historical group

Whether the attribute group is a historical type that you can roll off to a data warehouse.

Attribute descriptions

Information such as description, type, source, and warehouse name, as applicable, for each attribute in the attribute group.

Some attributes are designated as key attributes. A *key attribute* is an attribute that is used in warehouse aggregation to identify rows of data that represent the same object.

The Source information sometimes uses C programming code syntax for if-then-else clauses to describe how an attribute is derived, for example:

```
(CPU_Pct < 0 ) || (Memory_Pct < 0 )? 0 : 1
```

This example means that if the CPU_Pct attribute is less than 0 or if the Memory_Pct attribute is less than 0, then the attribute is set to 0. Otherwise, the attribute is set to 1.

Performance Object Status attribute group

The Performance Object Status attribute group contains information that reflects the status of other attribute groups so you can see the status of all of the performance objects that make up this application all at once. Each of these other performance attribute groups is represented by a row in this table (or other type of view). The status for an attribute group reflects the result of the last attempt to collect data for that attribute group, which allows you to see whether the agent is performing correctly. Unlike other attribute groups, the Performance Object Status attribute group does not reflect the state of the monitored application. This attribute group is most often used to determine why data is not available for one of the performance attribute groups.

Historical group

This attribute group is eligible for use with Tivoli Data Warehouse.

Attribute descriptions

The following list contains information about each attribute in the Performance Object Status attribute group:

Node attribute: This attribute is a key attribute.

Description

The managed system name of the agent.

Type

String

Source

The source for this attribute is the agent.

Warehouse name

NODE

Timestamp attribute

Description

The local time at the agent when the data was collected.

Type

String

Source

The source for this attribute is the agent.

Warehouse name

TIMESTAMP

Query Name attribute: This attribute is a key attribute.

Description

The name of the attribute group.

Type

String

Warehouse name

QUERY_NAME or ATTRGRP

Object Name attribute

Description

The name of the performance object.

Type

String

Warehouse name

OBJECT_NAME or OBJNAME

Object Type attribute

Description

The type of the performance object.

Type

Integer with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values shown in parentheses. The following values are defined:

- WMI (0)
- PERFMON (1)
- WMI ASSOCIATION GROUP (2)
- JMX (3)
- SNMP (4)
- SHELL COMMAND (5)
- JOINED GROUPS (6)
- CIMOM (7)
- CUSTOM (8)
- ROLLUP DATA (9)
- WMI REMOTE DATA (10)
- LOG FILE (11)
- JDBC (12)
- CONFIG DISCOVERY (13)
- NT EVENT LOG (14)
- FILTER (15)
- SNMP EVENT (16)
- PING (17)
- DIRECTOR DATA (18)
- DIRECTOR EVENT (19)
- SSH REMOTE SHELL COMMAND (20)

Any other value is the value returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

OBJECT_TYPE or OBJTYPE

Object Status attribute**Description**

The status of the performance object.

Type

Integer with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values shown in parentheses. The following values are defined:

- ACTIVE (0)
- INACTIVE (1)

Any other value is the value returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

OBJECT_STATUS or OBJSTTS

Error Code attribute**Description**

The error code associated with the query.

Type

Integer with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values shown in parentheses. The following values are defined:

- NO ERROR (0)
- GENERAL ERROR (1)
- OBJECT NOT FOUND (2)

