

Administering webMethods Optimize

Version 9.8

April 2015

This document applies to webMethods Optimize Version 9.8 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2003-2015 Software AG, Darmstadt, Germany and/or Software AG USA Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

Table of Contents

About this Guide.....	21
Document Conventions.....	21
Online Information.....	22
For System Administrators.....	23
Concepts.....	25
Overview.....	26
webMethods Business Activity Monitoring (BAM) Components.....	26
Implementation.....	27
Data Collection.....	28
Data Communication.....	28
Data Processing and Analysis.....	29
Data Presentation.....	29
Database Components.....	30
Installation and Configuration Checklists.....	30
Data Collection.....	30
Infrastructure Data Collector.....	31
Web Service Data Collector.....	32
Key Performance Indicators (KPIs).....	34
KPI instances and KPIs.....	34
Business Processes.....	35
Rules.....	36
KPI Rules.....	36
Event and Threshold Rules.....	36
Users.....	37
Administrators.....	37
Business Analysts.....	37
Users, Groups, and Roles.....	37
Starting, Stopping, and Accessing Optimize.....	39
Starting Optimize Manually.....	40
Starting Optimize on a UNIX System.....	40
Starting an Analytic Engine.....	40
Starting My webMethods Server.....	40
Starting a Web Services Data Collector.....	40
Starting on Solaris 64-bit and HP-UX PA-RISC Platforms.....	41
Editing the Analytic Engine Script.....	41
Editing the Web Service Data Collector Script.....	41
Changing the Webmbam User when running Optimize on UNIX.....	41
Starting Optimize on a Windows System.....	42
Installing an Analytic Engine as a Windows Service.....	42
Uninstalling an Analytic Engine as a Windows Service.....	42

Installing a Web Service Data Collector as a Windows Service.....	43
Uninstalling a Web Service Data Collector as a Windows Service.....	43
Starting webMethods Components on a Windows System.....	43
Stopping Optimize.....	44
Stopping Analytic Engines.....	44
Stopping My webMethods Server.....	45
Stopping Web Service Data Collectors.....	45
Starting and Stopping Infrastructure Data Collector.....	46
Starting Infrastructure Data Collector.....	46
Stopping Infrastructure Data Collector.....	46
Managing Infrastructure Data Collector as a Service.....	47
Starting Infrastructure Data Collector as a Daemon in Linux.....	48
Viewing Information about Optimize Components.....	48
Viewing Information about Optimize-Related Products.....	50
Viewing Information About Jobs.....	50
Configuring Optimize.....	53
Configuring and Deploying Your Environment.....	54
Changing Your Configuration Settings.....	57
Configuring Rule Violation Alerts and Actions.....	57
Configuring E-mail Alerts.....	57
Specifying E-mail Alert Settings and Templates.....	58
Creating a Dynamic Alert Template.....	60
The Action Context.....	60
Example.....	62
Creating a Static Custom Alert Template.....	63
Example.....	64
Specifying Substitution Variables.....	64
Configuring SNMP Alerts.....	66
Configuring Web Service Actions.....	68
Changing Subcomponent Communication Settings.....	68
Changing Analytic Engine or Web Service Data Collector Port Settings.....	68
Changing Analytic Engine or Web Service Data Collector Attributes for Web Services.....	69
Changing My webMethods Host and Port Communication Attributes.....	69
Importing a Certification Authority (CA) Certificate.....	70
Defining Logical Names for the Analytic Engines and Web Service Data Collectors.....	70
Securing Optimize Engines with SSL.....	71
Configuring Optimize Web Services for SSL.....	71
Changing the Configuration Agent Protocols.....	72
Securing the Optimize Analytic Engine Web Services.....	73
Securing the Optimize Web Service Data Collector Services.....	73
Configuring the Optimize Engines to Communicate with a Secured Broker Server.....	74
Updating the Configuration Settings in the Central Configurator.....	75

Enabling Data-Level Security.....	75
Changing the Optimize Authentication Credentials.....	76
Adding New Optimize Credentials.....	77
Specifying Statistical Intervals.....	78
Optimizing Analytic Engine Performance.....	79
Multi-NIC System Considerations.....	83
The Java Service Wrapper.....	84
The Java Service Wrapper Configuration Files.....	84
JVM Configuration.....	85
JVM Configuration Properties.....	85
Specifying the JDK or JRE for Analytic Engine, Infrastructure Data Collector and Web Service Data Collector.....	86
Configuring JVM Memory Allocation.....	86
The Wrapper Log.....	87
Logging Properties.....	87
Configuring Additional JVM Parameters.....	88
Configuring Analytic Engine High Availability.....	89
Optimizing Web Service Data Collector Performance.....	90
Connecting Process Engines to Analytic Engines.....	91
Adjusting RAM for the Analytic Engine and Web Service Data Collectors.....	91
Configuring Optimize to use an Analytic Engine.....	92
Configuring Optimize for Analytic Engine Clustering.....	93
Configuring Optimize to Monitor Multiple Integration Servers and Analytic Engines.....	93
Configuring EDA/NERV Settings for Optimize.....	94
Guaranteed Event Storage in an Optimize Environment.....	94
Implementing Guaranteed Event Storage in Optimize.....	95
Configuring the MAPI Flag.....	95
Disabling the MAPI Flag.....	96
Setting System Cache Properties.....	96
Disabling Disk Caching.....	98
Configuring the PPM Server Settings.....	98
Configuring Business Processes.....	99
Overview.....	100
Preparing to Monitor Processes.....	100
Setting Up a webMethods-Executed Process.....	100
Setting Up an Externally Executed Process.....	101
Identifying a Process to Monitor.....	101
Editing a Process.....	103
Instance Analytics Tab.....	105
Standardized Error Types.....	107
Cycle-Time Standardized Error Types.....	108
Wait-Time Standardized Error Types.....	110
Iterations Standardized Error Types.....	113
Deadline Standardized Error Types.....	115
Editing an Existing Standardized Error Type.....	118

Deleting a Previously Defined Standardized Error Type.....	118
Custom Error Types.....	119
Working with Stages and Milestones.....	120
Adding a Stage.....	120
Modifying a Stage.....	122
Deleting a Stage.....	122
Viewing Stages in the Process Diagram.....	123
Enabling and Disabling EDA Event Emission.....	123
About Synchronizing Stages and Events with Software AG Designer.....	125
Working with the Overview Configuration Page.....	125
Enabling and Disabling Configurations for Viewing on the Analytics Overview Page.....	126
Creating, Editing, Copying, and Deleting Configurations.....	127
Deleting a Configuration.....	129
Copying a Configuration.....	129
Working with Configuration Components.....	129
Adding a Folder to the Component Tree.....	130
Adding a Process, Stage or Process Step to the Component Tree.....	131
Adding a Stage to a Process on the Components Tress.....	131
Adding a Process Step to the Components Tree.....	132
Adding a KPI or KPI Filter to the Component Tree.....	132
Adding a KPI Filter to the Component Tree.....	133
Editing Components on the Component Tree.....	133
Rearranging Nodes in the Component Tree by Cutting and Pasting.....	134
Rearranging Nodes in the Component Tree by Dragging and Dropping.....	134
Sorting a Node's Children or Descendants in the Component Tree.....	135
Deleting Nodes from the Component Tree.....	135
Granting User Access to Business Processes.....	136
Granting Access to Business Processes.....	137
Configuring KPIs.....	139
Overview.....	140
Configuring Business Data.....	140
Configuring System Data.....	142
Defining Dimensions.....	143
Viewing Existing Dimensions.....	144
Adding a New Dimension.....	145
Editing an Existing Dimension.....	147
Deleting Dimensions.....	149
Mapping Events.....	150
Adding a New Event Map.....	150
Adding an Event Map with New Fields.....	156
Adding an Event Map that Combines Business Data with Process and Step Metrics.....	160
Editing an Existing Event Map.....	162
Deleting an Event Map.....	164

Defining Dimension Hierarchies.....	164
Viewing Existing Dimension Hierarchies.....	165
Adding or Editing a Dimension Hierarchy.....	165
Deleting Dimension Hierarchies.....	166
Defining KPIs.....	167
Adding Individual KPIs.....	169
Adding Composite KPIs.....	171
KPI Back Population.....	172
Intelligent Links.....	173
Adding Intelligent Links.....	174
Editing Intelligent Links.....	175
Deleting Intelligent Links.....	176
Subscribing to webMethods Events.....	176
Filtering the List of Subscribed Event Types.....	176
Subscribing to Event Types.....	177
Granting User Access to KPIs.....	177
Granting Access to KPIs.....	178
Deleting Corrupted KPI Definitions.....	179
Extracting Optimize Assets from My webMethods Server.....	179
Configuring and Using the Infrastructure Data Collector.....	181
Overview.....	182
Monitoring Infrastructure Data Collector.....	182
Configuring Integration Servers and webMethods Brokers.....	182
Configuring IS Packages.....	182
WmlSExtDC.....	182
WmARTEExtDC.....	184
Configuring the SNMP Data Collector.....	184
Creating and Defining the MIB Configuration File.....	185
Specifying the Devices to Manage.....	189
Changing Infrastructure Data Collector Service Name.....	190
Changing Infrastructure Data Collector Ports.....	190
Changing the Infrastructure Data Collector Primary Port.....	191
Changing the Infrastructure Data Collector Configuration Port.....	191
Defining HTTPS Connectors.....	192
Server-side Configuration.....	192
Client-side Configuration.....	192
Configuring Infrastructure Data Collector for SSL.....	193
Discovering Assets and Monitoring Components.....	195
Overview.....	196
Interaction between Infrastructure Data Collector and Optimize.....	196
Scaling Your Architecture.....	197
Infrastructure Data Collector.....	198
Optimize.....	199
Discovering Assets.....	199
The Discovery Page.....	200

Add Asset.....	204
Add Network.....	212
Editing the Config.ini file for Universal Messaging SSL Configuration.....	216
The Assets Page.....	216
Editing Connection Parameters.....	218
Deleting SNMP Component Assets.....	223
Suspending Monitoring.....	224
Resuming Monitoring.....	225
Monitoring Components.....	225
Displaying a List of Monitored Components.....	226
Configuring Monitor Components.....	228
Using the Monitor and Suspend Services.....	230
Setting Up Logging for Optimize.....	231
Overview.....	232
Setting Up Logging for Analytic Engines and Data Collectors.....	232
Setting Up Logging for Infrastructure Data Collectors.....	233
Installing and Configuring the WmOptimize Package.....	235
Installing and Configuring the WmOptimize Package.....	236
Enabling the WmOptimize Package.....	236
Configuring the WmOptimize Package.....	236
Adding Optimize Users to the OptimizeUsers Group.....	240
Integration with Software AG MashZone.....	241
Overview.....	242
Displaying Data from Optimize Analytic Engine into Software AG MashZone.....	242
Constructing Queries to Display Data from Optimize Analytic Engine into Software AG MashZone.....	244
General Purpose Parameters.....	244
KPI Instance Readings Parameters.....	245
Dimension Instances Parameters.....	247
KPI Definitions Parameters.....	248
Problems Parameters.....	248
For Developers.....	251
Configuring and Using the Web Service Data Collector.....	253
Overview.....	254
Configuring the Web Service Data Collector.....	254
Generating the Client.....	255
Sending Event and Dimensional Data.....	255
addEvent (event) and addEvents (events).....	257
addEventWithCallback.....	258
addStronglyTypedEvent (event) and addStronglyTypedEvents (events).....	259
addDimension (dimension) and addDimensions (dimensions).....	260
addStronglyTypedDimension (dimension) and addStronglyTypedDimensions (dimensions).....	261
Sending Process Data.....	261
pushProcessControlOperation (ProcessControlOperation).....	262

pushProcessControlOperationWithCallback.....	265
pushStepControlOperation (StepControlOperation).....	269
pushStepControlOperationWithCallback.....	273
pushExecutionError (ExecutionError).....	277
pushExecutionErrorWithCallback.....	278
Setting Up Web Service Actions.....	281
Overview.....	282
Configuring Web Service Actions.....	282
Optimize REST Services API.....	287
Overview.....	288
Time Parameters.....	288
Max Results Parameter.....	288
HTTP Request Errors.....	289
Error Response Structure.....	289
Error Response Example.....	289
Statistical Results.....	289
Running the Restful Services using localhost.....	290
The Optimize Restful Services.....	290
Process Model Rest API Services.....	290
Process Model List.....	291
Input Parameters.....	291
Output Parameters.....	291
.....	
Process Models Result Example.....	292
Process Steps.....	292
Input Parameters.....	292
Output Parameters.....	293
.....	
Step Models Result Example.....	293
Process Stages.....	293
Input Parameters.....	294
Output Parameters.....	294
.....	
Stage Models Result Example.....	294
Process Analytics REST API.....	295
Process Metric Data.....	295
Input Parameters.....	295
Output Parameters.....	296
.....	
Process Metrics Result Example.....	297
Process Instances.....	297

Input Parameters.....	298
Output Parameters.....	298
.....	
Process Instances Result Example.....	299
Process Tracking Errors.....	299
Input Parameters.....	300
Output Parameters.....	300
.....	
Process Tracking Errors Result Example.....	300
Step Instances.....	300
Input Parameters.....	301
Output Parameters.....	301
.....	
Step Instances Result Example.....	302
Stage Metric Data.....	303
Input Parameters.....	303
Output Parameters.....	304
.....	
Stage Metrics Result Example.....	304
Stage Volumes.....	305
Input Parameters.....	305
Output Parameters.....	305
.....	
Stage Volume Result Example.....	306
Stage Tracking Errors.....	306
Input Parameters.....	306
Output Parameters.....	307
.....	
Stage Errors Result Example.....	307
Stage Cycle Times.....	307
Input Parameters.....	308
Output Parameters.....	308
.....	
Stage Cycle Times Result Example.....	308
Stage Instances.....	309
Input Parameters.....	309
Output Parameters.....	310
.....	

Stage Instances Result Example.....	310
Stage Instances for a Process Instance.....	311
Input Parameters.....	311
Output Parameters.....	311
.....	
Stage Instances Result Example.....	312
Active Process Instance Count for a Point in Time.....	312
Input Parameters.....	312
Output Parameters.....	313
.....	
Process Count Result Example.....	313
Active Process Instance Count for Time Range.....	313
Input Parameters.....	314
Output Parameters.....	314
.....	
Process Count Result Example.....	314
For Database Administrators.....	315
Installing Optimize Database Component Sets.....	317
Overview.....	318
Database Component Sets.....	318
Order of Database Component Installation.....	318
Configuring Optimize Database Pools.....	321
Overview.....	322
Concepts.....	322
Multiple Analytic Engines.....	324
Installing Optimize Database Component Sets.....	324
Defining Database Connection Pools.....	325
Mapping Database Connection Pools to Optimize Activities.....	326
Purging Data from Optimize Database Components.....	329
Overview.....	330
Data Purge.....	330
Installing Data Purge.....	330
Required Permissions for Oracle Database Users.....	330
Running Data Purge.....	331
Oracle.....	331
SQL Server.....	331
DB2.....	332
Resetting Referential Constraints if Data Purge Fails.....	332
Configuring Data Purge.....	333
Setting Data Purge Attributes for the Analytic Engine.....	333

Setting Purge Operation Parameters.....	335
Specifying Information to Log.....	338
Specifying People to Notify about the Oracle Purge Operation.....	338
Viewing Purge Operation Events.....	338
Live Purge.....	340
Configuring Live Purge.....	341
Running Live Purge.....	344
Stopping Live Purge.....	344
Viewing Live Purge Results.....	345
Using the Live Purge Standalone Tool.....	347
Prerequisites.....	347
Starting the Live Purge Standalone Tool.....	347
Stopping the Live Purge Standalone Tool.....	348
Monitored KPI Definitions.....	349
Overview.....	350
Universal Messaging KPI Definitions.....	350
com.softwareag.um.realm.node.....	350
Discovery Mechanism.....	351
KPIs.....	351
com.softwareag.um.realm.channel.....	353
Discovery Mechanism.....	353
KPIs.....	353
com.softwareag.um.realm.queue.....	355
Discovery Mechanism.....	355
KPIs.....	355
com.softwareag.um.realm.datagroup.....	356
Discovery Mechanism.....	357
KPIs.....	357
com.softwareag.um.realm.interface.....	357
Discovery Mechanism.....	358
KPIs.....	358
com.softwareag.um.realm.threadpool.....	361
Discovery Mechanism.....	361
KPIs.....	361
Broker KPI Definitions.....	361
wm.brokerdc.BrokerServer.....	362
Discovery Mechanism.....	362
KPIs.....	362
wm.brokerdc.Broker.....	366
Discovery Mechanism.....	366
KPIs.....	366
wm.brokerdc.DocumentType.....	369
Discovery Mechanism.....	369
KPIs.....	370

Territory KPI Definitions.....	371
KPIs.....	371
Gateway KPI Definitions.....	373
Discovery Mechanism.....	373
KPIs.....	373
Integration Server KPI Definitions.....	375
com.wm.isextdc.IntegrationServer.....	376
Discovery Mechanism.....	376
KPIs.....	377
com.wm.isextdc.IsPackage.....	379
Discovery Mechanism.....	379
KPIs.....	379
com.wm.isextdc.IsService.....	380
Discovery Mechanism.....	380
KPIs.....	380
com.wm.isextdc.IsFtpPort.....	383
Discovery Mechanism.....	383
KPIs.....	383
com.wm.isextdc.IsSocketPort.....	384
Discovery Mechanism.....	384
KPIs.....	384
com.wm.isextdc.IsHttpPort.....	385
Discovery Mechanism.....	385
KPIs.....	386
com.wm.isextdc.IsHttpsPort.....	387
Discovery Mechanism.....	387
KPIs.....	387
com.wm.isextdc.IsEmailPort.....	388
Discovery Mechanism.....	388
KPIs.....	388
com.wm.isextdc.RepoV2.....	389
Discovery Mechanism.....	389
KPIs.....	390
com.wm.isextdc.IsBrokerTransport.....	390
Discovery Mechanism.....	391
KPIs.....	391
com.wm.isextdc.IsScheduler.....	392
Discovery Mechanism.....	392
KPIs.....	392
com.wm.isextdc.IsSchedulerTask.....	392
Discovery Mechanism.....	393
KPIs.....	393
com.wm.isextdc.IsCluster.....	394
Discovery Mechanism.....	395
KPIs.....	395

com.wm.isextdc.IsClusterServer.....	395
Discovery Mechanism.....	395
KPIs.....	395
com.wm.isextdc.isJMSTrigger.....	396
Discovery Mechanism.....	396
KPIs.....	397
com.wm.isextdc.isNativeTrigger.....	397
Discovery Mechanism.....	397
KPIs.....	397
com.wm.isextdc.IsJDBCFunctionalAlias.....	398
Discovery Mechanism.....	399
KPIs.....	399
webMethods Trading Networks KPI Definitions.....	400
com.wm.tnextdc.TNServer.....	400
Discovery Mechanism.....	401
KPIs.....	401
com.wm.tnextdc.TaskManager.....	405
Discovery Mechanism.....	405
KPIs.....	405
com.wm.tnextdc.TNDBConnectionPool.....	406
Discovery Mechanism.....	406
KPIs.....	407
My webMethods Server KPI Definitions.....	408
com.softwareag.mws.....	408
Discovery Mechanism.....	409
KPIs.....	409
com.softwareag.mws.directory.....	412
Discovery Mechanism.....	412
KPIs.....	412
com.softwareag.mws.handler.sessionmanager.....	413
Discovery Mechanism.....	413
KPIs.....	413
com.softwareag.mws.handler.authmanager.....	414
Discovery Mechanism.....	414
KPIs.....	414
com.softwareag.mws.handler.presmanager.....	415
Discovery Mechanism.....	415
KPIs.....	415
com.softwareag.mws.handler.dispatchmanager.....	416
Discovery Mechanism.....	416
KPIs.....	416
com.softwareag.mws.taskengine.....	417
Discovery Mechanism.....	417
KPIs.....	417
com.softwareag.mws.taskevents.....	418

Discovery Mechanism.....	418
KPIs.....	418
com.softwareag.mws.portlet.action.....	419
Discovery Mechanism.....	419
KPIs.....	419
com.softwareag.mws.portlet.render.....	419
Discovery Mechanism.....	420
KPIs.....	420
com.softwareag.mws.portlet.pcalayout.....	420
Discovery Mechanism.....	421
KPIs.....	421
com.softwareag.mws.cache.....	421
Discovery Mechanism.....	422
KPIs.....	422
com.softwareag.mws.dirservicecache.....	423
Discovery Mechanism.....	423
KPIs.....	423
com.softwareag.mws.connectionpool.....	424
Discovery Mechanism.....	424
KPIs.....	424
com.softwareag.mws.database.....	425
Discovery Mechanism.....	425
KPIs.....	426
Infrastructure Data Collector KPI Definitions.....	426
wm.infrastructuredc.InfrastructureDataCollector.....	426
Discovery Mechanism.....	426
Adapter KPI Definitions.....	426
wm.brokerdc.CustomAdapter.....	427
Discovery Mechanism.....	427
KPIs.....	427
com.wm.isextdc.AdapterService.....	429
Discovery Mechanism.....	429
KPIs.....	429
com.wm.artextdc.ARTConnection.....	432
Discovery Mechanism.....	432
KPIs.....	433
com.wm.artextdc.ARTListener.....	434
Discovery Mechanism.....	434
KPIs.....	434
com.wm.artextdc.ARTListenerNotification.....	435
Discovery Mechanism.....	435
KPIs.....	435
com.wm.artextdc.ARTPollingNotification.....	438
Discovery Mechanism.....	438
KPIs.....	438

Terracotta KPI Definitions.....	440
com.softwareag.tc.connectiongroup.....	440
Discovery Mechanism.....	440
KPIs.....	440
com.softwareag.tc.server.....	440
Discovery Mechanism.....	441
KPIs.....	441
Terracotta Connection.....	443
Ehcache.....	444
TSA.....	444
TSA Server.....	445
CacheManager.....	446
Cache.....	448
Presto KPI Definitions.....	454
com.softwareag.presto.....	454
Discovery Mechanism.....	454
KPIs.....	455
Built-In Rules.....	457
Overview.....	458
Infrastructure Data Collector Rules.....	458
Infrastructure Data Collector About to Run Out of Memory.....	458
Infrastructure Data Collector Thread Count Trend.....	458
Infrastructure Data Collector Failure Status.....	459
ART Rules.....	459
ART Adapter Service Critical Resource Message.....	459
ART Adapter Service Failure Status.....	460
ART Connection Critical Resource Message.....	460
ART Connection Failure Status.....	460
ART Connection Resource Failure.....	461
ART Listener Critical Resource Message.....	461
ART Listener Failure Status.....	462
ART Listener Notification Critical Resource Message.....	462
ART Listener Notification Failure Status.....	463
ART Polling Notification Critical Resource Message.....	463
ART Polling Notification Failure Status.....	463
Broker Server Rules.....	464
Broker Server Critical Resource Message.....	464
Broker Server Failure Status.....	464
Broker Server Low Guaranteed Disk Space.....	465
Broker Rules.....	465
Broker Critical Resource Message.....	465
Broker Failure Status.....	466
Document Type Notification Critical Resource Message.....	466
Custom Adapter Rules.....	467

Custom Adapter Failure Status.....	467
Territory Rules.....	467
Territory Critical Resource Message.....	467
Territory Failure Status.....	468
Gateway Rules.....	468
Gateway Failure Status.....	468
Analytic Engine Rules.....	468
Analytic Engine About to Run Out of Memory.....	468
Error in Analytic Engine.....	469
Thread Leak.....	469
Integration Server Rules.....	470
Integration Server Completed Requests Abnormal.....	470
Integration Server Critical Resource Message.....	470
Integration Server Failure Status.....	471
Integration Server Used Memory Too Big.....	471
IsBrokerTransport Critical Resource Message.....	471
IS Broker Transport Failure Status.....	472
IS Cluster Critical Resource Message.....	472
IS Cluster Failure Status.....	473
IS Email Port Critical Resource Message.....	473
IS Email Port Failure Status.....	474
IS FTP Port Critical Resource Message.....	474
IS FTP Port Failure Status.....	474
IS HTTP Port Critical Resource Message.....	475
IS HTTP Port Failure Status.....	475
IS HTTPS Port Critical Resource Message.....	476
IS HTTPS Port Failure Status.....	476
IS Socket Port Critical Resource Message.....	476
IS Socket Port Failure Status.....	477
Repository Server Critical Resource Message.....	477
Repository Server Failure Status.....	478
IS Package Rules.....	478
Integration Server Package Down.....	478
Integration Server Package Failure Status.....	479
IS Package Critical Resource Message.....	479
IS Service Rules.....	479
Integration Server Service Failure Status.....	479
Integration Server Service Request Time Trend.....	480
IS Service Critical Resource Message.....	480
My webMethods Server Rules.....	481
DirServiceCache authenticateCacheSize High.....	481
DirServiceCache dnCacheSizeHigh.....	481
DirServiceCache Failure Status.....	482
DirServiceCache uriCacheSize High.....	482
Handler.authManager Failure Status.....	483

Handler.dispatchManager Failure Status.....	483
Handler.presManager Failure Status.....	484
Handler.sessionManager Failure Status.....	484
MWSCache cacheSize High.....	485
MWSCache Failure Status.....	485
MWSConnectionPool Closed.....	485
MWSConnectionPool Failure Status.....	486
MWSConnectionPool poolSize High.....	486
MWSDatabase FailureStatus.....	487
MWSDirectory FailureStatus.....	487
MWSTaskEngine FailureStatus.....	488
MWSTaskEvents FailureStatus.....	488
MywebMethodsServer FailureStatus.....	488
MywebMethodsServer HTTP_Request averageTime Trending Up.....	489
MywebMethodsServer Memory.freeMemory low.....	489
Portlet.Action Failure Status.....	490
Portlet.PCALayout Failure Status.....	490
Portlet.Render Failure Status.....	490
Trading Networks Rules.....	491
TN Database Connection Pool Failure Status.....	491
TN DB Connection Pool Critical Resource Message.....	491
TN Server Critical Resource Message.....	492
TN Server Failure Status.....	492
TN Server Used Threads Too Big.....	493
TN Task Manager Critical Resource Message.....	493
TN Task Manager Failure Status.....	493
Terracotta Rules.....	494
Terracotta Cache Failure Status.....	494
Terracotta Cache Manager Failure Status.....	494
Terracotta Connection Failure Status.....	495
TSA Server Failure Status.....	495
TSA Failure Status.....	496
Presto Rules.....	496
Presto Failure Status.....	496
Universal Messaging Rules.....	497
Realm Channel Failure Status.....	497
Realm Datagroup Failure Status.....	497
Realm Failure Status.....	498
Realm Interface Failure Status.....	498
Realm Queue Failure Status.....	498
Realm Threadpool Failure Status.....	499
Built-In Services.....	501
Overview.....	502
Summary of Elements in This Folder.....	502

pub.optimize.monitoring:addEvent (event).....	503
Input Parameters.....	504
Output Parameters.....	505
pub.optimize.monitoring:addEvents (events).....	505
Input Parameters.....	505
Output Parameters.....	506
pub.optimize.monitoring:addDimension (dimension).....	506
Input Parameters.....	506
Output Parameters.....	507
pub.optimize.monitoring:addDimensions (dimensions).....	507
Input Parameters.....	507
Output Parameters.....	508
pub.optimize.monitoring:pushExecutionError (ExecutionError).....	508
Input Parameters.....	508
Output Parameters.....	510
pub.optimize.monitoring:pushProcessControlOperation (ProcessControlOperation).....	510
Input Parameters.....	510
Output Parameters.....	514
pub.optimize.monitoring:pushStepControlOperation (StepControlOperation).....	514
Input Parameters.....	514
Output Parameters.....	518
pub.optimize.monitoring:createNewConnection.....	518
Input Parameters.....	518
Output Parameters.....	519
pub.optimize.monitoring:destroyConnection.....	519
Input Parameters.....	519
Output Parameters.....	520
pub.optimize.mashzone.adapter:retrieveData.....	520
Input Parameters.....	520
Output Parameters.....	521
Optimize Schema Overview.....	523
Understanding the Optimize Schema.....	524
Table Naming and Versioning.....	525
Fact Tables (BAM_FACT_<Event Map Name>_V1).....	525
Dimension Tables (BAM_DIM_<Dimension Name>_V1).....	526
Glossary.....	527
Index.....	541

About this Guide

This guide explains how to use webMethods Optimize to monitor resource availability and events, establish thresholds, and respond to alerts. Part I contains information of interest to all users, and Part II contains information primarily of interest to business users.

This guide is for administrators who will configure Optimize to work with your company's systems and business processes, and for developers who will develop custom interfaces for Optimize. Part I contains information primarily of interest to system administrators, Part II contains information of interest to developers, and Part III contains information of interest to database administrators.

Note that this guide covers both Optimize for Process and Optimize for Infrastructure. Information that is specific to each version of the product is called out within the chapters as appropriate.

To use this guide effectively, administrators and developers should be experienced in using advanced software applications for system administration and should have a basic understanding of XML.

Important: Values entered into XML files are case sensitive, so they must be entered exactly as represented in this document.

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Narrowfont	Identifies storage locations for services on webMethods Integration Server, using the convention <i>folder.subfolder:service</i> .
UPPERCASE	Identifies keyboard keys. Keys you must press simultaneously are joined with a plus sign (+).
<i>Italic</i>	Identifies variables for which you must supply values specific to your own situation or environment. Identifies new terms the first time they occur in the text.
Monospace font	Identifies text you must type or messages displayed by the system.

Convention	Description
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the symbol.
[]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at <http://documentation.softwareag.com>. The site requires Empower credentials. If you do not have Empower credentials, you must use the TECHcommunity website.

Software AG Empower Product Support Website

You can find product information on the Software AG Empower Product Support website at <https://empower.softwareag.com>.

To submit feature/enhancement requests, get information about product availability, and download products, go to [Products](#).

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the [Knowledge Center](#).

Software AG TECHcommunity

You can find documentation and other technical information on the Software AG TECHcommunity website at <http://techcommunity.softwareag.com>. You can:

- Access product documentation, if you have TECHcommunity credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.
- Access articles, code samples, demos, and tutorials.
- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.
- Link to external websites that discuss open standards and web technology.

I For System Administrators

■ Concepts	25
■ Starting, Stopping, and Accessing Optimize	39
■ Configuring Optimize	53
■ Configuring Business Processes	99
■ Configuring KPIs	139
■ Configuring and Using the Infrastructure Data Collector	181
■ Discovering Assets and Monitoring Components	195
■ Setting Up Logging for Optimize	231
■ Installing and Configuring the WmOptimize Package	235
■ Integration with Software AG MashZone	241

The chapters in this part of the guide contain information about tasks primarily performed by Optimize system administrators.

1 Concepts

■ Overview	26
■ webMethods Business Activity Monitoring (BAM) Components	26
■ Implementation	27
■ Installation and Configuration Checklists	30
■ Data Collection	30
■ Infrastructure Data Collector	31
■ Web Service Data Collector	32
■ Key Performance Indicators (KPIs)	34
■ Business Processes	35
■ Rules	36
■ Users	37

Overview

This chapter explains the components and concepts behind webMethods Optimize business activity monitoring. It describes the architecture of the Optimize implementation, and the processes by which data is collected and fed into the Optimize application. In addition, it discusses the interaction between Key Performance Indicators (KPIs), operational data, business processes, and rules in analyzing and displaying collected data. Finally, it describes the tasks that system and business users perform and the roles that users can be assigned.

webMethods Business Activity Monitoring (BAM) Components

Optimize supports various levels of Business Activity Monitoring (BAM) depending on customer needs. Business activity monitoring enables you to analyze real time business metrics information, including system performance, the volume of business activity and its responsiveness, serious errors that may have occurred, and other key performance indicators (KPIs). Aggregate analytics working in parallel with instance-level data provide actionable information that you can use to eliminate problems, increase efficiency, and take advantage of business opportunities.

Optimize uses special data collectors to gather data from other webMethods components that is used to support business activity monitoring. Note that there is no specific Optimize interface; rather, Optimize is an implementation of webMethods functionality that is available through the standard My webMethods user interface.

Two different implementations of Optimize are available to webMethods customers: Optimize for Infrastructure and Optimize for Process. These implementations are explained below.

■ Optimize for Infrastructure

This implementation enables you to monitor system data in real time. System data consists of data about equipment and applications, such as queue length and whether a managed webMethods component is online or offline. You can monitor individual objects as well as the overall status of your system. Optimize for Infrastructure contains two components: Infrastructure Data Collector and Optimize. Optimize for Infrastructure uses the Infrastructure Data Collector to collect system data.

■ Optimize for Process.

This implementation enables you to monitor business processes, in addition to managed objects and system status. You can track key performance indicators (KPIs) that relate to operational factors such as margin, revenue, customer satisfaction, and inventory levels. Optimize enables you to compare actual performance with rules you establish. If a monitored item goes out of compliance with a rule, Optimize alerts you so that you can address the problem. In addition, you can use Six Sigma to analyze your business processes. Six Sigma is a disciplined methodology for improving business process performance by eliminating defects.

Using this methodology, you can measure the number of defects in a process and systematically determine how to eliminate them. Optimize for Process also includes business visualization tools that provide at-a-glance mashups of Optimize monitoring information.

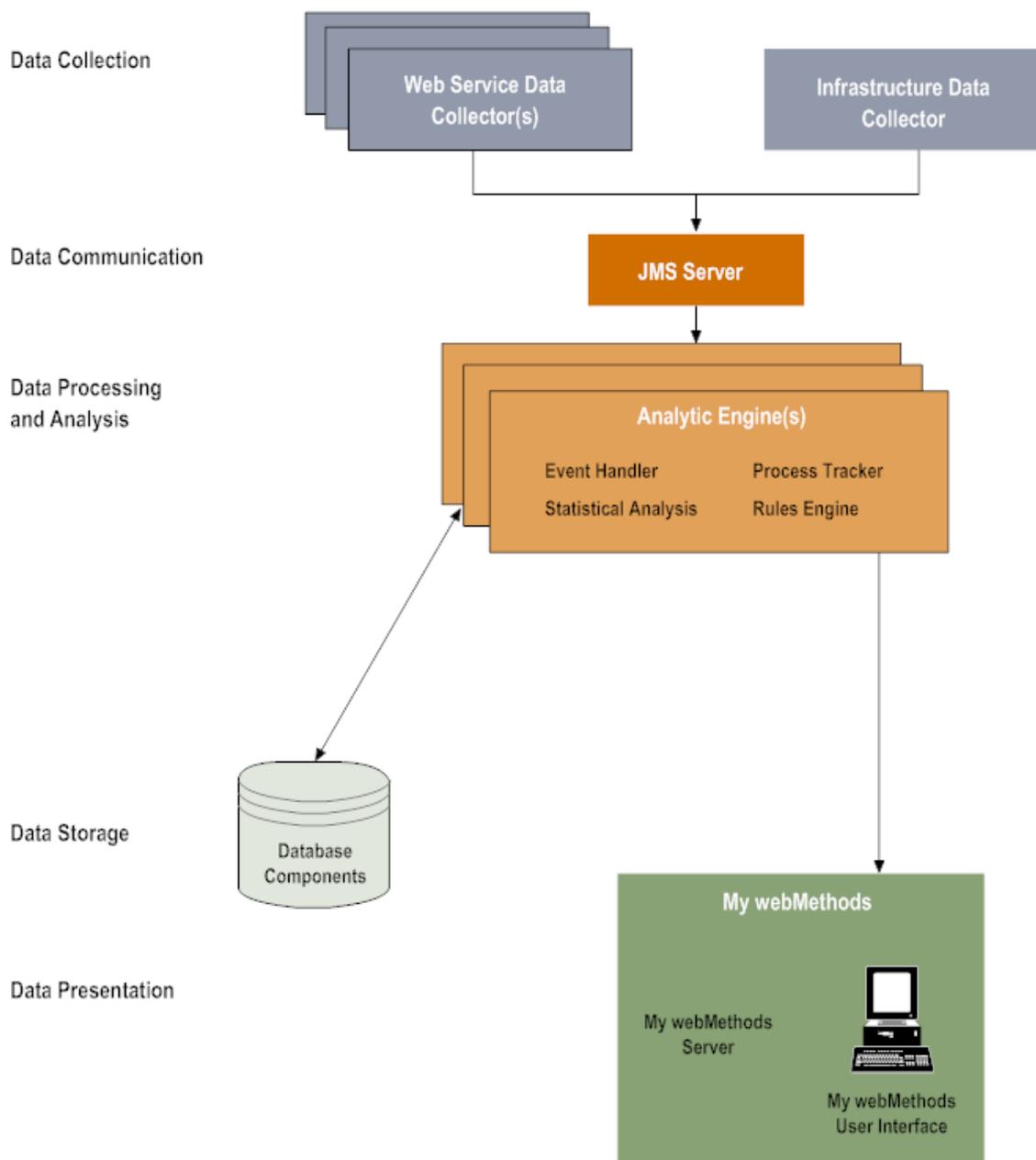
Implementation

Optimize implementations perform the following five main functions:

Function	Subcomponent
Data collection	One or more Web Service Data Collectors Infrastructure Data Collector, containing SNMP data collector
Data communication	A JMS Server, usually either webMethods Universal Messaging or webMethods Broker
Data processing and analysis	One or more Analytic Engines
Data storage	Database components
Data presentation	My webMethods user interface

Each subcomponent can reside on its own separate host, or all subcomponents can reside on the same host.

Typical Optimize implementation



Data Collection

Optimize collects data from other webMethods applications, through the Infrastructure Data Collector, or through the Web Service Data Collector(s). For more information about data collection, see "[Data Collection](#)" on page 30.

Data Communication

Both webMethods Universal Messaging and webMethods Broker provide a Java Message Service (JMS) communication layer that allows data collected by the Data

Collector to be fed into the Optimize Analytic Engine for processing. For more information about webMethods Universal Messaging, see the Terracotta website. For more information about webMethods Broker, see the *webMethods Broker Client Java API Programmer's Guide*.

Data Processing and Analysis

Optimize processes data by way of one or more *Analytic Engines*. The Analytic Engine receives business, system, and process data from the data collection engine, and saves the data in the databases. Optimize analyzes data, saves the analysis in the database, sends information about the status of business and system activity to the My webMethods user interface, and performs other actions when problems occur, such as sending alerts to specified users.

An Analytic Engine contains the following things:

- *An event handler*, which receives business and system data from the Infrastructure Data Collector and Web Service Data Collector and from other webMethods products. Business data consists of the underlying data that make up a business process, such as revenue and order number and is specific to Optimize for Process. System data consists of data about equipment or applications, such as queue length and whether a managed component is online or offline. The analytic engine writes this data to the database in a format that business intelligence software can use to generate reports.
- *A process tracker*, which receives business process data from the Web Service Data Collectors and other webMethods products, and tracks the state of executing processes. As a process executes, the process tracker takes measurements about the process instance and its steps, such as wait time and cycle time. This information can be used to track performance of a business process or to alert you when certain conditions exist, such as performance degradation or a failure in process execution.
- *An analysis engine*, which runs algorithms on the data.
- *A rules engine* which compares the data it receives from the Analytic Engine to rules that detect when a process or resource requires user attention.
- *A Web services layer*, which sends the results of the analysis to the My webMethods Server. The My webMethods user interface presents the results, along with icons that indicate that a rule violation has occurred or that Optimize has diagnosed a trend.

Data Presentation

Optimize presents data by way of the My webMethods user interface. My webMethods enables users to log in to all webMethods components that are incorporated into My webMethods using a single user name and password.

In addition to presenting data, the My webMethods user interface provides access to administrative functions. Using the My webMethods administrative functions, Optimize administrators can create users, groups, and roles; grant access to My webMethods functions using Lightweight Directory Access Protocol (LDAP) or other means; and configure system, business, and business process data. Optimize users can use other

functional areas of My webMethods to create rules for comparing data, monitoring business and system data, and tracking business processes.

For more information about the My webMethods user interface, see *Working with My webMethods*.

Database Components

Optimize stores data in relational databases. The Optimize database consists of four components: the Analysis, Process Tracker, Process Audit Log (PAL), and My webMethods Server (MWS) database components. For more information about database components, see "[For Database Administrators](#)" on page 315 in this guide.

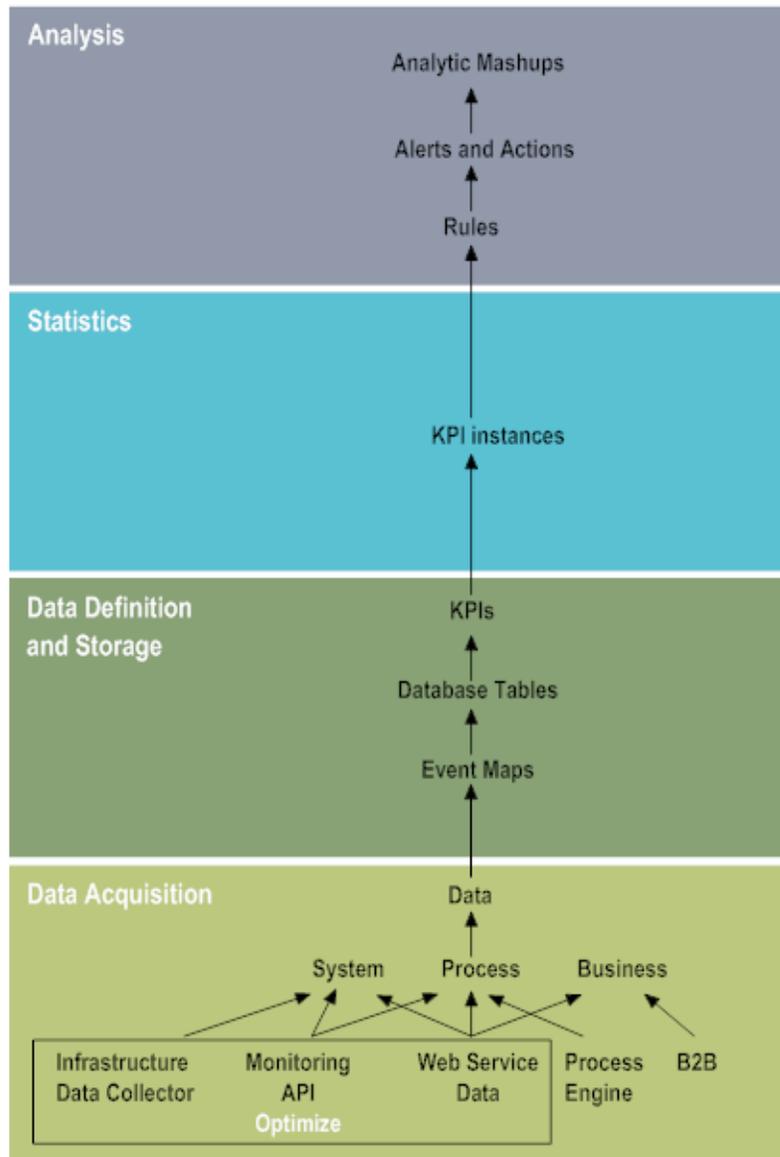
Installation and Configuration Checklists

The Installation and Configuration checklists have been moved to Appendix A in *Configuring BAM*.

Data Collection

Data collected from many parts of the enterprise infrastructure is fed into the Optimize environment and stored. KPIs that represent information vital to the business are defined and applied to the data. These KPIs are analyzed in real time in order to learn normal behavior and detect abnormal behavior within the environment. Rules are applied to these KPIs and their analyses in order to define conditions that represent problems and opportunities. Violations of these rules can be reported to the appropriate people and automated corrective actions can be taken. Further detail can be provided about a particular violation to determine the true root cause of the situation. Once Optimize has learned enough about the business, it can predict the occurrence of problems and thus allow staff to correct the problems even before they occur. Optimize also allows you to correlate information from disparate parts of the enterprise, such as the system and infrastructure, business processes and business data, and services. In addition, it provides analytic mashups, which are graphical overlays of correlated data such as business process information overlaid with real-time statistics.

Optimize collects data from other webMethods applications, through the Infrastructure Data Collector, or through the Web Service Data Collector. The following diagram illustrates how Optimize collects, defines, stores, and analyzes data.



Infrastructure Data Collector

The Infrastructure Data Collector is the primary method for collecting infrastructure and SNMP data. The Infrastructure Data Collector uses the Monitoring API to pass operational data and monitoring model data to the Analytic Engine. The Monitoring API is designed as a programmatic interface to the systems monitoring functionality of a business integration platform. It communicates over JMS, utilizing a JMS queue, to provide secure and reliable communications between Optimize and the Infrastructure Data Collector.

The Infrastructure Data Collector lets you identify webMethods component resources that you want to monitor and analyze. Using the Infrastructure Data Collector, you can monitor the following:

Component resource	Description
Broker Servers	When you monitor a Broker Server, Infrastructure Data Collector automatically monitors related items. Such items include Brokers, Custom Adapter (Broker Clients), and Document Types that are running on the Broker Server.
Integration Servers	When you monitor an Integration Server, Infrastructure Data Collector automatically monitors related items as well. Such items include the Integration Server repository and all Integration Server ports; Trading Networks Server, if you have that component installed; packages that are running on the Integration Server; and the connections, polling notifications, and services of adapters that are running on the Integration Server. To lessen the load on Infrastructure Data Collector, you can choose to monitor only certain services on an Integration Server.
webMethods adapters	Infrastructure Data Collector enables you monitor adapters installed on Integration Servers. When you start monitoring an Integration Server, Infrastructure Data Collector automatically starts monitoring all adapters that are installed on the Integration Server.
SNMP agents	Infrastructure Data Collector enables you specify how often the SNMP data collector discovery and monitoring agents poll SNMP agents for variables and table instances. You can also specify how often the agents attempt to connect to SNMP agents when one or more agents are down.