- COUNTER NOT FOUND (3)
- NAMESPACE ERROR (4)
- OBJECT CURRENTLY UNAVAILABLE (5)
- COM LIBRARY INIT FAILURE (6)
- SECURITY INIT FAILURE (7)
- PROXY SECURITY FAILURE (9)
- NO INSTANCES RETURNED (10)
- ASSOCIATOR QUERY FAILED (11)
- REFERENCE QUERY FAILED (12)
- NO RESPONSE RECEIVED (13)
- CANNOT FIND JOINED QUERY (14)
- CANNOT FIND JOIN ATTRIBUTE IN QUERY 1 RESULTS (15)
- CANNOT FIND JOIN ATTRIBUTE IN QUERY 2 RESULTS (16)
- QUERY 1 NOT A SINGLETON (17)
- QUERY 2 NOT A SINGLETON (18)
- NO INSTANCES RETURNED IN QUERY 1 (19)
- NO INSTANCES RETURNED IN QUERY 2 (20)
- CANNOT FIND ROLLUP QUERY (21)
- CANNOT FIND ROLLUP ATTRIBUTE (22)
- FILE OFFLINE (23)
- NO HOSTNAME (24)
- MISSING LIBRARY (25)
- ATTRIBUTE COUNT MISMATCH (26)
- ATTRIBUTE NAME MISMATCH (27)
- COMMON DATA PROVIDER NOT STARTED (28)
- CALLBACK REGISTRATION ERROR (29)
- MDL LOAD ERROR (30)
- AUTHENTICATION FAILED (31)
- CANNOT RESOLVE HOST NAME (32)
- SUBNODE UNAVAILABLE (33)
- SUBNODE NOT FOUND IN CONFIG (34)
- ATTRIBUTE ERROR (35)
- CLASSPATH ERROR (36)
- CONNECTION FAILURE (37)
- FILTER SYNTAX ERROR (38)
- FILE NAME MISSING (39)
- SQL QUERY ERROR (40)
- SQL FILTER QUERY ERROR (41)
- SQL DB QUERY ERROR (42)
- SQL DB FILTER QUERY ERROR (43)
- PORT OPEN FAILED (44)
- ACCESS DENIED (45)
- TIMEOUT (46)
- NOT IMPLEMENTED (47)
- REQUESTED A BAD VALUE (48)
- RESPONSE TOO BIG (49)
- GENERAL RESPONSE ERROR (50)
- SCRIPT NONZERO RETURN (51)
- SCRIPT NOT FOUND (52)
- SCRIPT LAUNCH ERROR (53)
- CONF FILE DOES NOT EXIST (54)
- CONF FILE ACCESS DENIED (55)
- INVALID CONF FILE (56)
- EIF INITIALIZATION FAILED (57)

- CANNOT OPEN FORMAT FILE (58)
- FORMAT FILE SYNTAX ERROR (59)
- REMOTE HOST UNAVAILABLE (60)
- EVENT LOG DOES NOT EXIST (61)
- PING FILE DOES NOT EXIST (62)
- NO PING DEVICE FILES (63)
- PING DEVICE LIST FILE MISSING (64)
- SNMP MISSING PASSWORD (65)
- DISABLED (66)
- URLS FILE NOT FOUND (67)
- XML PARSE ERROR (68)
- NOT INITIALIZED (69)
- ICMP SOCKETS FAILED (70)
- DUPLICATE CONF FILE (71)

Any other value is the value returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

ERROR_CODE or ERRCODE

Process Groups attribute group

This is the discovery attribute group for Process Groups. This attribute group is for internal use only.

Historical group

This attribute group is eligible for use with Tivoli Data Warehouse.

Attribute descriptions

The following list contains information about each attribute in the Process Groups attribute group:

Node attribute: This attribute is a key attribute.

Description

The managed system name of the agent.

Type

String

Source

The source for this attribute is the agent.

Warehouse name

NODE

Timestamp attribute

Description

The local time at the agent when the data was collected.

Type

String

Source

The source for this attribute is the agent.

Warehouse name

TIMESTAMP

Subnode MSN attribute: This attribute is a key attribute.

Description

The Managed System Name of the subnode agent.

Type

String

Warehouse name

SUBNODE_MSN or SN_MSN

Subnode Affinity attribute

Description

The affinity for the subnode agent.

Type	String
Warehouse name	SUBNODE_AFFINITY or SN_AFFIN
<u>Subnode Type attribute: This attribute is a key attribute.</u>	
Description	The Node Type of this subnode.
Type	String
Warehouse name	SUBNODE_TYPE or SN_TYPE
<u>Subnode Resource Name attribute</u>	
Description	The Resource Name of the subnode agent.
Type	String
Warehouse name	SUBNODE_RESOURCE_NAME or SN_RES
<u>Subnode Version attribute</u>	
Description	The Version of the subnode agent.
Type	String
Warehouse name	SUBNODE_VERSION or SN_VER

SOA All Groups attribute group

All ITCAM for SOA groups.