Web Service Data Collector

The Web Service Data Collector is installed automatically with Optimize. Web services provide an interface to Optimize through which client programs can send operational data about applications or equipment in your enterprise (such as databases, printers, or disk drives), process data (such as the start and end time of each step in process instances), or business data (such as customers, order quantities, and revenues). Optimize then displays the data in the form of system KPI instances for operational data or business KPI instances for business process data.

Tip: You can use Software AG Designer to generate the client program and the SOAP messages for the Web Service Data Collector automatically. For details, see the “Working with Web Services” section of the *webMethods Service Development Help*.

Depending on the size of your enterprise, you can distribute the load by deploying Web services to systems that host applications and processes you want to monitor. Web services gather the summarized data and send them to the Analytic Engine, which in turn stores the data in the database and displays it in the My webMethods user interface. In addition, the Analytic Engine runs algorithms on the data to produce a variety of diagnoses. These diagnoses enable Optimize to automatically and continually compare summarized data with a baseline of historical behavior and indicate when types of data are deviating from that baseline.

Note: You can provide your own definitions of what types of data are out of compliance by defining rules. For more information about defining rules, see Chapter 5 “Defining Rules” in *webMethods Optimize User’s Guide*.

If you have webMethods Integration Server and Designer, you can build Integration Server services that use Web services to gather data for Optimize. For example, you could do any of the following:

- You can use Integration Server services to gather data about webMethods components. Suppose you are using webMethods Broker. You could create an Integration Server service that uses Web services to create system KPI instances for types of data in certain Broker documents, listens for those Broker documents, gets the appropriate field values from the documents, and uses Web services to push the data to the created system KPI instances in Optimize. For more information, see [“Configuring and Using the Web Service Data Collector” on page 253](#).
- You can use webMethods adapters with Integration Server services to gather data about resources in your enterprise. Suppose your enterprise includes an SAP system and you use the webMethods SAP Adapter to connect that resource to your webMethods components. You could create an Integration Server service that uses Web services to create system KPI instances for types of data relating to the SAP system, uses the SAP Adapter to query the SAP system for the data at regular intervals, and uses Web services to push the data to the created system KPI instances in Optimize.
- In Optimize for Process, you can use Web services to gather data about business processes in your enterprise. Suppose you want to monitor a business process. You would use Web services to push data about each process instance to Optimize as the process instance executes. Typically, you would push data about the start and end of each step in the process instance and about process run-time events that might indicate that an error or a warning occurred during process execution. You could obtain that data from a log file, from a database, or from the business process code itself.
- Also in Optimize for Process, you can use Web services to gather data about your enterprise’s business transactions. If you are monitoring business data, you would

use Web services to push data about each transaction instance to Optimize as the instance executes. Typically, you would push data about customers, products, order quantities, and revenue.

Key Performance Indicators (KPIs)

You can monitor quantitative business and system data by defining *key performance indicators (KPIs)*. KPIs are quantifiable measurements that reflect the critical success factors of an organization. Optimize uses KPIs to monitor data for exceptions and trends and to help you answer questions such as the following:

- What is the queue length for this operating system?
- How many orders over \$10,000 have completed for this process? (Optimize for Process)
- What is the revenue for the Southeast Region? (Optimize for Process)

For example, when using Optimize for Process, in an order management process, you might define KPIs for how many orders were received, their dollar amounts, and whether they were processed successfully. You can also define KPIs that compare volume by order channel or cycle time by customer type.

In Optimize, KPIs consist of the following things:

- **Dimensions**, such as customer, region, dealer, or sales person
- **Event maps**, which associate business data, such as dimensions, *measures* (measurable and monitorable numeric quantities such as revenue), and transactional attributes, with a particular business process
- **KPI definitions**, which associate a measure with a *dimension* (a means for breaking out your data into slices), such as revenue by region or queue length by host
- **Hierarchies**, which determine how dimensions are related. Optimize uses these relationships to aggregate and organize data on My webMethods business process and system monitoring pages.

KPI instances and KPIs

You can create a *key performance indicator (KPI)* for each type of data that the Web Service Data Collector collects. Optimize creates and stores data that corresponds to the KPI in a *KPI instance*. For example, for a KPI that tracks Broker client connection state, Optimize creates one KPI instance for each tracked Broker. For more information about the Web Service Data Collector, see "[Web Service Data Collector](#)" on page 32.

When you create a KPI, you can specify a collection interval to summarize the data stored for the KPI instance associated with the KPI. The Analytic Engine runs algorithms on the data to produce a variety of diagnoses. These diagnoses enable Optimize to automatically and continually compare summarized data samples to a baseline of historical behavior and indicate when types of data are deviating from that baseline.

Note: You can provide your own definitions of when types of data are out of compliance by defining rules. For more information about defining rules, see Chapter 5 “Defining Rules” in *webMethods Optimize User’s Guide*.

System KPI instances store data about applications, and *business KPI instances* store data about business data and processes. In the My webMethods interface, each type of data that Optimize collects is represented by the KPI instance in which the data is stored.

Each KPI instance performs a specific function for its data, as described in the following table.

KPI instance	Function
Sum	Sums the data points collected over each collection interval. An example use for this type of KPI instance is to tally process step errors.
Average	Calculates the average value of the data collected over each collection interval. An example use for this type of KPI instance is to track CPU usage.
Last Value	Stores the value of the last data collected in the collection interval. An example use for this type of KPI instance is to track the size of a queue.
State	Stores the value of last data collected in each collection interval, where the possible values are 1 (online) or 0 (offline). An example use for this type of KPI instance is to track the state of an application.
Count	Counts the number of data points over each collection interval. An example use for this type of KPI instance is to count the number of older events.

Business Processes

Optimize for Process can analyze and display status information for the following two types of business processes:

- *webMethods-executed processes* (processes that are executed and tracked by the webMethods Process Engine).
- *Externally-executed processes* (processes that are not executed and tracked by the webMethods Process Engine). For externally-executed processes, Optimize has no knowledge of the rules and data that determine whether these transitions are valid. Therefore, for these processes, Optimize does not recognize invalid step transitions

based on branch or split conditions and does not detect or report errors when such conditions occur.

You can use Software AG Designer to design business process models. For more information about using Software AG Designer to set up business processes and process steps, see the Software AG Designer online help. For more information about setting up and identifying those business processes you want Optimize to monitor, see ["Configuring Business Processes" on page 99](#).

Rules

Optimize evaluates data against *rules* you define. A rule identifies a significant condition under which a process or resource becomes problematic, requires attention, or is of interest. A rule can be configured to alert users or take programmatic action when the condition occurs. Rules can also be configured to send user alerts when the process or resource returns to compliance. You can define KPI rules, event rules and threshold rules.

When you define a rule, you identify the type of data to which to apply the rule (for example, Broker queue length). By default, Optimize then applies the rule to that type of data on all processes or applications (for example, all Brokers in your entire environment). However, you might want to apply the rule to that type of data for specific processes or applications only (for example, for certain Brokers only). In this case, you create a *rule filter* that identifies only the specific processes or applications. When you define the rule, you associate the rule filter with the rule.

During rule definition, you also identify the users to whom to send alerts when the rule is violated, or you can specify other actions to perform when a rule violation occurs (for example, send an SNMP alert to an SNMP manager or invoke a Web service).

KPI Rules

KPI rules apply to KPIs that were created for business or system data. Optimize evaluates data against a KPI rule over one or more intervals according to the collection interval specified when the KPI was first created.

Event and Threshold Rules

Event rules and *threshold rules* differ from KPI rules in that Optimize evaluates data points against the rule as they are collected (that is, in real time) rather than waiting for a KPI collection interval (that is, an aggregation of data points collected over an interval). Event and threshold rules generally apply to types of data that require immediate response, such as an application unexpectedly shutting down or CPU usage exceeding unacceptable limits. Optimize maintains state for event rules so that you can track data over multiple intervals, but it does not maintain state for threshold rules. On the other hand, for capturing any individual order with a size greater than a million dollars, a threshold rule would be preferable to an event rule.

For more information on KPI, event, and threshold rules, see Chapter 5 "Defining Rules" in *webMethods Optimize User's Guide*.

Users

Optimize has two types of users: administrators and business analysts. Note that “business analyst” is a type of user that exists only in the context of Optimize for Process.

Administrators

Administrators perform the following tasks:

- Install Optimize.
- Set up user access.
- Identify the mail server to use to send alerts about problems to users.
- Change interface and communication settings.
- Specify settings for statistical intervals and statistical confidence.
- Define business and system data and identify processes and system objects to monitor.
- Configure data collectors and create rule filters.
- Define rules to use to evaluate collected data.
- Set up logging.
- Maintain KPI instances and the database.

Business Analysts

“Business analyst” is a type of user that exists only in the context of Optimize for Process. Business analysts perform the following tasks:

- Define rules to use to evaluate collected data.
- Scan for business and system problems.
- View analytic mashups, and rule details to help locate the source of a problem.
- Respond to alerts to resolve a problem.

Users, Groups, and Roles

Optimize Administrators can use the My webMethods Roles, Groups, and Users pages (**Navigate > Applications > Administration > User Administration**) to create users, groups, and roles. Administrators can also grant access to My webMethods functions using Lightweight Directory Access Protocol (LDAP) or data-level security. For more information on users, groups, and roles, see *Administering My webMethods Server*. For more information on data-level security, see the following sections in this guide:

["Enabling Data-Level Security"](#) on page 75 and ["Granting User Access to KPIs"](#) on page 177.

2 Starting, Stopping, and Accessing Optimize

■ Starting Optimize Manually	40
■ Stopping Optimize	44
■ Starting and Stopping Infrastructure Data Collector	46
■ Managing Infrastructure Data Collector as a Service	47
■ Starting Infrastructure Data Collector as a Daemon in Linux	48
■ Viewing Information about Optimize Components	48
■ Viewing Information about Optimize-Related Products	50
■ Viewing Information About Jobs	50

Starting Optimize Manually

During Optimize installation, you configured each Analytic Engine, and each Web Service Data Collector to start and stop automatically at host system startup and shutdown. However, when you start Optimize for the first time after installation, you must do so manually as described below.

Important: Running Optimize on an operating system that uses an IPv6 network interface requires modification of the wrapper.conf file for the Analytic Engine. The file is located in the [installation directory]\optimize\analysis\conf directory. Add the following JVM directive under the "Java Additional Parameters" section:

```
wrapper.java.additional.20=-Djava.net.preferIPv4Stack=true
```

Starting Optimize on a UNIX System

To start Optimize, you must start each Analytic Engine, the My webMethods Server, and each Web Service Data Collector.

Starting an Analytic Engine

To start each Analytic Engine on a UNIX system

1. Log on to the Analytic Engine host system as the webmbam user.
2. Navigate to the *Optimize_directory*/analysis/bin directory and enter the command:

```
./startup.sh
```

Starting My webMethods Server

To start My webMethods Server on a UNIX system

1. Log on to the My webMethods Server host system as the webmbam user.
2. Navigate to the MWS/server/default/bin directory and enter the command:

```
./startup.sh
```

Starting a Web Services Data Collector

To start each Web Service Data Collector on a UNIX system

1. Log on to the Web Service Data Collector host system as the webmbam user.
2. If you have not already done so, run the WSdatacollector_rc_deploy.shrun command script (for instructions, see *Installing webMethods and Intelligent Business Operations Products*).
3. Navigate to the *Optimize_directory* /dataCollector/bin directory and enter the command:

```
./startup.sh
```

4. After you have started the Analytic Engine, the My webMethods Server, and the Web Service Data Collectors, log in to Optimize through the My webMethods user interface. For more information about My webMethods, see *Working with My webMethods* .

Starting on Solaris 64-bit and HP-UX PA-RISC Platforms

Use the instructions above for starting on a UNIX system, but first edit the scripts as follows:

Editing the Analytic Engine Script

To edit the script for the Analytic Engine

1. Open the wrapper.conf file for the Analytic Engine using a text editor. The file is located in the *[installation directory]\optimize\analysis\conf* directory.
2. Add the following JVM directive under the "Java Additional Parameters" section:
`wrapper.java.additional.21=-DGLUE_OPTS="$GLUE_OPTS -d64"`
3. Save the file.

Editing the Web Service Data Collector Script

To edit the script for the Web Service Data Collector

1. Open the wrapper.conf file for the Web Service Data Collector in a text editor. The file is located in the *[installation directory]\optimize\dataCollector\conf* directory.
2. Add the following JVM directive under the "Java Additional Parameters" section:
`wrapper.java.additional.21=-DGLUE_OPTS="$GLUE_OPTS -d64"`
3. Save the file.

Changing the Webmbam User when running Optimize on UNIX

The UNIX rc scripts use the "webmbam" user to ensure that the Optimize engines (Analytic Engine and Web Service Data Collector) run using a UNIX user other than root. Note that the "webmbam" user is not created by the webMethods installer when Optimize is installed. The user must be created by the UNIX administrator. The rc scripts, rscript.sh, determine start up order for Optimize components, and they are located in the bin directory under each Optimize engine - *[installation directory]/optimize/analysis/bin* and *[installation directory]/optimize/dataCollector/bin*.

To configure a user other than webmbam

1. Create the desired UNIX user.
2. Ensure that user has execution and write privileges to the following directories - *[installation directory]/optimize/analysis* and *[installation directory]/optimize/dataCollector*

3. Open the `kenobiEnv.sh` file located at `[installation directory]/optimize/analysis/bin` with an appropriate text editor.
4. Update the "OPTIMIZE_USER" setting to the UNIX user created earlier overwriting the "webmbam" user.
5. Open the `kenobiEnv.sh` file located at `[installation directory]/optimize/dataCollector/bin` using an appropriate text editor.
6. Update the "OPTIMIZE_USER" setting to the UNIX user created earlier overwriting the "webmbam" user.

Starting Optimize on a Windows System

To start Optimize, you must start each Analytic Engine, My webMethods Server, and each Web Service Data Collector. Depending on how they were installed, you can choose one of three ways to start these components:

- As a Windows service.
- From a batch file.
- From the Windows **Start** menu.

Note: When starting Optimize components under Windows Vista and newer Windows operating systems, the user must have full administrative privileges for the installation directory on the applicable machine (e.g., C:\SoftwareAG). Note that full explicit privileges are required; it is not sufficient for a user to be simply a member of the administrative group.

Before you can start the Optimize components as Windows services, you need to install them as services.

Installing an Analytic Engine as a Windows Service

To install an Analytic Engine as a Windows service

On the Analytic Engine host, navigate to the `Optimize_directory\optimize\analysis\bin` directory and execute one of the following commands from a command prompt depending on whether you want to have the service start automatically:

- To install the service without starting it, type the following:

```
service.bat -install
```
- To install the service and start it immediately, type the following:

```
service.bat -installstart
```

Uninstalling an Analytic Engine as a Windows Service

To uninstall an Analytic Engine that was installed as a Windows Service, navigate to the `Optimize_directory\optimize\analysis\bin` directory and execute the following command:

```
service.bat -remove
```

Installing a Web Service Data Collector as a Windows Service

To install a Web Service Data Collector as a Windows service

On the Web Service Data Collector host system, navigate to the *Optimize_directory*\optimize\dataCollector\bin directory and execute one of the following from a command prompt:

- To install the service without starting it, type the following:
`service.bat -install`
- To install the service and start it immediately, type the following:
`service.bat -installstart`

Uninstalling a Web Service Data Collector as a Windows Service

To uninstall a Web Service Data Collector that was installed as a Windows service, navigate to the *Optimize_directory*\optimize\dataCollector\bin directory and execute the following command:

```
service.bat -remove
```

Starting webMethods Components on a Windows System

To start each Analytic Engine, My webMethods Server, and each Web Service Data Collector on a Windows system

1. Start each Analytic Engine by doing one of the following on the Analytic Engine host system:
 - Open the Windows **Services** window and start the **Software AGwebMethodsAnalytic Engine 8.2** service.
 - From a command prompt, navigate to the *Optimize_directory* \analysis\bin directory and enter the command:
`startup.bat`
 - From the Windows **Start** menu, choose **Programs > Software AGStart Servers > Start Optimize Analytic Engine 8.2**.
2. Start My webMethods Server by doing one of the following on the My webMethods Server host system:
 - Open the Windows **Services** window and start the **Software AGwebMethods My webMethods Server 8.2(default)** service.
 - From a command prompt, navigate to the MWS\server\default\bin directory and enter the command:
`startup.bat`
3. Start each Web Service Data Collector by doing one of the following on the Web Service Data Collector host system:

- Open the Windows **Services** window and start the **Software AG webMethods Optimize Web Service Data Collector 8.2** service.
 - From a command prompt, navigate to the *Optimize_directory\dataCollector\bin* directory and enter the command:

```
startup.bat
```
 - From the Windows **Start** menu, choose **Programs > Software AG > Start Servers > Start Optimize Web Service Data Collector 8.2**.
4. After you have started the Analytic Engines, My webMethods Server, and the Web Service Data Collectors, log in to Optimize through the My webMethods user interface. For more information about My webMethods, see the *Working with My webMethods* .

Stopping Optimize

This section explains how to stop the Analytic Engines, My webMethods Server, and the Web Service Data Collectors. To bring Optimize to a complete stop, perform all three procedures.

Stopping Analytic Engines

To stop the Analytic Engines

System	Action
UNIX	<p>Log on to each Analytic Engine host system as the webmbam user, navigate to the <i>Optimize_directory/analysis/bin</i> directory, and enter the command:</p> <pre>./shutdown.sh</pre>
Windows	<p>Log on to each Analytic Engine host system and do one of the following:</p> <ul style="list-style-type: none"> ■ Open the Windows Services window and stop the Software AG webMethods Analytic Engine 8.2 service. ■ From a command prompt, navigate to the <i>Optimize_directory\analysis\bin</i> directory and enter: <pre>shutdown.bat</pre> ■ From the Windows Start menu, choose Programs > Software AG > Stop Servers > Stop Optimize Analytic Engine 8.2.

Stopping My webMethods Server

To stop My webMethods Server

System	Action
UNIX	<p>Log on to the My webMethods Server host system as the webmbam user, navigate to the MWS/server/default/bin directory, and enter the command:</p> <pre>./shutdown.sh</pre>
Windows	<p>Log on to the My webMethods Server host system and do one of the following:</p> <ul style="list-style-type: none"> ■ Open the Windows Services window and stop the Software AG webMethods My webMethods Server 8.2 (default) service. ■ From a command prompt, navigate to the MWS\server\default\bin directory and enter: <pre>shutdown.bat</pre> ■ From the Windows Start menu, choose Programs > Software AG > Stop Servers > Stop My webMethods Server 8.2.

Stopping Web Service Data Collectors

To stop Web Service Data Collectors

System	Action
UNIX	<p>Log on to the Web Service Data Collector host system as the webmbam user, navigate to the <i>Optimize_directory</i>/dataCollector/bin directory, and enter the command:</p> <pre>./shutdown.sh</pre>
Windows	<p>Log on to the Web Service Data Collector host system and then do one of the following:</p> <ul style="list-style-type: none"> ■ Open the Windows Services window and stop the Software AG webMethods Optimize Web Service Data Collector 8.2 service. ■ From a command prompt, navigate to the <i>Optimize_directory</i>\dataCollector\bin directory and enter: <pre>shutdown.bat</pre> ■ From the Windows Start menu, choose Programs > Software AG > Stop Servers > Stop Optimize Web Service Data Collector 8.2.

Starting and Stopping Infrastructure Data Collector

Starting Infrastructure Data Collector

The method you use to start Infrastructure Data Collector depends on the operating system on which it is running. Refer to the following table for operating system-specific instructions.

Note: When starting Infrastructure Data Collector under Windows Vista and newer Windows operating systems, the user must have full administrative privileges for the installation directory on the applicable machine (e.g., C:\SoftwareAG). Note that full explicit privileges are required; it is not sufficient for a user to be simply a member of the administrative group.

To start Infrastructure Data Collector

System	Action
Windows	<p>Start > Programs > Software AG > Start Servers > Start Optimize Infrastructure Data Collector 8.2</p> <p>Alternatively, you can execute the following:</p> <pre><install dir>\InfrastructureDC\bin\startup.bat</pre>
UNIX	<p>Open a command window, navigate to the <i>Software AG_directory/InfrastructureDC/bin</i> directory, and run this command: <code>./startup.sh</code>.</p>

Stopping Infrastructure Data Collector

The following table describes how to stop Infrastructure Data Collector on Windows and Unix systems.

To stop Infrastructure Data Collector

System	Action
Windows	<p>Start > Programs > Software AG > Stop Servers > Stop Optimize Infrastructure Data Collector 8.2</p> <p>Alternatively, you can execute the following:</p> <pre><install dir>\InfrastructureDC\bin\Shutdown.bat</pre>

System	Action
UNIX	Open a command window, navigate to the <i>Software AG_directory/InfrastructureDC/bin</i> directory, and run this command: <code>./Shutdown.sh</code> .

Managing Infrastructure Data Collector as a Service

You can use the `infradc_service` script to install, run, and manage Infrastructure Data Collector as a Windows service. The script contains options to start and stop Infrastructure Data Collector, as well as options to check the status of and print information about the component.

To invoke the `infradc_service` script

1. Open a command prompt.
2. Type `infradc_service <command>`. The following table describes the commands/parameters available with the `infradc_service` script.

Command	Result
<code>-start</code>	Start as a Windows service.
<code>-stop</code>	Stop as a running Windows service.
<code>-install</code>	Install as a Windows service.
<code>-installstart</code>	Install as a Windows service and start.
<code>-remove</code>	Remove as a Windows service.
<code>-dump</code>	Request a thread dump.
<code>-query</code>	Query the current status of the service.
<code>-querysilent</code>	Silent query the current status of the service.
<code>-version</code>	Print the wrapper's version information.
<code>-help</code>	Print this help message.

Starting Infrastructure Data Collector as a Daemon in Linux

You can configure Infrastructure Data Collector to start as a daemon on a Linux system by making a slight modification to the `custom_setenv.sh` script as described in the following procedure. Note that you do not need to modify the `startup.sh` file. The changes you make to the `custom_setenv.sh` files will override the default startup script settings automatically.

To start Infrastructure Data Collector as a daemon on a Linux system

1. If the applicable Infrastructure Data Collector is running, stop it before proceeding.
2. Navigate to the *Software AG_directory* `/profiles/InfraDC/bin` folder on the machine running the applicable Infrastructure Data Collector and open the `custom_setenv.sh` file with an appropriate text editor.
3. Locate the `STARTUP_MODE=console` line and change it to `STARTUP_MODE=daemon`.
4. Save the file and close it.
5. Start Infrastructure Data Collector as you normally would.

Viewing Information about Optimize Components

Optimize is installed in environments appropriate to your system configuration. Environments are groupings of webMethods product components that share common configuration settings. You can view information about Optimize components installed in any of the environments defined in your system from the System Information page.

To view information about Optimize Components

1. In My webMethods: **Navigate > Applications > Administration > System-Wide > System Information.**

The System Information page is displayed.