Historical group

This attribute group is eligible for use with Tivoli Data Warehouse.

Attribute descriptions

The following list contains information about each attribute in the SOA All Groups attribute group:

Node attribute: This attribute is a key attribute.

Description	The managed system name of the agent.
Type	String
Source	The source for this attribute is the agent.
Warehouse name	NODE

Timestamp attribute

Description	The local time at the agent when the data was collected.
Type	String
Source	The source for this attribute is the agent.
Warehouse name	TIMESTAMP

Group ID attribute: This attribute is a key attribute.

Description	The ID of the group.
--------------------	----------------------

Type	String
Warehouse name	GROUP_ID or GRPID
<u>Type attribute</u>	
Description	The type of group.
Type	Integer (32-bit gauge) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values shown in parentheses. The following values are defined: <ul style="list-style-type: none"> • Unknown Group (0) • Service Group (1) • Process Group (2) Any other value is the value returned by the agent in the Tivoli Enterprise Portal.
Warehouse name	TYPE
<u>Label attribute</u>	
Description	The name of the group
Type	String
Warehouse name	LABEL
<u>Description attribute</u>	
Description	A description of the group.
Type	String
Warehouse name	DESCRIPTION or DESC
<u>Number Of Unavailable FS attribute</u>	
Description	The number of unavailable front-end services in the group.
Type	Integer (32-bit gauge) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values shown in parentheses. The following values are defined: <ul style="list-style-type: none"> • No Value (-1) Any other value is the value returned by the agent in the Tivoli Enterprise Portal.
Warehouse name	NUMBER_OF_UNAVAILABLE_FS or NUMUNFS
<u>Number Of Aggregation attribute</u>	
Description	The number of operation aggregates in the group.
Type	Integer (32-bit gauge) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values shown in parentheses. The following values are defined: <ul style="list-style-type: none"> • Value Exceeds Maximum (2147483647)

- Value Exceeds Minimum (-2147483648)

Any other value is the value returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

NUMBER_OF_AGGREGATION or NUMAGGR

Performance attribute

Description

The average response time, in seconds, for all service operations in the group

Type

Real number (32-bit gauge) with 3 decimal places of precision with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values shown in parentheses. The following values are defined:

- No Value (-1)

Any other value is the value returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

PERFORMANCE or PERFORM

Status attribute

Description

The status of the group.

Type

String

Warehouse name

STATUS

Volume attribute

Description

Message counts for the group.

Type

Integer (32-bit gauge) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values shown in parentheses. The following values are defined:

- No Value (-1)

Any other value is the value returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

VOLUME

SOA Group Status attribute group

ITCAM for SOA group status.

Historical group

This attribute group is eligible for use with Tivoli Data Warehouse.

Attribute descriptions

The following list contains information about each attribute in the SOA Group Status attribute group:

Node attribute: This attribute is a key attribute.

Description

The managed system name of the agent.

Type

String

Source	The source for this attribute is the agent.
Warehouse name	NODE
<u>Timestamp attribute</u>	
Description	The local time at the agent when the data was collected.
Type	String
Source	The source for this attribute is the agent.
Warehouse name	TIMESTAMP
<u>Group ID attribute: This attribute is a key attribute.</u>	
Description	Group ID
Type	String
Warehouse name	GROUP_ID or GRPID
<u>Type attribute</u>	
Description	The type of group.
Type	Integer (32-bit gauge) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values shown in parentheses. The following values are defined: <ul style="list-style-type: none"> • Unknown Group (0) • Service Group (1) • Process Group (2) Any other value is the value returned by the agent in the Tivoli Enterprise Portal.
Warehouse name	TYPE
<u>Label attribute</u>	
Description	The name of the group
Type	String
Warehouse name	LABEL
<u>Description attribute</u>	
Description	A description of the group.
Type	String
Warehouse name	DESCRIPTION or DESC
<u>Number Of Unavailable FS attribute</u>	
Description	The number of unavailable front-end services in the group.
Type	Integer (32-bit gauge) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The

warehouse and queries return the values shown in parentheses. The following values are defined:

- No Value (-1)

Any other value is the value returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

NUMBER_OF_UNAVAILABLE_FS or NUMUNFS

Number Of Aggregation attribute

Description

The number of operation aggregates in the group.