System Information

Select an Environment: BVTEnv

On-Line	Type	Component ID	Host	Port	Version	Fix	Additional Information
✓	Analytic Engine	Analytic Engine v8.0.0.0	BAMBVT1	12503	8.0.1.0.111		CPU.type: Intel Family 15 Model 2 Stepping 9 Hostname: bambvt1.AME.ad.sag OS.Service.Pack: Service Pack 2 HW.memory[GB]: 3.50 OS.Name: Windows.2003 OS.Release: 5.2 OS.Build.no. 3790 CPU.cores: 4 3876 total process instances 4624 total KPI instances (not including base Analytic Engine KPIs)
✓	Data Collection Services	Web Service Data Collector v8.0.0.0	BAMBVT1	12603	8.0.1.0.111		CPU.type: Intel Family 15 Model 2 Stepping 9 Hostname: bambvt1.AME.ad.sag OS.Service.Pack: Service Pack 2 HW.memory[GB]: 3.50 OS.Name: Windows.2003 OS.Release: 5.2 OS.Build.no. 3790 CPU.cores: 4
✓	Prediction Engine	Prediction Engine v8.0.0.0	BAMBVT1	12513	8.0.1.0.111		CPU.type: Intel Family 15 Model 2 Stepping 9 Hostname: bambvt1.AME.ad.sag OS.Service.Pack: Service Pack 2 HW.memory[GB]: 3.50 OS.Name: Windows.2003 OS.Release: 5.2 OS.Build.no. 3790 CPU.cores: 4

1 - 3 of 3 Total

- From the **Select an Environment** list, select the environment for which you want to view component information.

You can view the following information about the Analytic Engine and Web Service Data Collector:

Field	Description
Online	Whether the component is online. ✓ = Online ○ = Off-line
Type	Optimize component type: ■ Analytic Engine ■ Web Service Data Collector
Component ID	Unique ID for the component. Identical to the Logical Server ID provided by the Configuration Engine.
Host	Host name or IP address of the system on which the component is running.
Port	Configuration port for the component on the host.
Version	Version number of the software installed on the component.

Field	Description
Fix	Fix number of the software installed on the component, if any.
Additional Information	Details about the computer hardware and operating system on which the component is running.

Viewing Information about Optimize-Related Products

You can view the following information about Central Configuration, Optimize for Infrastructure, Monitor, My webMethods Server, Optimize for Process, Messaging, and Task Engine from the About page accessed from the tool bar located in the top right corner of the My webMethods banner:

Field	Description
Version	Product version.
Build Number	Product build number.
Build Date	Product build date.
Service Pack	Product build service pack number, if any.
Fix	Product fix number, if any.
Release	Product release name.

Viewing Information About Jobs

The Job Manager manages tasks and activities within Optimize. It includes a simple user interface that enables Optimize administrators to view activity of the Job Manager. Viewable activities include failed jobs and any completed jobs for which audit logging was specified. There are two Job Manager related pages: the Job Audit page and the Job Queue page. These pages are explained below:

- Job Audit page - This page displays basic information for current activities. Displayed information includes name, category, run time and status.
- Job Queue page - This page displays jobs in the order in which they are scheduled for completion. Displayed information includes the job name, category, scheduled time, persistence (Transient or persistent) and status. Also, each job includes a icon that you can click to view additional details

To view either of these pages navigate the following path in My webMethods and select the desired page: **Navigate > Applications > Administration > Analytics > Job Management.**

3 Configuring Optimize

- Configuring and Deploying Your Environment 54
- Configuring Rule Violation Alerts and Actions 57
- Changing Subcomponent Communication Settings 68
- Securing Optimize Engines with SSL 71
- Enabling Data-Level Security 75
- Changing the Optimize Authentication Credentials 76
- Specifying Statistical Intervals 78
- Optimizing Analytic Engine Performance 79
- The Java Service Wrapper 84
- Configuring Analytic Engine High Availability 89
- Optimizing Web Service Data Collector Performance 90
- Connecting Process Engines to Analytic Engines 91
- Adjusting RAM for the Analytic Engine and Web Service Data Collectors 91
- Configuring Optimize to use an Analytic Engine 92
- Configuring Optimize for Analytic Engine Clustering 93
- Configuring Optimize to Monitor Multiple Integration Servers and Analytic Engines 93
- Configuring EDA/NERV Settings for Optimize 94
- Guaranteed Event Storage in an Optimize Environment 94
- Implementing Guaranteed Event Storage in Optimize 95
- Configuring the PPM Server Settings 98

Configuring and Deploying Your Environment

Optimize uses the My webMethods Define Environments page, (sometimes referred to as the webMethods Central Configuration tool) as the graphical user interface for configuring an Optimize system without manually editing properties files. The Define Environments page enables you to configure Optimize components, such as an Analytic Engine and Data Collector. In addition, to connect My webMethods to Optimize (an Analytic Engine) or to an Integration Server, you must use the My webMethods System Settings page.

A working group of webMethods product components that share common configuration settings is called an *environment*. A default Optimize environment contains the following logical servers: a Broker, an Analytic Engine, a Web Service Data Collector, and a My webMethods Server. For more complete information on configuring your environment, see *Configuring BAM*.

One to three default environments are available in Optimize, depending on the chosen installation option. These are BAM only, BPM only, and BAM/BPM mixed (also known as BPMS). If you choose a BPM only or a BAM/BPM mixed installation, the BPM environment is immediately accessible from the My webMethods monitoring pages using a default Integration Server. If you choose a BAM-only environment, or want to access the BAM components of a mixed installation, you must configure, deploy, and add one or more Analytic Engines to the System Settings page before you can use them to generate and view monitoring information. Refer to "[Configuring Optimize to use an Analytic Engine](#)" on page 92 in this chapter for instructions to add an Analytic Engine.

A **Server** drop-down menu appears at the top of most My webMethods pages that enables users to select from the servers/environments available to them.

Access to the Define Environments page is through the My webMethods interface. Refer to *Configuring BAM* for an overview of the process of installing and configuring Optimize and My webMethods. Once you have set up your environment using the Define Environments page, you can modify your configuration at any time through My webMethods.

To configure and deploy your environment

1. In My webMethods, navigate the following path: **Applications > Administration > System-Wide > Environments > Define Environments**.

The Define Environments page is displayed.

2. Add an environment:

- a. Click **Add Environments**.

The Add Environment page is displayed.

- b. On the **Environment Information** panel, enter a name for the environment.

- You might want to choose a name that reflects the environment functionality, such as “development” or “production.”
- c. In the **Description** field, enter a description for the environment.
 - d. Click **Save**.
3. Add logical servers:
- a. Open your environment by clicking its name in the Define Environments page. The Edit Environment page is displayed.
 - b. Click the **Design Servers** tab.
 - c. Click **Add From Template** to create a default set of logical servers.

The default set of Optimize logical servers contains one each of the following:

- Analytic Engine
- JMS Server (Either Broker Server or webMethods Universal Messaging)
- Data Collection Services (Web Service Data Collector for Optimize for Process, and Infrastructure Data Collector for Optimize for Infrastructure)
- My webMethods Server

Tip: To make your logical servers easier to recognize, you can rename them with information about their location or use. To rename a logical server, click its name in the Design Servers tab and modify the name and description in the Edit Logical Server dialog.

- d. Click **Add** to add another server not included in the default logical server set.
4. Configure servers:
- a. Click the **Configure Servers** tab.
 - b. Do one of the following:
 - To modify a server configuration, click the name of the attribute you want to change. Click **Save** when you have made your changes.

Tip: To keep your changes, you must click **Save** before closing each configuration editor. If you close an editor without clicking **Save**, you will lose your changes.

- Do nothing to accept the default attribute settings.

Important: If more than one Optimize environment exists on your network, you must define a unique Cluster Port number by clicking **Cache Configuration** in the Default Settings section of the configuration tree. A conflict will result if the cluster port is not unique. Choose a port number between 10000 and 64768 to avoid conflicts.

5. Define Hosts:
 - a. Click the **Define Hosts** tab.
 - b. For each host in your environment, click **Add Host**, supply the display name and host name or IP address of the host in the Add/Edit Host dialog, and click **OK**.
6. Map servers:
 - a. Click the **Map Servers** tab.
 - b. For each logical server, click the name of the server (or click  **Edit**) to display the Edit Host Mapping swap box. For each logical server, specify one or more host server locations by double-clicking the host name(s) to move from the **Available Hosts** list to the **Mapped Hosts** list, and click **Save**.
7. Map Endpoints:
 - a. Click the **Map Endpoints** tab.
 - b. Check the incoming and outgoing connections for each logical server and modify port numbers, if necessary.
 - c. Click **Save** to save changes.
8. Map DB Pools:

Note: In order to map DB pools, you need to have previously configured DB pools. See "[Configuring Optimize Database Pools](#)" on page 321 for detailed DB pool configuration instructions.

 - a. Click the **Map DB Pools** tab.
 - b. For each database component, select a pool from the drop-down list.
 - c. To make changes to a pool already associated with this component, click  **Edit** to display the Add/Edit Pool panel, make any necessary changes to the pool information, database connection, or pool settings, and click **Save** on the Add/Edit Pool page to save these changes.
 - d. Click **Save** on the Database Components panel to save any changes in the mapping of pools to the database components.
9. Validate your environment:
 - a. Click the **Validate** tab.
 - b. Click **Finish**.
10. Deploy your environment:
 - a. Click  **Deploy**.
 - b. Click **Deploy All**.

Changing Your Configuration Settings

When you change settings (other than log settings) using the My webMethods Define Environments page, you must complete the following steps before the change can take effect:

1. Do one of the following:
 - Click **Deploy Updates** to send only the changed settings to the appropriate logical server(s).
 - Click **Deploy All** to send all settings to all logical servers in your environment.
 - Click **Deploy to File** to save all settings to all logical servers in your environment as a file.
2. Restart the logical server(s) for which you changed settings.

Configuring Rule Violation Alerts and Actions

When a rule is violated, you can configure Optimize to do one or more of the following:

- Send e-mail alerts to specified users. If you subscribe to a pager or mobile phone service that supports text messaging to an e-mail address, you can also receive e-mail alerts on these devices. You can customize the body and subject line of e-mail alerts by creating custom alert templates.
- Send SNMP alerts to an SNMP manager.
- Invoke a Web service action.

Configuring E-mail Alerts

You can configure Optimize to notify interested parties by e-mail when rules are violated. Any defined user can be specified as the alert recipient when an alert is added to a rule in the Add/Edit/Copy Rule page (for more information, see *webMethods Optimize User's Guide*).

The My webMethods Define Environments page Mail Settings panel enables you to specify a custom alert e-mail template, in addition to configuring such basic e-mail options as mail server settings and sender information. Optimize uses a default template to format e-mail alerts unless you specify a custom template. This section describes how to configure your mail settings and how to create and specify a custom alert template.

For your custom template, you can create either a *dynamic template* that uses dynamic variable resolution to support more advanced formatting (preferred), or a *static template* that uses a fixed set of substitution variables (supported for compatibility with previous installations of Optimize).

Specifying E-mail Alert Settings and Templates

You can configure the settings for sending e-mail alerts from the My webMethods Define Environments page.

To configure settings for sending e-mail alerts

1. In My webMethods, navigate the following path: **Applications > Administration > System-Wide > Environments > Define Environments**.

The Define Environments page is displayed

2. Click the name of the configured environment for which you want to configure e-mail alert settings.

The Edit Environment page is displayed.

3. Click the **Configure Servers** tab.

4. On the **CONFIGURATION** tree, find the Analytic Engine for which you want to configure mail alert settings, and click **Mail Settings** under the name of the Analytic Engine. The Analytic Engine Mail Settings panel is displayed.

Mail Server:	<input type="text" value="mail.webmethods.com"/>
Sender Domain:	<input type="text" value="webmethods.com"/>
Default Sender:	<input type="text" value="optimize@webmethods.com"/>
Admin Address:	<input type="text" value="admin@webmethods.com"/>
Default Mail Encoding:	<input type="text" value="UTF-8"/>
Templates:	<input type="text" value="Business monitor violation=
./templates/BusinessMonitorViolation.template"/>

5. Set the following attributes:

Attribute	Definition
Mail Server	Name of the host system of the SMTP server for sending e-mails containing alerts (for example, smtp.client.com).

Attribute	Definition
Sender Domain	E-mail domain for the <code>defaultSender</code> attribute (for example, <code>webmethods.com</code>).
Default Sender	Sender to specify in the From field of the alert e-mails. Make sure the e-mail address is within the e-mail domain specified on the <code>senderDomain</code> attribute.
Admin Address	Recipient to receive copies of e-mail alerts if the <code>adminEmailOn</code> attribute is set to <code>true</code> . This attribute also specifies the default address for sending test e-mail messages from the System Settings page (Applications > Administration > My webMethods > System Settings).
Default MailEncoding	Default encoding to use for e-mail, instant messenger, pager, and mobile phone text that is sent to users. You can specify any MIME-registered encoding name. The default is UTF-8. You can specify a different encoding for individual users on the Edit User page. For information on configuring user profiles, see <i>Working with My webMethods</i> .
Templates	<p>Used to specify custom e-mail alert templates.</p> <p>Syntax: <code><rule name > = <template name ></code></p> <ul style="list-style-type: none"> ■ Dynamic custom alert template filenames <i>must</i> end with the <code>.vm</code> extension. ■ Static custom alert template filenames typically end with the <code>.template</code> extension. <p>When the specified rule is violated, the e-mail alert is formatted based on the associated alert template. First it searches for a template configured with the rule's instance name. If one is not found, it searches for a template configured with the base rule. If no specified templates are found, the Analytic Engine uses the default format for the e-mail alert.</p> <p>For example, in the Mail Settings panel above, the template attribute specifies that when the rule named <code>Business monitor violation</code> is triggered, the Analytic Engine uses the static custom alert template named <code>BusinessMonitorViolation.template</code> that is in the default templates directory (<code>Optimize_directory/analysis/conf/templates</code>):</p>

Attribute	Definition
	<code>Business monitor violation= ./templates/BusinessMonitorViolation.template</code>

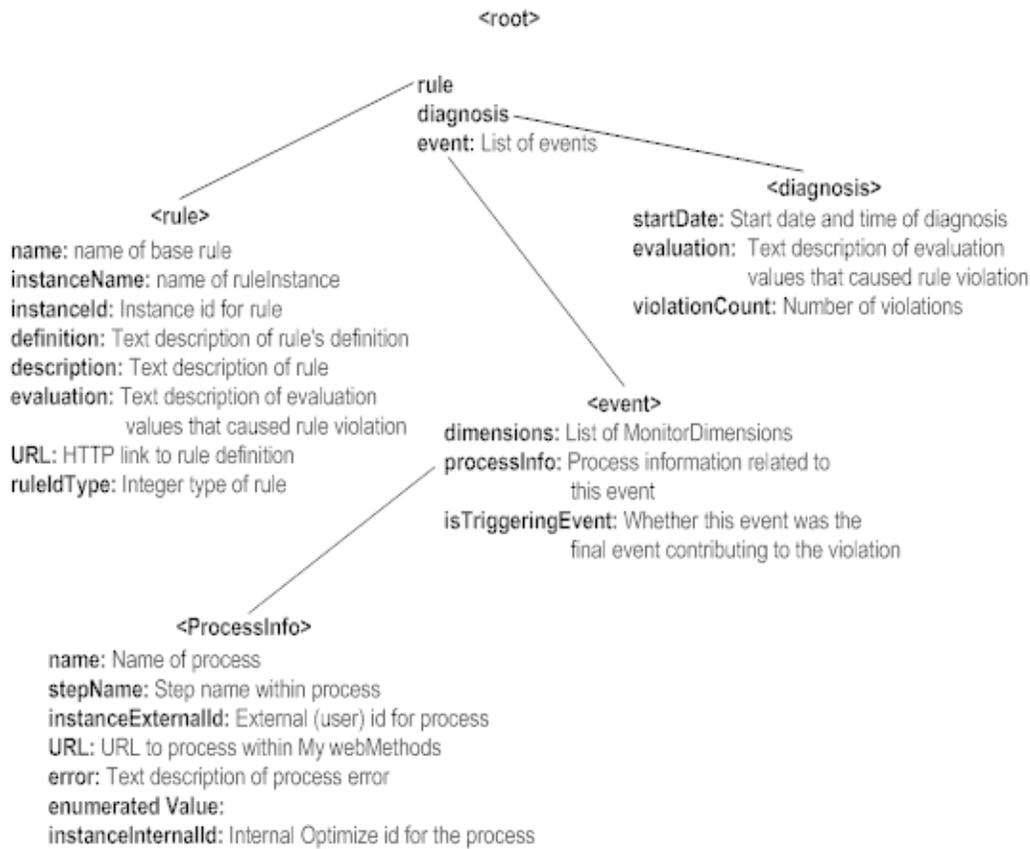
6. Click **Save**.
7. Click **Finish**.
8. Put the changes into effect. See "[Changing Your Configuration Settings](#)" on page 57.

Creating a Dynamic Alert Template

Dynamic alert templates use the Velocity open-source templating framework. Using dynamic variable resolution, Velocity allows flexible generation of text and creation of templates using any context-related data elements instead of a fixed set of previously available variables. That is, instead of requiring that the variable `${rule.name}` be explicitly defined, Velocity attempts to resolve the attribute "name" from the rule element within the context. If this variable cannot be resolved, the variable will appear verbatim in the generated e-mail, indicating that the substitution failed.

The Action Context

The context is an immutable representation of the rule's violation state. This context contains the following information related to the cause of the rule's violation:



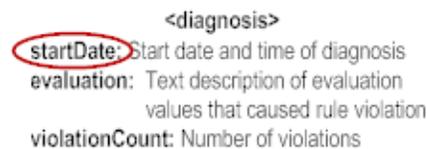
An example substitution:

`#{rule.name}`

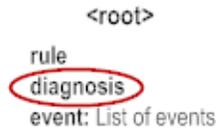
This substitution causes Velocity to look up a property/entity located at the <root> of the context named **rule**, then within the <rule> entity, to locate a property named **name**. The process ends when the last element in the <property>.<property> sequence is resolved. If the element cannot be resolved, the substitution text remains in the output.

Producing a substitution for a template is a simple matter of determining the required information and navigating to it.

For example, if you want to output the start date of the diagnosis, you would first find the element you want (in this case, startDate of the diagnosis).



and then you would navigate backward to the root of the context.



Once you have the variable substitution configured, simply surround it within `{ }` and use it within the template.

```
{diagnosis:startDate}
```

For further details on advanced formatting (looping, conditionals, etc.) refer to the Velocity documentation at <http://velocity.apache.org/engine/devel/getting-started.html>.

Important: A dynamic custom alert template file must end with a `.vm` extension for the template to be recognized and processed correctly. See "[Specifying E-mail Alert Settings and Templates](#)" on page 58 for instructions on specifying a custom template.

Example

The following example is the dynamic Velocity template for the default template used by Optimize to generate e-mail alerts. The `[SUBJECT]` and `[BODY]` tags denote the two currently supported sections of this template. Each section defines a separate Velocity template that uses the same action context.

```

[SUBJECT] $rule.instanceName
[BODY]
-----
Rule Details:
Name:      ${rule.name}
Instance name: ${rule.instanceName}
Description:  ${rule.description}
Status:     ${rule.status}
Severity:    ${rule.severity}
Definition:  ${rule.definition}
Evaluation:  ${rule.evaluation}
URL:        ${rule.URL}
Rule type:   ${rule.ruleTypeString}
SLA:        ${rule.sla}
Customer:    ${rule.customer}
-----
Diagnosis Details:
Start date:    ${diagnosis.startDate}
End date:      ${diagnosis.endDate}
Violation count: ${diagnosis.violationCount}
Violation id:  ${diagnosis.violationId}
-----
#foreach ($mapName in $context.getEvents().keySet())
#if ($rule.ruleTypeId == 1)
Event Details for Event Map: $mapName
#else
Event Details for KPI: $mapName
#end
#foreach ($event in $context.getEvents().get($mapName))
#if ($event.getDimensions().size() == 0)
Monitor dimensions: N/A
#else

```

```

Monitor dimensions:
#foreach ($dim in $event.dimensions)
$dim.dimensionName: $dim.dimensionValue
#end
#end
Attributes:
#foreach ($attrKey in $event.getAttributes().keySet())
#foreach ($attrValue in $event.getAttributes().get($attrKey))
$attrKey: $attrValue
#end
#end
#if ($event.getProcessInfo().size() > 0)
Number of Processes: $event.getProcessInfo().size()
#foreach ($procInfo in $event.processInfo)
#if ($procInfo.name)
Process info:
Name:      $procInfo.name
Id:        $procInfo.instanceExternalId
#if ($procInfo.stepName)
Step name: $procInfo.stepName
#end
URL:       $procInfo.getURL()
#else
Process info: N/A
#end
#end
#end
#end
#end
-----

```

For an example e-mail alert generated from this template, see Chapter 7, "Responding To Problem Alerts" in *webMethods Optimize User's Guide*.

Creating a Static Custom Alert Template

You can also create an alert template using a simple text document with a fixed set of substitution variables. This document contains elements for the e-mail alert subject and body. You specify the format of the e-mail using a combination of plain text and substitution variables.

When an alert is formatted, the Analytic Engine attempts to retrieve the template associated with the rule in the Mail Settings panel (see ["Specifying E-mail Alert Settings and Templates" on page 58](#)). If a template is found, the alert is formatted using the specified template. If no template is specified, or the specified template is not found, the alert is formatted using the default template.

As the subject and body of the alert are formatted, any substitution variables contained in the template are expanded. When used initially, these templates are stored in the Analytic Engine to reduce the overhead associated with file loading. If the template file is modified, you must restart the Analytic Engine (see ["Starting Optimize Manually" on page 40](#)) for the changes to take effect.

Example

The following is an example e-mail alert template. Note that the template is for a monitor alert. The substitution variables are explained in "Specifying Substitution Variables" on page 64.

```

Business Rule Violation - Template
-----
[SUBJECT]
Rule: ${Rule.Name} ${Diagnosis.Attributes}
[BODY]
Rule Details:
Rule Name:           ${Rule.Name}
Description:         ${Rule.Description}
Date / Time:         ${DateTime}
Rule Definition:     ${Rule.Definition}
Rule Evaluation:     ${Rule.Evaluation}
Rule SLA:            ${Rule.sla}
Rule Customer:       ${Rule.customer}
Rule Severity        ${Rule.severity}
Rule Status          ${Rule.status}
Out of Compliance:  ${Violation.Count}
View Rule:           ${Rule.URL}
Diagnosis Attributes:
${Diagnosis.Attributes}
Associated Text:
${Data}

```

Specifying Substitution Variables

You can use substitution variables to access the rule definition and diagnosis information. Variables also provide access to process information when an alert on a process event is triggered. These variables are case sensitive and must be specified exactly as defined below. If a variable cannot be found, the Analytic Engine will substitute a value of VARIABLE_NOT_FOUND in its place in the resulting e-mail alert.

Variable	Description
<i>`\${Data}`</i>	An iteration of associated business or process data.
<i>`\${Data.<Field>}`</i>	The value of the associated data named<Field>. See the DISPLAY NAME column of the Add/EditEvent Map page (Applications > Administration> Analytics > KPIs > Business Data , and click the name of the event map) for possible attributes to include. Results may vary depending on rule expression settings and evaluation conditions. Contact Global Consulting Services for assistance.
<i>`\${DateTime}`</i>	The date and time of the rule violation.
<i>`\${Diagnosis.Attributes}`</i>	An iteration of rule diagnosis attributes.

Variable	Description
<i>#{Process.Event}</i> *	The process event that triggered this alert.
<i>#{Process.InstanceId}</i> *	The process instance identifier.
<i>#{Process.Name}</i> *	The name of the process.
<i>#{Process.StepName}</i> *	The current step in the process.
<i>#{Process.URL}</i> *	The URL of the process instance in the user interface.
<i>#{Rule.Definition}</i>	The rule definition.
<i>#{Rule.Description}</i>	The description of the rule.
<i>#{Rule.Evaluation}</i>	The rule evaluation result.
<i>#{Rule.Name}</i>	The name of the rule.
<i>#{Rule.URL}</i>	The URL of the rule definition in the user interface.
<i>#{Rule.sla}</i>	Whether the rule is associated with a Service Level Agreement (SLA), Yes or No.
<i>#{Rule.customer}</i>	A customer name associated with the rule.
<i>#{Rule.severity}</i>	The severity label assigned to the rule.
<i>#{Rule.status}</i>	The status of the rule. Rule Violated = rule is in violation. In Compliance = rule has gone back into compliance.
<i>#{Violation.Count}</i>	The number of times the rule is out of compliance.

* Availability of the *#{Process.*}* substitution variables depends on the context of the rule you are accessing.

Examples:

- *#{Process.Event}*: To access this variable within a template, the rule must include the Error Type in the KPI or event rule definition. For example, a rule specified for the

Intrinsic Process Metrics KPIs could be "Error Count by Process and Error Type" or "Error Count by Step and Error Type".

- $\{Process.Step\}$: For this variable to be available, the KPIs or event rules must include the process step dimension.

If the variable cannot be substituted, a message indicating the problem is displayed in place of these variables.

Configuring SNMP Alerts

Typically, Optimize sends alerts about rule violations by e-mail. Alternatively, you can configure Optimize to send alerts using SNMP. With SNMP alerts, when a rule is violated the Analytic Engine sends the SNMP alert to the SNMP manager associated with the rule.

The Analytic Engine sends the SNMP alert named `webMethodsAlert` as an enterprise-specific alert consisting of two OCTET string type variables, one containing the name of the rule that was violated and one containing the monitor ID. The SNMP manager can retrieve the subject and body strings according to the structure defined in the Optimize Management Information Base (MIB) file `WEBMETHODS-COMMON-MIB.txt`, located in the `Optimize_directory\analysis\conf\MIB` directory.

You configure SNMP settings through the My webMethods Define environments page.

To configure settings for sending SNMP alerts

1. In My webMethods, navigate the following path: **Applications > Administration > System-Wide > Environments > Define Environments**.

The Define Environments page is displayed.

2. Click the name of the configured environment for which you want to <task>.

The Edit Environment page is displayed.

3. Click the **Configure Servers** tab.

4. On the **CONFIGURATION** tree, find the Analytic Engine for which you want to configure SNMP alert settings, and click **SNMP Alert Settings** under the name of the Analytic Engine.

The SNMP Alert Settings page is displayed.

5. Identify the SNMP manager to receive the alert by uncommenting and editing this XML:

```
<properties>
<!-- (Remove this line to uncomment the file.)
<property name="SNMPManager">
  <string meta="managerName">SNMP_Manager_Host1</string>
  <string meta="host">host1</string>
  <integer meta="port">161</integer>
  <property name="community">
    <string meta="handle">passwordHandle</string>
    <string meta="password">password</string>
  </property>
</property>
```

```
--> (Remove this line to uncomment the file.)
</properties>
```

<u>For...</u>	<u>Substitute the...</u>
<i>SNMP_Manager_Host1</i>	Name of the SNMP manager you want to receive alerts. This value is a symbolic name and can be any string denoting the SNMP manager host. This name also appears on the Rule Editor page when configuring the SNMP alert.
<i>host1</i>	Actual host name of the SNMP manager to receive the alert. Do not specify <code>localhost</code> ; you must use the host name or IP address of the SNMP manager.
<i>161</i>	Port number of the SNMP manager to receive the alert.
<i>passwordHandle</i>	Handle used for identifying the password specified in the password string. Use <code>SNMPCommunityHandle</code> if you will only specify one password for SNMP Alert settings, or specify a unique handle for each password if multiple passwords will be used.
<i>password</i>	Specify a password. Must be accompanied by a user name. Once this settings file is saved and processed, the password is encrypted, and it is displayed only as asterisks.

If a substitution cannot be made, an empty string is used instead.

Optimize displays the SNMP managers you identified in the SNMP Alert Settings panel on the Edit Rule page. For more information about selecting an SNMP manager on this page, see *webMethods Optimize User's Guide*.

6. If you want to identify more than one SNMP manager to receive alerts, copy and edit the XML shown in step 5 for each SNMP manager.
7. Click **Save**.
8. Apply the changes by re-deploying your environment (see "[Changing Your Configuration Settings](#)" on page 57) and restarting the Analytic Engine as described in "[Starting Optimize Manually](#)" on page 40.
9. Make sure the new settings work, as follows:
 - a. In My webMethods, navigate the following path: **Applications > Administration > Analytics > Rules > Rule List**.
 - b. On the Rule List page, click **Create Rule**.

- c. In the **Actions** area on the Edit Rule page, click **Add Action**. Ensure that the **Action Name** list contains the SNMP managers you added to the SNMP Alert Settings panel.

If the **Action Name** list does not contain the SNMP managers you identified, check the syntax of the XML in the SNMP Alert Settings panel.

Configuring Web Service Actions

For information on configuring Web service actions, see "[Setting Up Web Service Actions](#)" on page 281.

Changing Subcomponent Communication Settings

During configuration, you specified the ports that Optimize uses to communicate with its subcomponents and with My webMethods Server on the Define Environments page. Use this procedure any time you need to change the port settings after initial configuration.

Changing Analytic Engine or Web Service Data Collector Port Settings

To change the port numbers Optimize uses to communicate with the Analytic Engine or Web Service Data Collector, you must change them on their host systems. You can do this through the My webMethods Define Environments page. Remember that you must use the System Settings page to change the port numbers that My webMethods uses to communicate with Analytic Engines or Integration Servers.

Note: Refer to "[Changing Infrastructure Data Collector Ports](#)" on page 190 in this guide for instructions on how to change the Infrastructure Data Collector primary port.

To change Analytic Engine or Web Service Data Collector port settings

1. In My webMethods, navigate the following path: **Applications > Administration > System-Wide > Environments > Define Environments**.

The Define Environments page is displayed.

2. Click the name of the configured environment for which you want change port settings.

The Edit Environment page is displayed.

3. Click the **Map Endpoints** tab.
4. On the **CONFIGURATION** tree, find the Analytic Engine or Web Service Data Collector for which you want to change the port, and edit the port number.
5. Click **Save**.

6. To specify the port number My webMethods uses to communicate with the Analytic Engine, navigate to the My webMethods System Settings page: **Applications > Administration > My webMethods > System Settings**.

The information you enter in the **Server Settings** panel on the System Settings page depends on the type of environment you have. See "[Configuring Optimize to Monitor Multiple Integration Servers and Analytic Engines](#)" on page 93 for details.

7. Put your settings into effect. See "[Changing Your Configuration Settings](#)" on page 57.

Changing Analytic Engine or Web Service Data Collector Attributes for Web Services

These settings must be changed on the Analytic Engine host systems. You can do this with the My webMethods Define Environments page.

To change the attributes that the Analytic Engines or Data Collectors use to run Web services on other components

1. In My webMethods, navigate the following path: **Applications > Administration > System-Wide > Environments > Define Environments**

The Define Environments page is displayed.

2. Click the name of the configured environment for which you want to change web service attributes.

The Edit Environment page is displayed.

3. Click the **Map Endpoints** tab.
4. On the **CONFIGURATION** tree, find the Analytic Engine or Data Collector for which you want to change Web service attributes.
5. Edit the text boxes in the Incoming Connections column of the **WS Registry** row under the name of the Analytic Engine or Data Collector. Specify the same host and port you specified in "[Changing Analytic Engine or Web Service Data Collector Port Settings](#)" on page 68.
6. Click **Save** and **Finish**.
7. Put your settings into effect. See "[Changing Your Configuration Settings](#)" on page 57.

Changing My webMethods Host and Port Communication Attributes

To change the host names and port numbers that My webMethods uses to communicate with Optimize and Integration Server

1. In My webMethods: **Applications > Administration > My webMethods > System Settings**

Optimize displays the System Settings page.

2. In the **Server Settings** panel, change the host names and port numbers of the Integration Server and Analytic Engines as necessary.

The information you enter in the **Server Settings** panel on the System Settings page depends on the type of environment you have. See "[Configuring Optimize to Monitor Multiple Integration Servers and Analytic Engines](#)" on page 93 for details.

3. If you want My webMethods Server to connect to these subcomponents using SSL, select the **Use SSL** check box.
4. Click **Save**.
5. If you change the host name or port number here, you must change it in webMethods Central Configuration as well. See "[Changing Analytic Engine or Web Service Data Collector Port Settings](#)" on page 68.

Importing a Certification Authority (CA) Certificate

If you selected the **Use SSL** check box in step 3 of "[Changing My webMethods Host and Port Communication Attributes](#)" on page 69, you must import the issuing Certification Authority (CA) certificate into your keystore, as described below:

To import the issuing CA certificate into your keystore

1. On the Optimize Analytic Engine host system, open a command prompt.
2. Navigate to the `jvm/win150/lib/security` directory.
3. Import the `cacerts` file using the following command:

```
keytool -import -v -keystore cacerts -file cacert.der -alias MYPORTALCA
```

The default password for the `cacerts` file is `changeit`. If you are not authorized to import this file using the default password, contact your system administrator.

Tip: For more information about the `cacerts` file, see <http://download.oracle.com/javase/1.4.2/docs/tooldocs/windows/keytool.html>.

4. List the `cacerts` file to test the import step using the following command:

```
keytool -list -keystore cacerts
```

Defining Logical Names for the Analytic Engines and Web Service Data Collectors

You define names for your logical servers when you configure your environment (see "[Configuring and Deploying Your Environment](#)" on page 54). Use the following procedure to redefine logical names for your Analytic Engines or Web Service Data Collectors.

To define logical names

1. In My webMethods, navigate the following path: **Applications > Administration > System-Wide > Environments > Define Environments**
The Define Environments page is displayed.
2. Click the name of the configured environment for which you want to define logical server names.
The Edit Environment page is displayed.
3. Click the **Design Servers** tab.
4. Click  **Edit**, or the name of the logical server you want to redefine. The Edit Logical Server dialog is displayed.
5. Enter a new name in the **Logical Server Name** field, and a new description in the **Description** field.
6. Click **Save** and **Finish** to save changes.

Securing Optimize Engines with SSL

This section contains information about configuring Optimize engines for SSL. If you are configuring an entire system, depending on your configuration and security goals, you may also need to configure other components such as Broker Servers, Infrastructure Data Collector and Integration Servers.

Depending on your security requirements and goals, you may need to secure other Software AG components, such as Broker Server and Integration Server. For information about securing Broker Server for SSL, see *Administering webMethods Broker*. For information about securing Integration Server for SSL, see *webMethods Integration Server Administrator's Guide*.

Configuring Optimize Web Services for SSL

In order to secure Optimize for SSL, you must configure all of the following components: Analytic Engine and the Web Services Data Collector. For each component, you must set up the appropriate SSL KeyStore and TrustStore in the GlueSSLProperties.xml file, and you must change the Configuration Agent protocol to https in the EndpointRegistry.xml file. The following procedure assumes that you have obtained the appropriate KeyStore and TrustStore files. For more information on installing KeyStore and TrustStore certificates, see *Administering My webMethods Server*.

Also, this procedure assumes that you have configured the Key and Trust stores for the Central Configuration back end web-application. Refer to the "Security" chapter of *Configuring BAM* for more information. Completing this procedure enables secure communication between the Central Configuration portlets and the back end web-application.

Configuration of the Analytic Engine and WS Data Collector are identical. Configuring the Infrastructure Data Collector is a different process and is explained in the “Configuring Infrastructure Data Collector” chapter of *Administering webMethods Optimize*.

Note that the Configuration Agent port for the Web Service Data Collector is 15001.

To configure an Optimize engine for SSL:

1. Copy the appropriate Key and Trust store files to the `conf\security\ssl` folder.
2. Open the `<install directory> \conf\glue\GlueSSLProperties.xml` file and edit the entry for the Key and Trust store file to point to the files you copied in the preceding step.
3. Save your changes to the `GlueSSLProperties.xml` file and close it.

There are several ways to verify SSL configuration. If you are running the Analytic Engine as a console application on a Windows server, you can check the console window. You should see the following messages in this window.

```
[STARTUP] Glue 8.0 Fix 2 build 3
[STARTUP] soap/http server started on https://<server name and
domain>:15000/services
[STARTUP] soap/http server started on https://<server name and
domain>:12503/services
```

Changing the Configuration Agent Protocols

After you have installed authentication certificates, the next step is to change the configuration agent endpoint protocols to use SSL.

To change endpoint protocols to use SSL

1. Shut down all Optimize components.
2. For each of the Optimize components (Analytic Engine and Web Service Data Collector), navigate to the `\conf\System` directory and edit the file named `EndpointRegistry.xml`. Change the protocol for the Configuration Agent to `https`, as shown in the following example:

```
<endpoint host="localhost" name="Configuration Agent" port="15000"
protocol="https"/>
```

3. Restart all Optimize components.
4. Log in to My webMethods Server and navigate to the Define Environments page. Select the desired environment and navigate to the Map Endpoints page. On that page, change the protocol for the Configuration Agent endpoints to `https`.
5. Navigate to the main page and select the environment for deployment. Select the option labeled **Deploy Updates**. If the endpoints are defined correctly, you should see a message indicating that the deployment was successful. If there is an error, verify that the endpoints are correct and that all of the Optimize components started correctly.

Securing the Optimize Analytic Engine Web Services

The webMethods Optimize Analytic Engine exposes web services used by the user interfaces deployed in My webMethods Server. By default these services are bound to port 12503 and the http protocol. This section describes how to change the web services protocol to "https". These instructions assume that you have configured the Analytic Engine's Key and Trust Stores as described in ["Configuring Optimize Web Services for SSL" on page 71](#), and that you have changed the Configuration Agent Endpoint protocol to enable SSL as described in ["Changing the Configuration Agent Protocols" on page 72](#). Once the Certificate configuration has been completed the Analytic Engine's Web Service Registry Endpoint protocol must be changed to "https".

To update the Analytic Engine Web Service Protocol to use HTTPS:

1. Shut down the Analytic Engine.
2. Open the `<install directory> \optimize \analysis \conf \system \EndpointRegistry.xml` file.
3. Locate the Web Service Registry (WS Registry) Protocol configuration, and change the setting from "http" to "https".
4. Start the Analytic Engine.

After starting the Analytic Engine you should see the following message:

```
[STARTUP] Glue 8.0 Fix 2 build 3
[STARTUP] soap/http server started on https://<server
name>.ame.ad.sag:15000/services
[STARTUP] soap/http server started on https://<server
name>.ame.ad.sag:12503/services
```

After the endpoint Registry has been manually modified, any Central Configuration environment that includes the updated Analytic Engine Logical Server should be updated and its configuration redeployed to keep both consistent. This procedure is analogous to what was done to update the Configuration Agent endpoint in ["Changing the Configuration Agent Protocols" on page 72](#). For more information, see the instructions on Mapping Endpoints in *Configuring BAM*.

Note: Now that the protocol has been changed for the Analytic Engine Web Service Registry, it is necessary, at the very least, that the Trust Store be configured in the My webMethods product. If you do not set up the trusted keys in My webMethods Server, all web service invocations to the Analytic Engine will fail.

Securing the Optimize Web Service Data Collector Services

The Optimize Web Service Data Collector Agent provides a web service-based interface that allows external applications to push data into Optimize. In general, these web services are good candidates to remain open and unencrypted. If desired, however, SSL encryption can be enabled just like the Analytic Engine Web Service Registry. The steps are identical with the exception of the specified Key Store and endpoint changes.

For instructions on configuring the Key Store please see the preceding section. In this example the Key Store file is "wsdatacollector.p12". The Trust Store is consistent and, by default, always configured as "sagCA.jks". Once the Key and Trust Stores have been configured the Web Service Registry protocol can be changed.

To secure the Optimize WS Data Collector Services:

1. Open the endpointRegistry.xml files located in the in the following folders:

```
<install directory> \optimize\dataCollector\conf\glue
```

```
<install directory> \optimize\dataCollector\conf\system
```

2. Locate the Web Service Registry (WS Registry) Protocol configuration.

By default, the lines that need to be changed will look similar to the following:

```
<?xml version="1.0" encoding="UTF-8"?><endpoint-registry> <endpoint
host=<hostname> name="WS Registry" port="12603" protocol="http"/>
```

3. Change the protocol setting in each file from "http" to "https".
4. Save the files and close them.

Configuring the Optimize Engines to Communicate with a Secured Broker Server

In the same way a Broker identity needed to be created in My webMethods Server to communicate with an SSL enabled Broker Server, the Optimize Engines need to be configured to trust the Broker Server certificate and to present their own Certificate(s) for authentication. This configuration can be performed using the My webMethods Define Environments page.

To configure Optimize Engines to communicate with a Secured Broker Server:

1. In My webMethods, navigate the following path: **Applications > Administration > System-Wide > Environments > Define Environments**
2. Select the environment that was created in the initial installation of the webMethods Optimize platform and select the Configure Servers tab. The SSL properties to use when connecting to the Broker Server are contained in the JNDI Configuration settings. If a single key and trust store is required for the environment you can configure the JNDI settings under Default Settings.

If you are using individual certificates (keystores) for each webMethods component, you must configure a JNDI configuration for each Optimize engine. Expand the logical server "Analytic Engine" and clear the "Use Default" indicator for the JNDI Configuration. The JNDI properties should now be displayed on the right hand pane. Check the Enable SSL and Encryption flags. Enter the location of the key store that identifies the Analytic Engine. Set the type to PKCS12 and enter the appropriate Distinguished Name. Enter the location of the trust store and the trust stores type. Finally enter the master password used when creating the key and trust stores. Save the configuration.

3. Repeat the process for the Web Service Data Collector and the Infrastructure Data Collector. Use the appropriate keystore and distinguished name for each server.
4. Deploy the changes to the environment and restart the various Optimize components. The restart is required to establish the secured connection to the Broker Servers. Refer to *Configuring BAM* for information about deploying and environment.

Updating the Configuration Settings in the Central Configurator

At this point the certificates and SSL setup should be complete for each of the Optimize components. These include the Analytic Engine, Infrastructure Data Collector and the Web Service Data Collector. Because the certificate setup and protocol change required manual updates, the components are not in sync with the Central Configurator. This next step simply updates the protocol information for each of the Optimize components and deploys out an "official" copy of the EndpointRegistry.xml configuration file.

To update the Configuration Settings in the Central Configurator:

1. In My webMethods, navigate the following path: **Applications > Administration > System-Wide > Environments > Define Environments**
2. Open the configuration for the applicable environment.
3. Select the Map Endpoints tab and change the protocol field under "Incoming Connections" for each of the Optimize components "Configuration Agent" WS endpoint to "https".
4. Click **Save** and then **Finish**.

You will be returned to the main Define Environments page. This page should show that the "Deployed" status has been updated to indicate that the configuration has been changed.

5. Click the green arrow on the right under "Actions" to navigate to the Deploy Configuration Files page.
6. Click the **Deploy Updates** button to push the configuration changes out to all of the Optimize components.

You should see that the "System Endpoint Registry" configuration has been deployed to each of the Optimize components.

Enabling Data-Level Security

This section applies only to Optimize for Process. When Data-Level Security (DLS) is enabled, access to KPIs and business processes is controlled by user role assignments. DLS is disabled by default, and when DLS is enabled, all roles are denied access to KPIs or business processes by default. For users to have access to KPIs or business processes after DLS is enabled, access must be assigned through their roles. For instructions on assigning users access to KPIs, see ["Granting User Access to KPIs" on page 177](#). For

instructions on assigning users access to business processes, see "[Granting User Access to Business Processes](#)" on page 136.

Note: Currently, Data-Level Security is supported only in a single server environment.

Note: If you have a BPMS environment and are using DLS, the Analytic Engine must be configured for DLS according to the following procedure, and the WmMonitor package must be configured for DLS in Integration Server. Refer to the *webMethods Monitor User's Guide* for WmMonitor DLS configuration instructions.

To enable Data-Level Security

1. In My webMethods, navigate the following path: **Applications > Administration > System-Wide > Environments > Define Environments**
The Define Environments page is displayed.
2. Click the name of the configured environment for which you want to enable DLS.
The Edit Environment page is displayed.
3. Click the **Configure Servers** tab.
4. On the **CONFIGURATION** tree, find the Analytic Engine for which you want to enable DLS, and click **Station Settings** under the name of the Analytic Engine.
5. Do one or more of the following:
 - Select **DLS Enabled** to enable DLS for this Analytic Engine.
 - Select **DLS Cache Enabled** to enable DLS cache for this Analytic Engine.
 - Select **SAML Enforced** to enforce SAML for this Analytic Engine.
6. Click **Save**.
7. Put your settings into effect. See "[Changing Your Configuration Settings](#)" on page 57.

Changing the Optimize Authentication Credentials

Optimize provides default authentication credentials for services, such as the OptimizeDeployer Service that enables you to deploy Optimize assets, and the `pub.optimize.mashzone.adapter.retrieveData` service from the WmOptimize package that enables you to retrieve data from the Analytic Engine and display it in Software AG MashZone. You can change these credentials, if desired, using the following procedure.

Note: When you first change these credentials, the new password is not encrypted; however, once the Analytic Engine is restarted, the password is encrypted. It is advisable that you keep a record of the credentials should you need to refresh your memory of them later.

To change the Optimize authentication credentials

1. Shut down the Analytic Engine.
2. Locate and open the *Software AG_directory /optimize/analysis/conf/security/glue/acl.xml* file using an appropriate text editor.
3. Locate the <name> attribute in the acl.xml file for the desired service and edit it as appropriate. As a reference, the default settings are shown in the following example.

```
<acl>
  <user>
    <name>Administrator</name>
    <password>GlueRealmPasswordHandle::0</password>
    <role>DeployAdministrator</role>
    <role>DataAccessAdministrator</role>
  </user>
</acl>
```

4. Locate the <password> attribute in the acl.xml file and edit it as desired.
5. Save and close the acl.xml file. Be sure to make note of your user name and password settings.
6. Start the Analytic Engine.

Adding New Optimize Credentials

Use the following procedure if you want to add new Optimize credentials. This procedure describes how to copy an existing set of credentials and edit them to create a new user.