Type

Integer (32-bit gauge) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values shown in parentheses. The following values are defined:

- Value Exceeds Maximum (2147483647)
- Value Exceeds Minimum (-2147483648)

Any other value is the value returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

NUMBER_OF_AGGREGATION or NUMAGGR

Performance attribute

Description

The response time in seconds.

Type

Real number (32-bit gauge) with 3 decimal places of precision with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values shown in parentheses. The following values are defined:

- No Value (-1)

Any other value is the value returned by the agent in the Tivoli Enterprise Portal.

Warehouse name

PERFORMANCE or PERFORM

Status attribute

Description

The status of the group.

Type

String

Warehouse name

STATUS

Volume attribute

Description

Message counts for the group.

Type

Integer (32-bit gauge) with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values shown in parentheses. The following values are defined:

- No Value (-1)

Any other value is the value returned by the agent in the Tivoli Enterprise Portal.

Warehouse name
VOLUME

Appendix E. Accessibility

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

The accessibility features in the product enable users to:

- Use assistive technologies, such as screen reader software and digital speech synthesizers, to hear what is displayed on the screen. Consult the product documentation of the assistive technology for details on using the technology with this product.
- Perform tasks with the software using only the keyboard.

General Navigation

Each page has four main sections:

- Headerbar
- Toolbar
- Main tabs
- Content

Each page has navigation points for screen readers. The following navigation points are all H1:

- Title bar
- Main tabs
- Main form
- Section labels
- Table labels

Menu Navigation

You use the Go To menu at the top of the screen to navigate to any of the applications that you have access to. The Go To menu is a cascading menu that is three levels deep at its deepest point. The following instructions describe how to get started with JAWS:

1. To get to the Go To menu press Alt+G.
2. When you open the menu, JAWS reads the first application in the menu. If JAWS does not begin to read the entry, restart the screen reader.
3. Navigate the list of applications in the menus by using the arrow keys.
4. JAWS indicates if a menu item has submenus. To get to a submenu, press the right arrow or enter.
5. Press the left arrow to move up a level in the hierarchy. If you press the left arrow at the highest level of the Go To menu, you leave the menu completely.
6. Press the Enter key to enter an application.

Accessibility help

The Accessibility Help panels provide details on general navigation, menu navigation, and hot keys. Click **Accessibility Help** from the toolbar of the product to access the help panels.

Screen reader setting

The product contains a screen reader flag. When you turn on the screen reader flag, the user interface is optimized to work with JAWS for Windows®. You use the **User** tab in the Users application to turn on the screen reader flag.

Keyboard shortcuts

You can navigate within the applications by using a combination of keys.

Accessible reports

To use the accessibility tools to read reports, you must access the reports in Microsoft Excel. In the reports applications, select the **Run Reports** option in the **Select Action** menu. With this option, you can email an .xls file version of a report to yourself at a scheduled time.

IBM and accessibility

For more information about the commitment that IBM has to accessibility, see the IBM Human Ability and Accessibility Center. The IBM Human Ability and Accessibility Center is at the following web address: <http://www.ibm.com/able>

Index

A

- accessibility x
- AddFltrCntrl_610 take action
 - command 226
- Adding operations to a process
 - group 167
- Adding operations to a service
 - group 166
- additional information
 - attributes 355
- AddMntrCntrl_610 take action
 - command 229
- AddRequesterIdentity_610 take action
 - command 233
- aggregate 71
- aggregate, operation 24
- application server runtime
 - environment 1
- Application Server Services Management
 - workspace 24, 39, 40
- attribute group 355
- attribute groups
 - list of all 355
 - overview 355
 - Performance Object Status 356
 - Process Groups 359
 - SOA All Groups 360
 - SOA Group Status 362
- attributes 355
 - additional information 355
 - Description 361, 363
 - Error Code 357
 - Group ID 360, 363
 - Label 361, 363
 - Node 356, 359, 360, 362
 - Number Of Aggregation 361, 364
 - Number Of Unavailable FS 361, 363
 - Object Name 356
 - Object Status 357
 - Object Type 356
 - overview 355
 - Performance 362, 364
 - Performance Object Status 356
 - Process Groups 359
 - Query Name 356
 - SOA All Groups 360
 - SOA Group Status 362
 - Status 362, 364
 - Subnode Affinity 359
 - Subnode MSN 359
 - Subnode Resource Name 360
 - Subnode Type 360
 - Subnode Version 360
 - Timestamp 356, 359, 360, 363
 - Type 361, 363
 - Volume 362, 364
- Average Message Size by Operation
 - view 40, 41
- Average Message Size by
 - Service:Operation:Type view 43

- Average Response Time by Operation
 - view 40, 41

B

- books vii
- BPD 90
- BPD node, deleting 123
- BPEL 89
- Business Process Definition 90
- Business Process Details view 66
- Business Processes for Service Port
 - view 65
- Business Processes for Service view 63

C

- call relationship 71
- client request 73
- client response 73
- client, deleting unmanaged 125
- Component view 85
- conventions
 - typeface xi
- cookies 374
- Creating a group 164

D

- DataPower 1, 128
- DataPower Console workspace 24, 46
- DataPower mediation 72
- DataPower WebGUI console 31
- DeleteRequesterIdentity_610 take action
 - command 235
- DeleteSubnode take action
 - command 236
- deleteUnmanagedClientAndOperations
 - script 125
- Deleting a service group or process
 - group 172
- deleting unmanaged object 123
- DelFltrCntrl_610 take action
 - command 238
- DelMntrCntrl_610 take action
 - command 241
- Description attribute 361, 363
- directories, variables for xi
- DisableDC_610 take action
 - command 243
- DisableReqIDMntr_610 take action
 - command 244
- Dynamic data 88

E

- Editing a group 165
- education
 - See Tivoli technical training

- EnableDC_610 take action command 245
- EnableReqIDMntr_610 take action
 - command 247
- Error Code attribute 357
- expert advice, situation 200

F

- Fault Details view 44
- Fault Summary by Operation view 41
- Faults Summary workspace 24, 25, 44
- firmware, DataPower 128
- Front-end service 150

G

- Group Detail 164
- Group ID attribute 360, 363
- Group Summary 152
- Group Summary workspace 24, 26
- Groups 164

H

- historical data
 - viewing 32

I

- IBM WebSphere Enterprise Service
 - Bus 1
- instance, operation 24
- integrating with other products 327
- integrating,
 - IBM Tivoli Business Service
 - Manager 327
 - IBM WebSphere Business
 - Modeler 328
 - IBM WebSphere Service Registry and
 - Repository 327
 - Tivoli Application Dependency
 - Discovery Manager 328
 - Tivoli Change and Configuration
 - Management 328
- Interaction Detail view 92
- IP address, updating for operation
 - instance 126
- ITCAM for SOA 1
- ITCAM4SOA_Home xi
- ITM_home xi

J

- Java Specification Request (JSR) 1
- JSR 101 and 109
 - see Java Specification Request 1

K

kd4UpdateIP, update IP address 126

L

Label attribute 361, 363
link symbol 185

M

manuals vii
mediation 1
 DataPower 72
 SCA 72
 WebSphere Message Broker 72
Mediation flow 89
Message and Fault Count view 51
Message Arrival by Operation view 39
Message Arrival by Service view 39
Message Arrival Details view 39
Message Arrival workspace 24, 37
Message Size by Operation view 52
Message Summary workspace 24, 25, 42
metric, relationship 78
Monitored Requester Identities view 50