To add new Optimize authentication credentials

1. Shut down the Analytic Engine.
2. Locate and open the *Software AG_directory /optimize/analysis/conf/security/glue/acl.xml* file using an appropriate text editor.
3. Locate the <user> section in the acl.xml file and copy it.
4. Paste the copy of the existing user section under the existing user section, and edit the <name> and <password> elements as desired.

Do not modify or delete the <role> elements.

As a reference, the modified settings must look as shown in the following example.

```
<acl>
  <user>
    <name>Administrator</name>
    <password>GlueRealmPasswordHandle::0</password>
    <role>DeployAdministrator</role>
    <role>DataAccessAdministrator</role>
  </user>
  <user>
    <name>MyUser</name>
    <password>MyPassword</password>
    <role>DeployAdministrator</role>
```

```

    <role>DataAccessAdministrator</role>
  </user>
</acl>

```

5. Save and close the acl.xml file. Be sure to make note of your user name and password settings.
6. Start the Analytic Engine.

Specifying Statistical Intervals

Optimize uses *statistical intervals* to generate statistics and evaluate rules. A statistical interval is a period of time from which the Analytic Engine takes collected data samples and creates common statistical values such as average, mean, and standard deviation.

You can group the days of the week and the minutes in the day to create statistical intervals. For example, you can choose to generate a new average, mean, and standard deviation for each type of data every work week (Monday through Friday). You can choose to generate a new average, mean, and standard deviation for each type of data every 12 hours.

To set statistical intervals

1. In My webMethods, navigate the following path: **Applications > Administration > System-Wide > Environments > Define Environments**
The Define Environments page is displayed.
2. Click the name of the configured environment for which you want to set statistical intervals.
The Edit Environment page is displayed.
3. Click the **Configure Servers** tab.
4. On the **CONFIGURATION** tree, find the Analytic Engine for which you want to set statistical intervals, and click **Monitor Behavior Settings**.
5. Set the `defaultDays` attribute as follows:

If you want to...	Set to...	Result
Group all days of the week into one group.	<code>all</code>	Optimize will generate statistical values from the data collected during the entire group of days to use for rules and fingerprints on each day.
Divide the days of the week into work days (Monday through Friday) and weekend	<code>work</code>	Optimize will generate statistical values from the data collected on work days to use for rules and fingerprints on each work day. Optimize will also generate statistical values from the data collected on the weekend days to use for rules

<u>If you want to...</u>	<u>Set to...</u>	<u>Result</u>
days (Saturday and Sunday).		and fingerprints on each weekend day. This is the default.
Not group any days.	day	Optimize will generate statistical values from the data collected on each day of the week to use for rules and fingerprints on the applicable day. For example, Optimize will generate statistical values from data collected on Monday to use for rules and fingerprints on Mondays.

6. Click **Save** and **Finish**.
7. Put your settings into effect. See ["Changing Your Configuration Settings" on page 57](#).

Optimizing Analytic Engine Performance

If the Analytic Engine is not performing as expected (for example, database queues are backing up, the Analytic Engine is consuming too many CPU resources, or Analytic Engine disk space is filling up too fast), you can adjust its settings for buffering events and caching process data.

To adjust Analytic Engine settings

1. Adjust Analytic Engine Analysis Engine settings as follows:
 - a. In My webMethods, navigate the following path: **Applications > Administration > System-Wide > Environments > Define Environments**
The Define Environments page is displayed.
 - b. Click the name of the configured environment for which you want to adjust analysis engine settings.
The Edit Environment page is displayed.
 - c. Click the **Configure Servers** tab.
 - d. On the **CONFIGURATION** tree, find the Analytic Engine for which you want to adjust analysis engine settings, and click **Analysis Engine Settings** under the name of the Analytic Engine.

<u>Attribute</u>	<u>Definition</u>
Tolerance	Default = 1.0.

Attribute	Definition
Trending Threshold	Default = 3.
Trending Tolerance	Default = 0.005

- e. Click **Save** and **Finish** to save your changes.
 - f. Put your changes into effect. See "[Changing Your Configuration Settings](#)" on page 57.
2. Adjust Analytic Engine cache configuration settings as follows:

- a. In My webMethods, navigate the following path: **Applications > Administration > System-Wide > Environments > Define Environments**.

The Define Environments page is displayed.

- b. Click the name of the configured environment for which you want to adjust cache configuration settings.

The Edit Environment page is displayed.

- c. Click the **Configure Servers** tab.
- d. On the **CONFIGURATION** tree, find the Analytic Engine for which you want to adjust cache configuration settings, and click **Cache Configuration Settings** under the name of the Analytic Engine.

Note: You can also adjust cache configuration settings for all logical servers in the environment by selecting Cache Configuration in the Default Settings section of the Configuration tree.

- e. Click **Save** and **Finish** to save your changes.
 - f. Put your changes into effect. See "[Changing Your Configuration Settings](#)" on page 57.
3. Adjust Analytic Engine data maintenance settings as follows:
- a. In My webMethods, navigate the following path: **Applications > Administration > System-Wide > Environments > Define Environments**
The Define Environments page is displayed.
 - b. Click the name of the configured environment for which you want to adjust data maintenance settings.
The Edit Environment page is displayed.
 - c. Click the **Configure Servers** tab.
 - d. On the **CONFIGURATION** tree, find the Analytic Engine for which you want to adjust data maintenance settings, and click **Data Maintenance Settings** under the name of the Analytic Engine.

Attribute	Definition
BusinessDaysToRetain	Number of days to retain system event data in the Process Tracker database component. After the specified number of days, the data is eligible to be purged by the purge procedures. The default is 30 days.
dataMaintenanceInterval	How often to recalculate the values in the operations_parameter table. The default is four hours.
Aggregated Business Days to Retain	Number of days to retain aggregated business event data in the Analysis database component. Aggregated business data represents a consolidation of business event data that is used to improve the performance of the KPI Summary and KPI Instance Details graphs. Aggregated business data takes up much less space and therefore can be kept for a longer period of time without consuming excessive disk space or affecting system performance. The default is 365 days.

- e. Click **Save** and **Finish** to save your changes.
 - f. Put your changes into effect. See "[Changing Your Configuration Settings](#)" on [page 57](#).
4. Adjust Analytic Engine JNDI settings as follows:
 - a. In My webMethods, navigate the following path: **Applications > Administration > System-Wide > Environments > Define Environments**.
The Define Environments page is displayed.
 - b. Click the name of the configured environment for which you want to adjust JNDI settings.
The Edit Environment page is displayed.
 - c. Click the **Configure Servers** tab.
 - d. On the **CONFIGURATION** tree, find the Analytic Engine for which you want to adjust JNDI settings, and click **JNDI Configuration** under the name of the Analytic Engine.

Note: You can also adjust JNDI configuration settings for all logical servers in the environment by selecting JNDI Configuration in the Default Settings section of the Configuration tree.

Attribute	Definition
Broker Name	If webMethods Universal Messaging (the default) is specified in the Naming Factory Type field, this field should be empty. If Broker is specified as the JMS Server in the Naming Factory Type field, this field should list the appropriate Broker name.
Naming Factory Type	List the specified JMS server, either Broker or " webMethods Universal Messaging ". If webMethods Universal Messaging is specified, then the Broker Name field should be empty. If Broker is specified, then the Broker Name field should list the appropriate Broker server name.
Enable SSL	If the box is checked, the Analytic engine will use a secure connection, with correct configuration. The check box is unchecked by default and is not required.
Encryption	If the check box is selected, Analytic Engine will use an encrypted connection. The check box is unchecked by default and is not required.
Key Store File	If applicable, lists the absolute path to the private key files that is located on a server accessible by the Analytic Engine. This field is required only if SSL is enabled.
Key Store Type	Specifies the KeyStore type, either PKCS12 or JKS. This field is required only if SSL is enabled.
Distinguished Name	Specifies the name of the appropriate certificate identifier. This field is required only if SSL is enabled.
Trust Store File	If applicable, lists the absolute path to the CA certificate file on a server accessible by the Analytic Engine. This field is required only if SSL is enabled.
Trust Store Type	Specifies the TrustStore file type, either PKCS12 or JKS. This field is required only if SSL is enabled.

- | Attribute | Definition |
|------------------------------|---|
| Key and Trust Store Password | Shows the encrypted pass phrase that is used to access the specified Key and Trust Stores. This field is required only if SSL is enabled. |
- e. Click **Save** and **Finish** to save your changes.
 - f. Put your changes into effect. See ["Changing Your Configuration Settings" on page 57.](#)
5. Adjust Analytic Engine Process Tracker settings as follows:
 - a. In My webMethods, navigate the following path: **Applications > Administration > System-Wide > Environments > Define Environments**
The Define Environments page is displayed.
 - b. Click the name of the configured environment for which you want to adjust Process Tracker settings.
The Edit Environment page is displayed.
 - c. Click the **Configure Servers** tab.
 - d. On the **CONFIGURATION** tree, find the Analytic Engine for which you want to adjust Process Tracker settings, and click **Process Tracker Settings** under the name of the Analytic Engine.

- | Attribute | Definition |
|------------------------|--|
| Days To Retain Process | Number of days to retain business or system event data in the Process Tracker database component. After the specified number of days, the data is eligible to be purged by the purge procedures. The default is 60 days. |
- e. Click **Save** and **Finish** to save your changes.
 - f. Put your changes into effect. See ["Changing Your Configuration Settings" on page 57.](#)

Multi-NIC System Considerations

If you experience web service communication problems with a server attempting to connect to an Optimize system running within My webMethods, that is running on a multi-NIC system (also referred to as a multi-homed system), you should make the following configuration adjustment to the Analytic Engine wrapper.conf file to prevent the server from referencing the Optimize system via "localhost". This change applies only to customers who wish to bind their web services to a specific IP address, and it

should make resolution of the host via host name function properly on a multi-NIC system.

Note: Changing this setting could affect any system that uses the REST services using localhost.

To make this change, use the following steps.

1. Open the `wrapper.conf` file located in the following folder: *Software AG_directory / optimize/analysis/conf*
2. Locate the following line in the `wrapper.conf` file: `wrapper.java.additional.19=-Dopt.wm.glue.ignoreLocalHostCheckOnBinding=false`
3. Change the value for the `ignoreLocalHostCheckOnBinding` property to `true`, so that the line read as follows: `wrapper.java.additional.19=-Dopt.wm.glue.ignoreLocalHostCheckOnBinding=true`.
4. Save the file and close it.
5. Restart the Analytic Engine.

The Java Service Wrapper

The Java Service Wrapper is a utility program developed by Tanuki Software, Ltd that is used to launch the JVM in which Optimize runs. You can control various aspects of Optimize components using settings that control how the Java Service Wrapper runs. This section describes how to customize some of the settings used by the Java Service Wrapper.

In addition to launching the JVM, the Java Service Wrapper offers features for monitoring the JVM, logging console output, and generating thread dumps. The following sections describe how Optimize uses the features of the Java Service Wrapper. For an overview of the Java Service Wrapper, see the *Working with the webMethods Product Suite and the Java Service Wrapper* cross-product document.

There are three Optimize components for which Java Service Wrapper configuration settings can be adjusted. These components are Analytic Engine, Web Service Data Collector, and Infrastructure Data Collector. There are numerous Java Service Wrapper settings that can be modified for these components, but Software AG only recommends modifications to only two parameters: log level and memory allocation. Other settings should not be customized unless you are directed to do so by Software AG Support.

The Java Service Wrapper Configuration Files

For applicable Optimize components, the configuration files for the Java Service Wrapper reside in the following folders.

- The Analytic Engine Java Service Wrapper configuration files resides in the following folder:

Software AG_directory/optimize/analysis/conf/wrapper.conf

- The Web Service Data Collector Java Service Wrapper configuration file resides in the following folder:

Software AG_directory/optimize/dataCollector/conf/wrapper.conf

- The Infrastructure Data Collector Java Service Wrapper configuration file resides in the following folder:

Software AG_directory/profiles/InfraDC/configuration/custom_wrapper.conf

When you start Optimize, property settings in one or both of the following files determine the configuration of the JVM and the behavior of the logging and monitoring features of the Java Service Wrapper.

File name	Description
wrapper.conf	Contains property settings that are installed by Optimize. You must edit this file directly to change Java Service wrapper settings for Analytic Engine and Web service Data Collector.
custom_wrapper.conf	For Infrastructure Data Collector, this file contains properties that modify the installed settings in wrapper.conf. If you need to modify the property settings for the Java Service Wrapper for Infrastructure Data Collector, make your changes in this file.

The following sections describe configuration changes that Optimize supports for Java Service Wrapper. For Optimize, do not make any configuration changes to the Java Service Wrapper other than the ones described in the following sections unless directed to do so by Software AG support.

JVM Configuration

When the Java Service Wrapper launches the JVM, it provides configuration settings that, among other things, specify the size of the Java heap, the size of the PermGen area, and the directories in the classpath.

JVM Configuration Properties

The `wrapper.java` properties in the Java Service Wrapper configuration files determine the configuration of the JVM in which Optimize runs.

The JVM property settings that Optimize installs are suitable for most environments. However, you can modify the following properties if the installed settings do not suit your needs. For procedures and additional information, see the webMethods cross-product document, *Working with the webMethods Product Suite and the Java Service Wrapper*.

Specifying the JDK or JRE for Analytic Engine, Infrastructure Data Collector and Web Service Data Collector

Optimize Analytic Engine, Optimize Infrastructure Data Collector and Web Service Data Collector must point to a JDK or JRE. By default, they point to the location of the JDK installed at the same time you installed Optimize. If necessary, you can specify a different location.

Before you specify the location of Java for Analytic Engine, Infrastructure Data Collector and Web Service Data Collector determine whether you need to specify the location of the JDK or the JRE.

Important: If you specify a different JDK or JRE, do *not* remove the JDK or JRE that Software AG Installer installed with Optimize. The JDK and JRE installed with Optimize are required to run the Software AG Uninstaller.

To specify the Java location for Analytic Engine, Infrastructure Data Collector and Web Service Data Collector

1. Open the wrapper.conf file in a text editor. You can find the respective wrapper.conf file in the following locations:
 - The Analytic Engine Java Service Wrapper configuration file resides in the following folder: *Software AG_directory/optimize/analysis/conf/wrapper.conf*
 - The Infrastructure Data Collector Java Service Wrapper configuration file resides in the following folder: *Software AG_directory/profiles/InfraDC/configuration/wrapper.conf*
 - The Web Service Data Collector Java Service Wrapper configuration file resides in the following folder: *Software AG_directory/optimize/dataCollector/conf/wrapper.conf*
2. Set the wrapper.java.command property so that it specifies the location of the JDK or JRE installation directory. For example:

```
wrapper.java.command=C:\SoftwareAG\jvm\jvm\bin\java
```
3. Save and close the file.
4. Restart Analytic Engine, Infrastructure Data Collector and Web Services Data Collector.

Configuring JVM Memory Allocation

Refer to the following table for information about setting the initial and maximum memory allocation for the JVM at startup.

Property	Value
<code>wrapper.java.initmemory</code>	Initial size (in MB) of the Java heap. For the Analytic Engine, the recommended value is 100
	<p>Note: If you specify a value for the <code>wrapper.java.initmemory</code> memory property, make sure that it is smaller than the value of the <code>wrapper.java.maxmemory</code> property.</p>
<code>wrapper.java.maxmemory</code>	Maximum size (in MB) to which the Java heap can grow. For Analytic Engine, the recommended value is 900.

To use the default values that are configured in the JVM itself, specify zero (0) in the properties of the `wrapper.conf` file. If you configure the memory amount in the Java wrapper (a non-zero value), the wrapper adds an appropriate `-Xms` parameter. However, if you configure the default JVM memory amount (a zero value), you can set the `-Xms` parameter manually as an additional property in the configuration file.

The Wrapper Log

The Java Service Wrapper records console output in a log file. The log contains the output sent to the console by the wrapper itself and by the JVM in which Optimize is running. The wrapper log is especially useful when you run Optimize as a Windows service, because console output is normally not available to you in this mode.

The Java Service Wrapper log files for Optimize components are located in the following locations:

- Analytic Engine: *Software AG_directory*\Optimize\analysis\bin\wraapper.log
- Web Service Data Collector: *Software AG_directory*\Optimize\dataCollector\bin\wrapper.log
- Infrastructure Data Collector: *Software AG_directory*\profiles\InfraDC\logs

To view a log file, open the log file in a text editor.

Logging Properties

The `wrapper.console` and `wrapper.log` properties in the wrapper configuration files determine the content, format, and behavior of the wrapper log.

The logging settings that Optimize installs are suitable for most environments. However, you can modify the following properties if the installed settings do not suit your needs. For procedures and additional information, see the webMethods cross-product document, *Working with the webMethods Product Suite and the Java Service Wrapper*.

Property	Value
<code>wrapper.console.loglevel</code>	Level of messages to display in the console.
<code>wrapper.console.format</code>	Format of messages in the console.
<code>wrapper.logfile</code>	File in which to log messages.
<code>wrapper.logfile.loglevel</code>	Level of messages to write in the log file.
<code>wrapper.logfile.format</code>	Format of messages in the log file.
<code>wrapper.logfile.maxsize</code>	Maximum size to which the log can grow.
<code>wrapper.logfile.maxfiles</code>	Number of old logs to maintain.
<code>wrapper.syslog.loglevel</code>	Level of messages to write to the Event Log on Windows systems or the syslog on UNIX.

Configuring Additional JVM Parameters

You can define and configure some additional Java parameters that are passed to the JVM when it is started. You define the additional parameters in the `wrapper.conf` file in the same way you configure memory allocation, as described in [The Java Service Wrapper](#) "The Java Service Wrapper" on page 84. Each additional parameter is defined in the form:

```
wrapper.java.additional.<n>=<parameter>
```

where:

- `<n>` is a numeric value, starting with 1 for the first additional parameter, and then incrementing by one for each consecutive parameter.
- `<parameter>` is the parameter name and value.

To configure additional JVM parameters

1. Browse to `Software AG_directory /optimize/analysis/conf/` and open the `wrapper.conf` file for editing.
2. Define the most frequently used additional parameters, as follows:

Parameter	Value
<code>wrapper.java.additional.<n>= -Xsssize</code>	256k

Parameter	Value
	Note Set this value to 512k for HP-UX 64-bit.
<code>wrapper.java.additional.<n>=-XX:PermSize=<i>value</i></code>	The default value for <code>PermSize</code> is set by the JVM. In the event of Perm out of memory errors, use this parameter to increase the <code>PermSize</code> value.
<code>wrapper.java.additional.<n>=-XX:MaxPermSize=<i>value</i></code>	The default value for <code>MaxPermSize</code> is set by the JVM. In the event of Perm out of memory errors, use this parameter to increase the <code>MaxPermSize</code> value.

3. Save the `wrapper.conf` file.
4. If the Analytic Engine is running, restart it to apply the modifications.

Configuring Analytic Engine High Availability

You can configure the Analytic JVM for automatic restart if the JVM stops unexpectedly, or experiences certain errors. You configure these settings by adding parameters to the `wrapper.conf` file in the same way you configure memory allocation, as described in ["The Java Service Wrapper" on page 84](#).

The Java Wrapper Service restarts the Analytic Engine in the following instances:

- The JVM terminates unexpectedly. This occurs automatically and you do not need to specify any parameters for this behavior.
- A thread dead lock is detected. Specify the deadlock interval parameter to define the interval at which dead lock detection is performed:
`wrapper.check.deadlock.interval=xxx` where `xxx` is the interval length in seconds. The default value is 60 seconds.
- A Heap or Perm out of memory error occurs. This is set by default and you do not need to specify any parameters for this behavior. It requires the additional parameter `-XX:+HeapDumpOnOutOfMemoryError` set by default as a `wrapper.java.additional.<n>=<parameter>` parameter.

Note: In case of an out of memory error, the JVM creates a heap dump in the `Software AG_directory/optimize/analysis/bin` directory (`java_pidXXXX.hprof`).

- The JVM does not respond - In normal operation, the Wrapper pings the JVM once every 90 seconds to make sure that its process has not frozen up. The ping timeout (300 seconds by default) is the amount of time to allow for a response from the JVM

before the Wrapper assumes that it is not responding and restarts it. These values are specified by the `wrapper.ping.interval` and the `wrapper.ping.timeout` parameters.

The JVM is terminated if the JVM does not shut down as expected. If the JVM does not shut down during the interval specified in the `wrapper.shutdown.timeout` parameter and the `wrapper.jvm_exit.timeout`, the JVM is terminated.

Optimizing Web Service Data Collector Performance

If you need to improve the performance of the Web Service Data Collector, you can adjust its queue settings.

To adjust Data Collector queue settings

1. In My webMethods, navigate the following path: **Applications > Administration > System-Wide > Environments > Define Environments**
The Define Environments page is displayed.
2. Click the name of the configured environment for which you want to adjust data maintenance settings.
The Edit Environment page is displayed.
3. Click the **Configure Servers** tab.
4. On the **CONFIGURATION** tree, find the Web Service Data Collector for which you want to adjust settings, and click **Data Collector Settings** under the name of the Data Collector. You can adjust the following Data Collector settings:

Attribute	Definition
<code>jmsResendQueueSize</code>	Relative storage size of the queue for re-sending information to the Web Service Data Collector. Default = 50000
<code>eventQueueSize</code>	Number of events that can be held in the Web Service Data Collector queue. Default = 50000
<code>dimensionQueueSize</code>	Relative size of each dimension held in the Web Service Data Collector queue. Default = 50000

Note: Only Web Service Data Collector queue settings can be adjusted from the **Configure Servers** tab. See *Configuring BAM* for information on optimizing Infrastructure Data Collector performance.

Connecting Process Engines to Analytic Engines

If you are using Optimize for Process and processes are enabled for analysis, you must connect the Process Engines to the Optimize Analytic Engines so that My webMethods can display the process metrics. Process Engines are represented in Integration Server as the WmPRT package, and you manage the settings for this package using the Integration Server Administration utility.

A JMS Server connects the Process Engines to the Analytic Engines. Typically, your JMS Server will be either **webMethods Universal Messaging** (default) or Broker. By default, the URL provided to the Process Engines for this JMS Server is `nsp://localhost:9000`. If your JMS Server is not **webMethods Universal Messaging**, or if this JMS Server is not on the same machine as a Process Engine, or the port it is using is not 9000 you must modify the URL. Also, if you are using Broker as a JMS Server, you must modify this URL.

To connect Process Engines to Analytic Engines

1. In Integration Server Administrator for an Integration Server that hosts a Process Engine, go to the **Packages > Management** page and click  for the WmPRT package.
2. Click **Settings** on the left hand navigation panel.
3. Click the **Edit Process Engine Settings** link at the top of the Process Engine Settings panel.
4. In the **JMS Server URL** box under the webMethods Optimize heading, change `localhost` to the correct host, and the port assignment to the correct port (if applicable), using the appropriate format for your JMS Server. The URL formats for **webMethods Universal Messaging** and Broker are as follows:
 - **webMethods Universal Messaging:** `nsp://localhost:9000`
 - **Broker:** `broker://localhost:6849/Broker #1`

Important: For a Broker JMS Server, do not change the `broker://` part of the URL; however, you should change `Broker #1` to the correct JMS Server name.
5. Click **Submit**.
6. Repeat these steps for every Integration Server that hosts a Process Engine.