N

Navigator
 ITCAM for SOA view 26
 Physical view 24
Navigator Physical view 36
Navigator view 23
Node attribute 356, 359, 360, 362
Number Of Aggregation attribute 361, 364
Number of Faults by Operation view 44
Number of Messages by Operation view 40
Number of Messages by
 Service:Operation:Type view 42
Number Of Unavailable FS attribute 361, 363

O

Object Name attribute 356
Object Status attribute 357
Object Type attribute 356
offline, operation instance status 76
one-way service 73, 79
open situation, operation instance status 75
operation 71
operation aggregate 24, 71
Operation Flow - All Flows view 92
Operation Flow for Process Group 161
Operation Flow for Service Group 161
operation instance 24, 71
 status, calculating 74
operation instance, deleting 122
operation instance, deleting unmanaged 125
operation instance, IP address update 126

Operation view 85
Operational Flow for Application Server workspace 24, 29, 91, 95
Operational Flow for Operation workspace 24, 29, 91, 93
Operational Flow workspace 92
Operational Flows workspace 24, 26, 91, 92
ordering publications ix

P

Performance attribute 362, 364
Performance Object Status attribute group 356
Performance Summary for Requester Identity workspace 24, 29
Performance Summary workspace 24, 25, 41
policy
 MessageArrival_610 325
privacy policy 374
Process group 149
 adding operations 167
 creating 164
 deleting 172
 editing 165
 removing operations 171
Process Groups attribute group 359
provider enter 73
provider leave 73
publications vii
 ordering ix

Q

queries, using attributes 355
Query Name attribute 356

R

reader requirements vii
Removing operations from a process group 171
Removing operations from a service group 166
Requester Identities for Operation workspace 24, 29, 50
Requester Identity Monitoring Configuration workspace 24, 49
Requester Identity Monitoring Status view 49
requester identity, monitoring by supported environments 47
requirements for readers vii
Response Time by Operation view 52

S

SCA component 85
scenarios, typical 9
Select Time Span 120
service x
Service Component Architecture mediation 72

Service Details view 58
Service group 149
 adding operations 166
 creating 164
 deleting 172
 editing 165
 removing operations 166
service management connect x
Service Port Details view 61
Service Registry Integration 53
service transaction flow 74
service-oriented architecture 1
service-to-service topology 71
Services Inventory view 41
Services Management Agent Environment workspace 24, 25, 40
Services Management Agent workspace 24, 36
Services Management workspace 24, 26, 53, 55
Services Overview view 56
SetReqIDTypeHostIP take action command 247
SetReqIDTypeUserInfo take action command 248
situation 199
 creating your own 213
 delta and percent functions 213
 Fault_610 201
 MaxMessageSize_610 202
 MaxResponseTimeCritical_610 203
 MaxResponseTimeWarning_610 204
 MessageArrivalClearing_610 205, 212
 MessageArrivalCritical_610 207
 MessageSize_610 209
 ResponseTimeCritical_610 210
 ResponseTimeWarning_610 211
situation, predefined 200
situations, using attributes 355
SMC x
SOA 1
SOA All Groups attribute group 360
SOA Domain Management Server 74
SOA Group Status attribute group 362
Static data 88
static topology 53
Status 151
Status attribute 362, 364
status, DataPower 77
status, operation aggregate 76
status, operation instance 74
Subnode Affinity attribute 359
Subnode MSN attribute 359
Subnode Resource Name attribute 360
Subnode Type attribute 360
Subnode Version attribute 360
subnode, deleting unmanaged 123
support x

T

Take Action command overview 223
Take Action commands
 AddFltrCntrl_610 226
 AddMntrCntrl_610 229
 AddRequesterIdentity_610 233

- Take Action commands (*continued*)
 - DeleteRequesterIdentity_610 235
 - DeleteSubnode 236
 - DelFltrCntl_610 238
 - DelMntrCntl_610 241
 - DisableDC_610 243
 - DisableReqIDMntr_610 244
 - EnableDC_610 245
 - EnableReqIDMntr_610 247
 - SetReqIDTypeHostIP 247
 - SetReqIDTypeUserInfo 248
 - updateLogging_610 249
 - updateTracing_610 251
 - UpdMntrCntl_610 252
- TBSM, integrating with 327
- Timestamp attribute 356, 359, 360, 363
- Tivoli Application Dependency Discovery Manager (TADDMM) , integrating with 328
- Tivoli Change and Configuration Management Database (CCMDB), integrating with 328
- Tivoli technical training x
- Tivoli user groups x
- topology
 - DataPower 128
 - DataPower, multiple display groups 129
 - DataPower, multiple domains 128
 - WebSphere Message Broker 130
- topology, service-to-service 71
- training, Tivoli technical x
- tuning 341
 - on-demand user overhead 351
- Type attribute 361, 363
- typeface conventions xi

U

- Unavailability 180
- unicode 257
- unmanaged client, deleting 125
- unmanaged object, deleting 123
- unmanaged operation instance, deleting 125
- updateLogging_610 take action command 249
- updateTracing_610 take action command 251
- UpdMntrCntl_610 take action command 252
- usage scenarios 9
- user groups, Tivoli x

V

- variables for directories xi
- view
 - Average Message Size by Operation 40, 41
 - Average Message Size by Service:Operation:Type 43
 - Average Response Time by Operation 40, 41
 - building your own 186
 - Business Process Details 66

- view (*continued*)
 - Business Processes for Service 63
 - Business Processes for Service Port 65
 - Fault Details 44
 - Fault Summary by Operation 41
 - historical data 32
 - Interaction 92
 - Message and Fault Count 51
 - Message Arrival by Operation 39
 - Message Arrival by Service 39
 - Message Arrival Details 39
 - Message Size by Operation 52
 - Monitored Requester Identities 50
 - Navigator 23
 - Number of Faults by Operation 44
 - Number of Messages by Operation 40
 - Number of Messages by Service:Operation:Type 42
 - Operation Flow - All Flows 92
 - Requester Identity Monitoring Status 49
 - Response Time by Operation 52
 - Service Details 58
 - Service Port Details 61
 - Services Inventory 41
 - Services Overview 56
- view, Navigator Physical 24
- Volume attribute 362, 364

W

- WebSphere Business Modeler, integrating with 328
- WebSphere Message Broker 1
- WebSphere Message Broker mediation 72
- WESB 1
- workflow policy 325
- workspace 24, 33
 - Application Server Services Management 24, 39, 40
 - DataPower Console 24, 46
 - Faults Summary 24, 44
 - Group Summary 24, 26
 - Message Arrival 24, 37
 - Message Summary 24, 42
 - Operational Flow 92
 - Operational Flow for Application Server 24, 91, 95
 - Operational Flow for Operation 24, 91, 93
 - Operational Flows 24, 26, 91
 - Performance Summary 24, 41
 - Performance Summary for Requester Identity 24
 - Requester Identities for Operation 24, 50
 - Requester Identity Monitoring Configuration 24, 49
 - Services Management 24, 26, 53, 55
 - Services Management Agent 24, 36
 - Services Management Agent Environment 24, 40
- workspace link, DataPower WebGUI console 31

- workspace link, from Situation Event Results 195
- workspace link, to a new workspace 187
- workspace link, to other products 29
- workspace, creating custom 185
- workspace, service-to-service topology 71
- WSRR, integrating with 327

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.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 <http://www.ibm.com/legal/copytrade.shtml>.

Adobe Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Intel is a trademark or registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, 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.

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



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

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 about 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.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites 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.

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.

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.

Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's user name for purposes of session management, authentication, and single sign-on configuration. These cookies cannot be disabled.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, See IBM's Privacy Policy at <http://www.ibm.com/privacy> and IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details> the section entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.



Printed in USA

SC23-8804-04