Adjusting RAM for the Analytic Engine and Web Service Data Collectors

From time to time, you may want to adjust the settings for the amount of RAM that Optimize needs on your Analytic Engine and Web Service Data Collector host systems to accommodate the number of monitors that the Analytic Engine and Web Service Data Collectors are hosting. Contact Global Consulting Services for assistance.

Configuring Optimize to use an Analytic Engine

When initially started, BPM only and BPMS Optimize installations use a default Integration Server (BPM environment) as the logical server component of their default environments. BAM only and BAM/BPM mixed installations cannot use the default Analytic Engine (BAM environment) or custom defined Analytic Engines listed on the System Settings page until these components have been deployed using the My webMethods Define Environments page (**Applications > Administration > System-Wide > Environments > Define Environments**). Once you have deployed an Analytic Engine, you can use the My webMethods System Settings page to review/define the host name and port, and you can define the server as the default server for monitoring, if desired. Also, you can test the connection to the server from this page.

Note that you can use the default BAM only Analytic Engine without any modifications after it has been deployed. If you want to change the host and/or port for the default Engine, you can do so from the System Settings page. If you want to use a custom-defined Analytic Engine, you must first add it (using the **Add** button on the System Settings page) in order for it to appear in the list of available servers. After it is added, you can modify its configuration as desired.

Note: All Analytic Engines and My webMethods must use the same operating system locale configuration. Refer to your operating system documentation for information about setting or changing the locale configuration.

Note: You can do everything described in this section for an Integration Server as well as for an Analytic Engine, if applicable to your system configuration. Integration Servers do not need to be configured and deployed using the My webMethods Define Environments page. Also, be aware that the fields displayed on the System Settings page vary depending on the installation type that was selected. If you have a BAM-only installation, the Integration Server configuration fields are not displayed. If you have a BPM-only installation, the Analytic Engine configuration fields are not displayed.

To configure an Analytic Engine

1. From My webMethods, navigate the following path: **Applications > Administration > My webMethods > System Settings**.

The System Settings page is displayed, with all currently configured Analytic Engines and Integration Servers shown in the **Server Settings** panel.

2. If you want to add a new Analytic Engine to the list of displayed servers, click **Add** to add the Analytic Engine host to your system configuration. When adding a new Analytic Engine, you must enter a server name, a port, and an Analytic Engine host name.
3. If the default host name and port name for an existing Analytic Engine are not correct for your system configuration, edit them as appropriate.

4. If you want to use a displayed Analytic Engine to monitor information by default, click the radio button in the **Default** column beside the server that corresponds to the desired Analytic Engine.
5. Select the check box in the **Use SSL** column if the server uses an SSL connection.
6. Click **Save** to save your configuration information.
7. Click **Check Server Status** to check the status of all configured servers.

Configuring Optimize for Analytic Engine Clustering

Optimize supports clustering of Analytic Engines to provide degrees of high availability and, in applicable cases, to increase system data throughput. See the “Managing webMethods Environments” chapter in *Configuring BAM* for specific information about Analytic Engine clustering.

Configuring Optimize to Monitor Multiple Integration Servers and Analytic Engines

You can configure Optimize to monitor multiple Integration Servers and/or multiple Analytic Engine servers, that exist in separate environments, using the My webMethods System Settings page. Note that additional Analytic Engines must be configured and deployed using the My webMethods Define Environments page (**Applications > Administration > System-Wide > Environments > Define Environments**) and configured on the System Settings page before they can be used. Once any additional Integration Servers or Analytic Engine hosts are successfully configured, a **Server** drop-down menu appears at the top of most My webMethods pages to enable users to select from the servers available to them.

Note: Configuring one or more Integration Servers and/or Analytic Engines is required only for BPM, BAM, or mixed BPM and BAM (BPMS) environments. Specifying the Analytic Engine host and port on the System Settings page is required for BAM-only installations and Optimize for Infrastructure installations that use only the Infrastructure Data Collector for data collection.

To configure multiple Integration Servers and Analytic Engines

1. From My webMethods, navigate the following path: **Applications > Administration > My webMethods > System Settings**.

The System Settings page is displayed with all currently configured Analytic Engines and Integration Servers shown in the **Server Settings** panel.

2. Click **Add** to add a new Integration Server, Analytic Engine, or combined Integration Server and Analytic Engine host.
3. Enter a name for your new server in the **Name** column, and do one of the following:

- To add an Integration Server, enter the host name or network address and port number in the **Integration Server(Monitor) Host** and **Port** columns, and then select the check box in the **Use SSL** column if the server uses an SSL connection.
 - To add an Analytic Engine server, enter the host name or network address and port number in the **Analytic Engine Host** and **Port** columns, and then select the check box in the **Use SSL** column if the server uses an SSL connection.
 - To add a combined Integration Server/Analytic Engine host, enter the host name or network address and appropriate port in *both* the **Integration Server(Monitor) Host** and **Analytic Engine Host** columns.
4. Click **Save** to save your configuration information.
 5. Click **Check Server Status** to check the status of all configured servers.
 6. Click **Delete** to delete a server configuration row, or click **Add** to configure another server. The default server cannot be deleted.
 7. The radio button in the **Default** column defines the default server. To change the default server, select the radio button beside the server you want to make the default and click **Save**.

Configuring EDA/NERV Settings for Optimize

To support subscription to and publication of events, if your Optimize system operates in an EDA enabled environment using NERV, you must configure the NERV JMS Provider setting in the profile for the Platform Manager (SPM) that is running on the Analytic Engine machine. Though it supports event subscription and publication, this configuration is also important for systems that use a Universal Messaging server that is installed remotely from the Optimize installation.

By default, the NERV JMS Provider is defined as `nsp://<host_name> :9000`. To configure this setting, you must edit the `COMMON-SYSPROPS-com.softwareag.eda.nerv.properties` file using the Command Central interface. Detailed instructions for configuring the NERV JMS Provider can be found in the "Modifying Transport Layer Configuration" section of the *Implementing Event-Driven Architecture with Software AG Products* guide. When using these instructions, however, note that Optimize by design does not have a profile on the Common Platform, so when you configure the JMS Provider you must set it in the profile for the Platform Manager (SPM) that is running on the same machine as the Analytic Engine.

Note that if you are using the `WmOptimize` package in an EDA environment with NERV, you must configure the JMS Provider in the profile for the Integration Server that is running on the same machine as the Analytic Engine.

Guaranteed Event Storage in an Optimize Environment

This section explains how to implement a guaranteed event storage solution in a typical Optimize environment. This solution provides maximum integrity and minimizes

the chances of losing event data due to system or component down time or software/hardware failure.

Implementing Guaranteed Event Storage in Optimize

Guaranteed event storage functionality in Optimize uses a custom Monitor API (MAPI) configuration and batching techniques to deliver events and provide for enhanced notification of event delivery problems. When implemented appropriately, this functionality supports robust event delivery and, in the event of an event delivery problem, helps users correct problems before they cause significant data corruption. Also, it enables you to configure your system to persist cached event data to disk. Note that for most Optimize configurations, there is a trade-off between event delivery integrity and performance; there are several variables involved in this trade-off, and some experimentation may be necessary to achieve the desired balance for your system.

This feature includes two customer-configurable components that support guaranteed event storage.

- Setting a MAPI flag to support synchronous event delivery to Analytic Engines.
- Setting System Properties that determine whether system caches are configured for high event delivery integrity (persistence) or high performance.

Note: The Analytic Engine is designed to recover from webMethods Broker connectivity and database connectivity issues. There is no need to restart the Analytic Engine when a database and/or Broker connectivity issue arises. Once the connectivity issue is addressed, Analytic Engine will continue to process data as expected.

Configuring the MAPI Flag

By default, events in Optimize are published by the Monitor API (MAPI) and handled asynchronously. This method provides optimal performance, but it means that users may not receive immediate notification of event publication errors. Using the flag described herein, you can configure the Optimize MAPI service to publish events synchronously. If an event delivery problem occurs in this configuration, errors are generated immediately.

The `-Dsag.opt.mapi.sync.event.flag` system property setting in the Integration Server startup file enables synchronous event delivery in all the components that use MAPI client mechanism running in the same Java Virtual Machine (JVM). It is important to remember that this flag is not limited to a single component; it can be set for any client JVM that uses the MAPI jar file.

Note that, depending on your system configuration, there may be multiple MAPI instances present in a system or JVM. In most cases, you can simply set the MAPI flag in the startup script for Integration Server or Infrastructure Data Collector or the custom build client application, etc.

To configure the MAPI flag on an Optimize system.

1. Open the Integration Server startup file located in the Integration Server/bin folder under your installation directory. On a Windows system, the file is named server.bat, while on a Unix system, it is named server.sh.
2. Add the following line to the “run integration server” section of the startup file.

```
-Dsag.opt.mapi.sync.event.flag=y
```

Here’s an example of the flag added to a generic startup file:

```
SET JAVA_OPTS=-Dsun.lang.ClassLoader.allowArraySyntax=true
-Dwatt.server.prepend.classes=%PREPEND_SYSTEM_CLASSPATH%;%APPEND_SYSTEM_CLASSPATH%;%PREPENDCLASSES% -Dwatt.server.append.classes=%APPENDCLASSES%
-Dcom.webmethods.sc.auditing.Use80TableFormat=true
-Djavax.xml.xpath.XPathFactory:http://java.sun.com/jaxp/xpath/dom=com.sun.org.apache.xpath.internal.jaxp.XPathFactoryImpl
-Dtangosol.coherence.shutdownhook=false -Dsag.opt.mapi.sync.event.flag=y
%DEBUG_JAVA_OPTS% %9
```

3. Save the file and close it.
4. Repeat the preceding steps for any other applicable components in your system.

Disabling the MAPI Flag

If you want to disable the MAPI flag, you must remove the event flag line from the configuration file.

Note that, depending on your system configuration, there may be multiple MAPI instances and flags present in a system or JVM. For most typical systems, there will be a MAPI flag in the startup script for Integration Server or Infrastructure Data Collector or the custom build client application, etc.

To disable the MAPI flag on an Optimize system.

1. Open the Integration Server startup file located in the Integration Server/bin folder under your installation directory. On a Windows system, the file is named server.bat, while on a Unix system, it is named server.sh.
2. Locate and delete the following line from the “SET_JAVA_OPTS=” section of the startup file.

```
-Dsag.opt.mapi.sync.event.flag=y
```

3. Save the file and close it.
4. Repeat the preceding steps for any other applicable components in your system.

Setting System Cache Properties

This section explains how to set system cache properties to maximize event data integrity on an Optimize system. By default, the Optimize Analytic Engine caches event data in memory, which means that, in some cases, data can be lost if system components terminate or experience hardware failure. To provide enhanced event delivery integrity, this property enables you to configure an Optimize 9.0 Analytic Engine to cache event

data to disk so that event data is persisted even if a system component crashes or shuts down for some reason.

Note that disk caching can entail a significant performance penalty on some systems, especially those that process a large number of events; so making decision about caching options is often a matter of balancing performance vs. event integrity. When cache properties are configured for guaranteed event delivery, performance will be degraded to some degree, according to the specifics of your data and system configuration.

Basic cache configuration properties for the Analytic Engine are set using two parameters in the `optimize\analysis\conf\wrapper.conf` file. The first parameter specifies the desired level of event integrity. The second parameter specifies the local folder to use for persisting event data when it is cached to disk.

The two level settings available for the event caching parameter are as follows.

- Normal - This is the default setting, and it is equivalent to having disc caching disabled. For practical purposes, there is no reason to explicitly set event caching to this level.
- High - This setting saves event data to a specified disk location. However, event data is held in memory for some time, and if the Analytic terminates abnormally for some reason, that data could be lost. There is some performance degradation, though the exact amount depends on the specific system configuration and the amount of data being processed. The primary performance/event integrity trade-off occurs during Analytic Engine shutdown.

To enable disk caching on a system, you must add the appropriate lines to your `wrapper.conf` file to point to a valid folder on your system suitable for saving cache information, and you must specify the event guarantee level. The new caching information will take effect when you next start the system. If you implement disk caching, the persistence folder specified must have appropriate security configuration so that the user running the Analytic Engine has permission to write to the folder.

To set system cache properties:

1. Locate and open the `optimize\analysis\conf\wrapper.conf` file in a text editor.
2. Based on your system configuration, add the appropriate line to indicate the event caching persistence folder and the desired level of event guarantee. The following examples are sample event caching lines for a system that uses `C:\temp` (Windows) and `/var/log` (Linux) as the persistence folder, and High as the event guarantee level:

```
wrapper.java.additional.300= -Dsag.opt.persistence.dir="C:\temp"
wrapper.java.additional.301= -Dsag.opt.event.guarantee=High
```

3. Save the file and close it. Remember that you must restart your system for any changes to take effect.

Note: To minimize the potential for data loss, you must exercise caution if you are switching a system between event caching levels. Before switching the event caching setting, shut off the data feed and allow data to be persisted

to disk. Then, shutdown the Analytic Engine normally before you make a change to the event caching setting.

4. You must restart your system for changes to take effect.

Disabling Disk Caching

Use the following procedure to disable disk caching on an Optimize 9.0 system.

To disable disk caching

1. Locate and open the wrapper.conf file in a text editor.
2. Locate the caching configuration line and delete it. See the previous procedure for information about this line in various system configurations.
3. Save the file and close it.

Configuring the PPM Server Settings

The My webMethods PPM Server Settings page enables you to configure a connection to a Software AG Process Performance Manager (PPM) server so that you can access advanced process analytics information. This connection is configured via a URL that specifies the appropriate server and port number.

To configure the PPM Server Settings

1. From My webMethods, navigate the following path: **Applications > Administration > My webMethods > System Settings > PPM Settings**.
2. Type an appropriate URL value in the **PPM URL** field. The URL should be of the following format: `http://<fully_qualified_server_name>:<port_number>/ppm/html`. Note that the actual URL required to connect may vary slightly based on the specific server configuration.
3. Optionally, type an appropriate value in the **PPM Client** field. Note that in case no user-specific configuration files exist on the PPM server side, the default Performance Dashboard is displayed.
4. Click **Test** to verify the validity of the connection if desired.
5. Click **Save** to save the setting you entered when you are done.

Note: If Single Sign-On using SAML is configured for My webMethods Server and PPM, you will not be required to provide credentials when connecting to the PPM server through the My webMethods Server user interface. For more information about setting up single sign-on using SAML, see the ARIS user documentation.

4 Configuring Business Processes

■ Overview	100
■ Preparing to Monitor Processes	100
■ Identifying a Process to Monitor	101
■ Editing a Process	103
■ Instance Analytics Tab	105
■ Working with Stages and Milestones	120
■ Enabling and Disabling EDA Event Emission	123
■ About Synchronizing Stages and Events with Software AG Designer	125
■ Working with the Overview Configuration Page	125
■ Working with Configuration Components	129
■ Granting User Access to Business Processes	136

Overview

The material in this chapter applies only to Optimize for Process. Optimize for Process can analyze and display status information for the following two types of business processes:

- *webMethods-executed processes* are processes that are executed and tracked by webMethods tools. webMethods-executed processes are also known as *BPM-only (or BPM) processes*.
- *Externally executed processes* are processes that are modeled and monitored using webMethods-provided tools, but these processes are executed by an external application. Externally executed processes are also known as *BAM-only (or BAM) processes*.

Once you have identified the processes you want to monitor, Optimize displays information about these monitored processes in the form of corresponding business monitors and process instance data on the Monitoring pages. See the *webMethods Optimize User's Guide* for more information about the Monitoring pages.

This chapter includes the following information:

- How to set up and identify the business processes you want Optimize to monitor.
- How to customize the display of the Analytics Overview page.

Preparing to Monitor Processes

Before you can identify the processes you want to monitor, you must first determine whether the process is a webMethods-executed (BPM) process or an externally executed (BAM) process. Then you will set up the processes using the Process Development perspective in Software AG Designer, as described below.

Setting Up a webMethods-Executed Process

For each webMethods-executed (BPM) process you want to monitor, do the following:

1. Create document types as necessary for input to and output from process steps. Be sure to mark the document types as publishable. For more information, see the *webMethods Service Development Help*.
2. Use the Process Development perspective in Software AG Designer to define a process model for the process. For steps that require documents as input or output, add subscriptions to the document types you created earlier. For more information, see the *webMethods BPM Process Development Help*.

Note: Optimize does not track process steps within external groups and does not display these steps on the Analytics Overview page. For more information, see "[Working with the Overview Configuration Page](#)" on page 125.

3. Build and upload the process model, or upload it for analysis only. Uploading the process model puts in motion the following three actions:
 - Alerts the webMethods Process Engine to start tracking the process.
 - Copies the process model to the Process Audit database component.
 - Makes the process available to Optimize for analysis.

For more information about uploading a process model, see the *webMethods BPM Process Development Help*.

Setting Up an Externally Executed Process

For each externally executed (BAM) process you want to monitor, do the following:

1. Define a process model for the process using the Process Development perspective in Software AG Designer. For more information, see the *webMethods BPM Process Development Help*.

Note: Optimize does not track process steps within external groups and does not display these steps on the Analytics Overview page. For more information, see ["Working with the Overview Configuration Page" on page 125](#).

2. Build and upload the process model, or upload it for analysis only. This copies the process model to the Process Audit database component and makes the process available to Optimize for analysis. For more information about updating a process model for monitoring, see the *webMethods BPM Process Development Help*.
3. Use the Web Service Data Collector to collect business process data (for instructions, see ["Configuring and Using the Web Service Data Collector" on page 253](#)).

Identifying a Process to Monitor

You use the Business Processes page to identify processes to monitor.

To identify a process to monitor

1. In My webMethods: **Navigate > Applications > Administration > Business > Business Processes**.
2. On the **Keyword** tab, click **Search**.

A list of uploaded business processes appears in the Business Processes window. The Business Process window includes the following information:

Column	Description
Process Name	Displays the name of the process.

Column	Description
Model Version	For a webMethods-executed (BPM) process, this indicates the version of the process model assigned by Software AG Designer. For an externally executed (BAM) process, this will always display 1 .
Type	Indicates whether the process is a webMethods-executed (BPM) process (webMethods) or an externally executed (BAM) process (External). Note: The Business Processes page contains the Type column only when webMethods Monitor is installed.
Execution Enabled	Shows whether the process is enabled for execution. This column is empty for externally executed (BAM) processes. Note: The Business Processes page contains the Execution Enabled column only when webMethods Monitor is installed.
Analysis Enabled	Shows whether the process is made available to Optimize for monitoring and analysis. Note that externally executed (BAM) processes are always available for monitoring and analysis.
Used	Indicates whether the process has been executed.
Date Deployed	Indicates the date on which the process was updated for monitoring in Software AG Designer.
EDIT	Contains the Edit icon. Click this icon to go to the Edit Process page.

- Use the Business Processes page to do any of the following:

To...	Do this...
Edit the time-out value or other error conditions for a process instance or for a step in a process instance.	Click the Edit icon and continue to "Editing a Process" on page 103 .

To...	Do this...
Enable or disable the execution of a webMethods-executed (BPM) process.	Click the Edit icon and continue to "Editing a Process" on page 103 .
Enable or disable the analysis of a webMethods-executed (BPM) process by Optimize.	Click the Edit icon and continue to "Editing a Process" on page 103 .
Remove a process from the list and delete its metadata from the database.	Select the check box beside the process name and then click the Delete button. Note: You can delete only unused processes.
Export the list of processes to a *.csv file	Click the Export Table button. For more information about exporting table data, see the <i>Working with My webMethods</i> .
Search or filter processes in the list	Use the Search panel. For more information about using the Search panel, see the <i>Working with My webMethods</i> .

Editing a Process

To edit a process

1. In My My webMethods: **Navigate > Applications > Administration > Business > Business Processes**
2. If necessary, enter search criteria on the **Keyword** tab, then click **Search**.
In the **Business Process** window, Optimize displays a list of processes that were defined and updated for monitoring using Software AG Designer.
3. To edit a process, click its process name or click the **Edit** icon in the **Edit** column. The Edit Process page appears.

Business Processes > Edit Process

Process Information Administration Save Cancel

Process Details **Instance Analytics**

0 selected Add Standardized Error Type... Create Custom Error Type... Delete

NAME ^	FOR	CONDITION	LAST OCCURRENCE	STOP	EDIT
Step Out Of Sequence	All Process Instances	Out of sequence	9/8/2014 5:54:56 AM	<input type="radio"/>	
Subprocess Error	All Process Instances	Sub process error	n/a	<input type="radio"/>	
System Error	All Process Instances	System error	n/a	<input type="radio"/>	

Process Stages and EDA Events

Stages Delete Stage Add Stage

Stage Name ^	Description ^	Start Milestone ^	End Milestone ^	Condition	Stop Tracking On Breach
stage 1	>1min [1 -> b1]	1 (S75)	Start b1 (S104)	Complete >	1 minutes <input type="checkbox"/>
stage 10	<2min [10 -> b10]	10 (S133)	Start b10 (S135)	Complete <	2 minutes <input type="checkbox"/>
stage 11	>2min [11->b11]	11 (S144)	Start b11 (S136)	Start >	2 minutes <input type="checkbox"/>
stage 2	<30sec [2 -> b2]	2 (S15)	Start b2 (S105)	Complete <	30 seconds <input type="checkbox"/>
stage 3	<1min [3 ->]b3	3 (S6)	Start b3 (S108)	Start <	1 minutes <input type="checkbox"/>
stage 4	<1min 4(->]b4	4 (S9)	Complete b4 (S107)	Start <	1 minutes <input type="checkbox"/>
stage 5	>45sec [5->join]	5 (S29)	Start join (S12)	Complete >	45 seconds <input type="checkbox"/>
stage 6	>10ms whole process	Start of process	End of process	>	10 milliseconds <input type="checkbox"/>
stage 7	>30sec [7 -> join *never reached]	7 (S70)	Start join (S12)	Complete >	30 seconds <input type="checkbox"/>
stage 8	<30ms [8 -> join *never reached]	8 (S26)	Start join (S12)	Complete <	30 milliseconds <input type="checkbox"/>
stage 9	<3ms [9 -> b9] stop emitting stage even	9 (S80)	Start b9 (S112)	Complete <	3 milliseconds <input checked="" type="checkbox"/>

1 - 11 of 11

Process Diagram

4. In the **Process Details** tab on the Edit Process page, you can do the following tasks:
 - a. Enable or disable the execution of a webMethods-executed (BPM) process by selecting or clearing the **Execution Enabled** check box.

Note: If you disable a webMethods-executed (BPM) process, its current instances will continue to run but new trigger documents cannot start new instances of the process.

- b. Enable or disable the Optimize analysis of a webMethods-executed (BPM) process by selecting or clearing the **Analysis Enabled** check box.

The **Analysis Enabled** check box is available only if the **Execution Enabled** check box is checked.
5. In the **Instance Analytics** tab on the Edit Process page, you can do the following:
 - a. View a list of error types associated with the current process.
 - b. Modify an existing error type to specify a time in which the process instance, or a step within the process instance, must complete.
6. In the **Process Stages and EDA Events** window:

- The **Stages** tab displays all process stages defined for this process model. For more information, see ["Working with Stages and Milestones" on page 120](#). If the process model contains no stages, this window is collapsed to show only the title bar. It cannot be expanded.

To sort the stages table by the contents of a column, click on a column name to activate the sort controls for that column.

- The **Events** tab enables you to enable or disable event emission for predefined EDA events from Process Engine. For more information, see ["Enabling and Disabling EDA Event Emission" on page 123](#).
7. The **Process Diagram** window shows the steps and, for externally executed processes, the events associated with the process. Note that this window interacts with the Stages tab. For more information, see ["Working with Stages and Milestones" on page 120](#).

The following right-click options are available in the **Process Diagram** window.

- **Fit to screen** - Resizes and re-centers the diagram based on the screen size.
 - **Hide transition labels** - Hides description and expression text on all transition lines in the diagram.
 - **Show transition descriptions** - Shows all applicable description text created in Designer for all transitions on the diagram.
 - **Show transition expressions** - Shows applicable transition expressions created in Designer for all transitions on the diagram.
8. Click **Save** to save your changes, or **Cancel** to return to the Business Processes page without saving. (**Save** and **Cancel** are replaced with **Close** for externally executed processes.)

Instance Analytics Tab

The **Instance Analytics** tab on the Edit Process page lists the following types of errors:

Error Type	Description
<i>Built-in System Errors</i>	Built-in system errors are error types that are automatically assigned to all processes by Process Tracker. For built-in system-error types, users can only configure whether the specific error condition stops (cancels) a process instance. The following three built-in system-error types are automatically assigned to all processes: Step Out Of Sequence, Subprocess Error, and System Error.

Error Type	Description
	<ul style="list-style-type: none"> ■ The built-in system-error types Step Out Of Sequence and System Error are displayed by default for all processes. ■ The built-in system-error type System Error indicates an exception detected in the Analytic Engine that affects the status (also known as the state) of a process instance or a step instance. System Error types are for troubleshooting or explaining an incorrect process status or step status. ■ The built-in system-error type Subprocess Error will appear only for processes that contain referenced subprocesses.

Standardized Errors

Standardized error types are user-defined error types tracked by Optimize. Standardized error types relate to conditions triggered by the Analytic Engine against a process instance, a step instance, or a step transition instance. Standardized error types use built-in error conditions that can be configured on the Add Standardized Error Type page.

Custom Errors

Custom error types are user-defined error types that are activated externally and tracked by Optimize. Custom error types can be defined by the user on the Create Custom Error Type page. Alternatively, custom error types can be created automatically by Optimize upon detection of the first received custom error, via the pushExecutionError (Execution Error) operation (for more information, see [Sending Process Data in "Configuring and Using the Web Service Data Collector" on page 253](#)).

The **Instance Analytics** tab on the Edit Process page lists the following information for error types that have been defined for the current process:

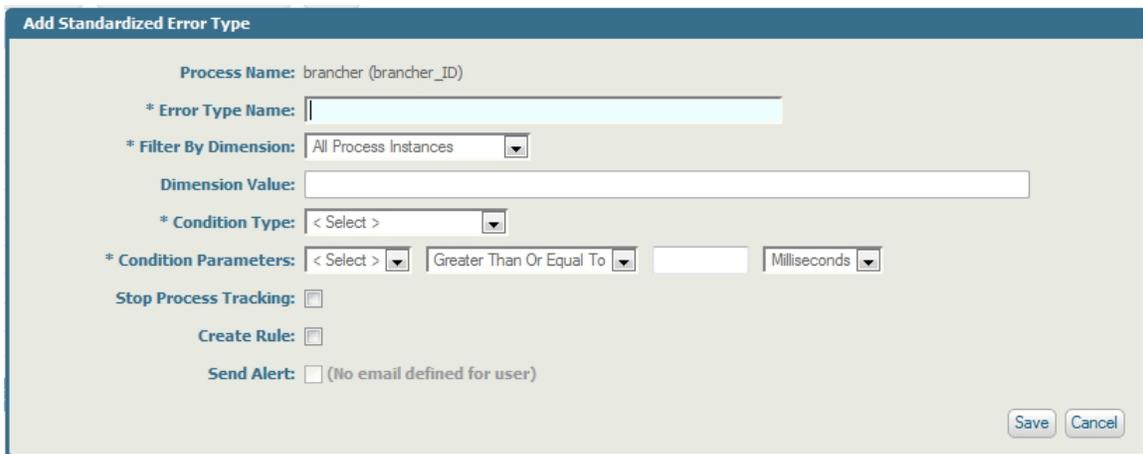
Column	Description
Name	Name of the error type.
For	Scope of the error condition. All Process Instances indicates that this error type applies to all instances of the process. If the error type applies to only one dimension of the process, that dimension is displayed in the following format:

Column	Description
	<i>dimension name (event mapname): dimension value</i>
Condition	The condition that, when true, causes an error of this type to display in the Process Errors panel on the Process Instance Detail page.
Last Occurrence	Timestamp of the last occurrence of an error of this type.
Stop	Whether tracking of this process instance will stop when the error condition occurs.  = The occurrence of this error condition stops the tracking of this process instance.  = Tracking of this process instance will continue even if this error condition occurs.
Edit	Click  to edit the definition of this error type. The three built-in system-error types (System Error, Step Out Of Sequence, and Subprocess Error) can only be edited to specify whether the occurrence of the error condition stops (cancels) the tracking of the process instance.

Standardized Error Types

Standardized error types are configurable by clicking the **Add Standardized Error Type** button in the **Instance Analytics** panel of the Edit Process page.

The Add Standardized Error Type dialog is displayed.



Add Standardized Error Type

Process Name: brancher (brancher_ID)

* Error Type Name:

* Filter By Dimension: All Process Instances

Dimension Value:

* Condition Type: < Select >

* Condition Parameters: < Select > Greater Than Or Equal To Milliseconds

Stop Process Tracking:

Create Rule:

Send Alert: (No email defined for user)

Standardized error types can be configured for the following five types of error conditions:

- Cycle Time
- Step Relative Wait Time
- Step Absolute Wait Time
- Iterations
- Deadline

Cycle-Time Standardized Error Types

Cycle-time error types are intended for configuring errors related to the time that it takes a process instance or a step in a process instance (also known as a step instance) to cycle through once, from start to finish.

To add a cycle-time standardized error type

1. Click the **Add Standardized Error Type** button in the **Instance Analytics** panel of the Edit Process page.

To configure a new cycle-time error type, provide the information described in the following table.

Field	Description
Error Type Name	<p>Enter a unique name for the error type.</p> <p>It can be helpful to create a name that indicates the parameters of the error type. Although this field will accept up to 150 alphanumeric characters, the name will be truncated at 50 characters on the Instance Analytics panel of the Edit Process page.</p>
Filter By Dimension	<p>Choose All Process Instances to apply the error type to all instances of the process. Alternatively, you can select a dimension from the drop-down list of all of the dimensions for all of the business event maps defined for this process. Each dimension listed will identify the event map with which that dimension is associated.</p> <p>Select a dimension from the drop-down list to apply the error type only to a specific instance of the selected dimension. If you select a dimension, enter the dimension instance in the text box to the right.</p>
Condition Type	Select the Cycle Time error type.

Field	Description
Condition Parameters	Use the drop-down menus to create an error condition consisting of: <i><operand></i> + <i><operator></i> + <i><value></i>
<i><operand></i>	Select an operand for the error condition from the first drop-down menu. Cycle-time operands consist of the name of the process instance, or the name of a step in the process instance, plus Cycle Time .
<i><operator></i>	<p>Select an operator for the error condition from the second drop-down menu.</p> <p>Choose from the following operators:</p> <ul style="list-style-type: none"> ■ Greater Than Or Equal To ■ Above Normal ■ One standard deviation above normal ■ Well Above Normal ■ Two standard deviations above normal
<i><value></i>	<p>This text box is available only if you selected the Greater Than Or Equal To operator. In the text box, enter a whole number against which to evaluate the error condition, and then select from the last drop-down menu a unit of measure for that whole-number value.</p> <p>Choose from the following units of measure:</p> <ul style="list-style-type: none"> ■ Milliseconds ■ Seconds ■ Minutes ■ Hours ■ Days ■ Weeks

Field	Description
Stop Process Tracking	Select this check box if you want Optimize to stop tracking this process instance if the error type occurs.
Create Rule	<p>This check box is available only if the current user has Access Privileges for the Rule List page (see <i>Administering My webMethods Server</i>). Select this check box if you want to automatically create a rule for this error type. The rule name will be the same as the error-type name, and the rule type will be Event.</p> <p>Errors appear only on the Process Instance Detail page. If you create an automatic rule for the error type, that rule will display on both the Problems page and the KPI Instance Detail page.</p> <p>If the error type is subsequently deleted, the associated rule remains unless it is manually deleted, as well.</p>
Send Alert	This check box is available only when Create Rule is selected and a valid e-mail address has been configured for the current user. Select this option if you want an e-mail alert sent to you when the rule you created is violated.

- Click **Save** to save the error type, or click **Cancel** to return to the Edit Process page without saving.

Wait-Time Standardized Error Types

Wait-time error types are intended for configuring errors related to the time that elapses leading to the start of a specified step in a process instance (the destination step). Wait-time error types are always applied to the destination step. The two available wait-time error types are Step Relative Wait Time and Step Absolute Wait Time.

A Step Relative Wait Time error type relates to the time that elapses between the completion of one step in a process instance (the source step) and the start of a subsequent, specified step in that same process instance (the destination step). A Step Absolute Wait Time error type relates to the time that elapses between the beginning of a process instance and the start of a specified step in that process instance.

To add a wait-time standardized error type

- Click the **Add Standardized Error Type** button in the **Instance Analytics** panel of the Edit Process page.

To configure a new wait-time error type, provide the information described in the following table.

Field	Description
Error Type Name	<p>Enter a unique name for the error type.</p> <p>It can be helpful to create a name that indicates the parameters of the error type. Although this field will accept up to 150 alphanumeric characters, the name will be truncated at 50 characters on the Instance Analytics panel of the Edit Process page.</p>
Filter By Dimension	<p>Choose All Process Instances to apply the error type to all instances of the process. Alternatively, you can choose to apply the error type to a specific dimension attribute, such as Customer. If you choose to apply the error type to a specific dimension attribute, you must also specify a value for that dimension attribute. In the example of the dimension attribute Customer, a value for that dimension attribute could be Software AG.</p> <p>To apply the error type to a specific dimension attribute, choose from the drop-down list of all of the available dimensions for all of the business event maps defined for this process instance. Each dimension listed will identify the event map with which it is associated.</p> <p>Once you've selected a dimension attribute from the drop-down list, enter a dimension value in the text box beside the drop-down list. The text box will default with the text of the selected dimension, up to a maximum of 80 alphanumeric characters.</p>
Condition Type	<p>Select either the Step Relative Wait Time or the Step Absolute Wait Time error type.</p>
Condition Parameters	<p>Use the drop-down menus to create an error condition consisting of: <i><operand></i> + <i><operator></i> + <i><value></i></p> <p><i><operand></i> Select an operand for the error condition from the first drop-down menu. Wait-time operands consist of the name of a step in the process instance plus Wait Time.</p> <p>Only steps that have an inbound transition (i.e., steps that can be entered from another step) appear in this drop-down menu.</p>

Field	Description
	<p data-bbox="570 327 704 357"><operator></p> <p data-bbox="776 327 1305 390">Select an operator for the error condition from the second drop-down menu.</p> <p data-bbox="776 415 1260 445">Choose from the following operators:</p> <ul data-bbox="776 470 1268 709" style="list-style-type: none"> <li data-bbox="776 470 1117 499">■ Greater Than Or Equal To <li data-bbox="776 520 992 550">■ Above Normal <li data-bbox="776 571 1256 600">■ One standard deviation above normal <li data-bbox="776 621 1045 651">■ Well Above Normal <li data-bbox="776 672 1268 701">■ Two standard deviations above normal
	<p data-bbox="570 751 672 781"><value></p> <p data-bbox="776 751 1305 991">This text box is available only if you selected the Greater Than Or Equal To operator. In the text box, enter a whole number against which to evaluate the error condition, and then select from the last drop-down menu a unit of measure for that whole-number value.</p> <p data-bbox="776 1016 1227 1079">Choose from the following units of measure:</p> <ul data-bbox="776 1104 971 1398" style="list-style-type: none"> <li data-bbox="776 1104 971 1134">■ Milliseconds <li data-bbox="776 1155 927 1184">■ Seconds <li data-bbox="776 1205 915 1234">■ Minutes <li data-bbox="776 1255 894 1285">■ Hours <li data-bbox="776 1306 883 1335">■ Days <li data-bbox="776 1356 899 1386">■ Weeks
Stop Process Tracking	<p data-bbox="570 1440 1305 1512">Select this check box if you want Optimize to stop tracking this process instance if the error type occurs.</p>
Create Rule	<p data-bbox="570 1554 1305 1759">This check box is available only if the current user has Access Privileges for the Rule List page (see <i>Administering My webMethods Server</i>). Select this check box if you want to automatically create a rule for this error type. The rule name will be the same as the error-type name, and the rule type will be Event.</p> <p data-bbox="570 1780 1305 1845">Errors appear only on the Process Instance Detail page. If you create an automatic rule for the error type, that</p>

Field	Description
	rule will display on both the Problems page and the KPI Instance Detail page. If the error type is subsequently deleted, the associated rule remains unless it is manually deleted, as well.
Send Alert	This check box is available only when Create Rule is selected and a valid e-mail address has been configured for the current user. Select this option if you want an e-mail alert sent to you when the rule you created is violated.

- Click **Save** to save the error type, or click **Cancel** to return to the Edit Process page without saving.

Iterations Standardized Error Types

Iterations error types are intended for configuring errors related to how many times a step in a process instance is repeated.

To add an iterations standardized error type

- Click the **Add Standardized Error Type** button in the **Instance Analytics** panel of the Edit Process page.

To configure a new iterations error type, provide the information described in the following table.

Field	Description
Error Type Name	Enter a unique name for the error type. It can be helpful to create a name that indicates the parameters of the error type. Although this field will accept up to 150 alphanumeric characters, the name will be truncated at 50 characters on the Instance Analytics panel of the Edit Process page.
Filter By Dimension	Choose All Process Instances to apply the error type to all instances of the process. Alternatively, you can choose to apply the error type to a specific dimension attribute, such as Customer. If you choose to apply the error type to a specific dimension attribute, you also will need to specify a value for that dimension attribute. In the example of the dimension attribute Customer, a value for that dimension attribute could be Software AG.

Field	Description
	<p>To apply the error type to a specific dimension attribute, choose from the drop-down list of all of the available dimensions for all of the business event maps defined for this process instance. Each dimension listed will identify the event map with which it is associated.</p> <p>Once you've selected a dimension attribute from the drop-down list, enter a dimension value in the text box beside the drop-down list. The text box will default with the text of the selected dimension, up to a maximum of 80 alphanumeric characters.</p>
Condition Type	Select the Iterations error type.
Condition Parameters	Use the drop-down menus to create an error condition consisting of: <i><operand></i> + <i><operator></i> + <i><value></i>
	<p><i><operand></i> Select an operand for the error condition from the first drop-down menu. Iteration operands consist of the name of a step in the process instance plus Iterations.</p>
	<p><i><operator></i> Select an operator for the error condition from the second drop-down menu.</p> <p>For iteration error types, only the following operator is available:</p> <ul style="list-style-type: none"> ■ Greater Than
	<p><i><value></i> Enter in the text box a whole number against which to evaluate the error condition.</p>
Stop Process Tracking	Select this check box if you want Optimize to stop tracking this process instance, or to stop tracking a step in this process instance, if the error type occurs.
Create Rule	This check box is available only if the current user has Access Privileges for the Rule List page (see the <i>Administering My webMethods Server</i>). Select this check box if you want to automatically create a rule for this

Field	Description
	<p>error type. The rule name will be the same as the error-type name, and the rule type will be Event.</p> <p>Errors appear only on the Process Instance Detail page. If you create an automatic rule for the error type, that rule will display on both the Problems page and the KPI Instance Detail page.</p> <p>If the error type is subsequently deleted, the associated rule remains unless it is manually deleted, as well.</p>
Send Alert	<p>This check box is available only when Create Rule is selected and a valid e-mail address has been configured for the current user. Select this option if you want an e-mail alert sent to you when the rule you created is violated.</p>

- Click **Save** to save the error type, or click **Cancel** to return to the Edit Process page without saving.

Deadline Standardized Error Types

Deadline error types are intended for configuring errors related to a fixed completion date/time for a process instance. Deadline error types allow you to designate a field (or fields) with a process event map as a date, including a specific time. You can then select any such date field in the process of configuring a deadline error type. Deadlines apply only to process cycle time and not to individual step processing times.

Deadline error types allow you to leverage Date type event-map attributes as deadlines, based on the value of the event-map attribute when received as part of business data for a process instance. For example, if your process is associated with an event map that has a `shippingDate` date attribute, you can specify a deadline timeout against `shippingDate`. When a `shippingDate` value is sent as part of business data for a process instance, that process instance must complete before the specified value of `shippingDate` plus or minus any additional offset time specified with the deadline error type.

To add a deadline standardized error type

- Click the **Add Standardized Error Type** button in the **Instance Analytics** panel of the Edit Process page.

To configure a new deadline error type, provide the following information:

Field	Description
Error Type Name	Enter a unique name for the error type.

Field	Description
	<p>It can be helpful to create a name that indicates the parameters of the error type. Although this field will accept up to 150 alphanumeric characters, the name will be truncated at 50 characters on the Instance Analytics panel of the Edit Process page.</p>
Filter By Dimension	<p>Choose All Process Instances to apply the error type to all instances of the process. Alternatively, you can choose to apply the error type to a specific dimension attribute, such as Customer. If you choose to apply the error type to a specific dimension attribute, you must also specify a value for that dimension attribute. In the example of the dimension attribute Customer, a value for that dimension attribute could be Software AG.</p> <p>To apply the error type to a specific dimension attribute, choose from the drop-down list of all of the available dimensions for all of the business event maps defined for this process instance. Each dimension listed will identify the event map with which it is associated.</p> <p>Once you've selected a dimension attribute from the drop-down list, enter a dimension value in the text box beside the drop-down list. The text box will default with the text of the selected dimension, up to a maximum of 80 alphanumeric characters.</p>
Condition Type	<p>Select the Deadline error type.</p> <p>The Deadline error type is available only if at least one process-associated event map has at least one Date attribute mapped.</p>
Condition Parameters	<p>Use the drop-down menus to create an error condition consisting of: <i><operand></i> + <i><operator></i> + <i><value></i></p> <p><i><operand></i> Select an operand for the error condition from the first drop-down menu. Deadline operands consist of "Dateattribute name" (event map name).</p> <p><i><operator></i> Select an operator for the error condition from the second drop-down menu.</p>

Field	Description
	<p>Choose from the following operators:</p> <ul style="list-style-type: none"> ■ Greater than or equal to ■ Use this operator to specify a “grace period” after the deadline occurs before issuing the error. ■ Less than or equal to ■ Use this operator to issue an error as an early warning before the deadline occurs. <p><value></p> <p>In the text box, enter a whole number against which to evaluate the error condition, and then select from the last drop-down menu a unit of measure for that whole-number value.</p> <p>Choose from the following units of measure:</p> <ul style="list-style-type: none"> ■ Milliseconds ■ Seconds ■ Minutes ■ Hours ■ Days ■ Weeks
Stop Process Tracking	Select this check box if you want Optimize to stop tracking this process instance, or to stop tracking a step in this process instance, if the error type occurs.
Create Rule	<p>This check box is available only if the current user has Access Privileges for the Rule List page (see <i>Administering My webMethods Server</i>). Select this check box if you want to automatically create a rule for this error type. The rule name will be the same as the error-type name, and the rule type will be Event.</p> <p>Errors appear only on the Process Instance Detail page. If you create an automatic rule for the error type, that rule will display on both the Problems page and the KPI Instance Detail page.</p>

Field	Description
	If the error type is subsequently deleted, the associated rule remains unless it is manually deleted, as well.
Send Alert	This check box is available only when Create Rule is selected and a valid e-mail address has been configured for the current user. Select this option if you want an e-mail alert sent to you when the rule you created is violated.

- Click **Save** to save the error type, or click **Cancel** to return to the Edit Process page without saving.

Editing an Existing Standardized Error Type

To edit a previously configured standardized error type

- On the Edit Process page, click  Edit beside the error type you want to edit.

The Edit Error Type page is displayed.

Note: For built-in system-error types (Step Out Of Sequence, Subprocess Error, and System Error), you can only enable or disable **Stop Process Tracking**. Otherwise, built-in system-error types cannot be edited.

- Make any changes to the error-type definition.
- Click **Save** to save your changes, or click **Cancel** to return to the Edit Process page without saving.

Deleting a Previously Defined Standardized Error Type

To delete a previously defined standardized error type

- On the Edit Process page, select the check box beside the name of the error type you want to delete.

Check boxes are available only for standardized or custom error types that have no associated error instances. Built-in system-error types cannot be deleted. Once you select an available check box beside the name of an error type, the **Delete** button is activated.

- Click **Delete**.

The error type is deleted, and if a rule was generated automatically when the error type was created, that rule is also deleted. However, any rules that were manually created for this error type also must be deleted manually.

Note: Built-in system-error types (Step Out Of Sequence, Subprocess Error, and System Error) cannot be deleted. In addition, any error type that has an existing instance cannot be deleted.

Custom Error Types

To add a custom error type to the list of error types displayed in the **Instance Analytics** panel on the Edit Process page, you must define the custom error-type name. Custom errors are activated externally and then tracked by Optimize.

To add a custom error type

1. On the Edit Process page, click the **Create Custom Error Type** button in the **Instance Analytics** panel.

The Create Custom Error Type dialog is displayed.

2. Enter a unique name for the error type in the **Error Type Name** field. The name must match the name originally given to this error type. Custom errors are sent into Optimize by a custom data collector. The custom data collector gives each custom error a name, and the name entered on the Create Custom Error Type page must match that name from the custom data collector.

Although this field will accept up to 150 alphanumeric characters, the name is truncated at 50 characters on the **Instance Analytics** panel of the Edit Process page.

3. Select **Stop Process Tracking** if you want Optimize to stop tracking this process instance, or to stop tracking a step in this process instance, if the error type occurs.
4. Select **Create Rule** if you want to automatically create a rule associated to this error type. The rule name will be the same as the error-type name, and the rule type will be Event.

This check box is available only if the current user has Access Privileges for the Rule List page (see *Administering My webMethods Server*).

Note: Errors appear only on the Process Instance Detail page. If you create an automatic rule for the error type, that rule is displayed on both the Problems page and the KPI Instance Detail page. If the error type is subsequently deleted, the associated rule remains unless it is manually deleted, as well.

5. If you selected **Create Rule**, you can select the **Send Alert** check box if you want an e-mail alert sent to you when the rule you created is violated.

This check box is available only if a valid e-mail address has been configured for the current user.

Working with Stages and Milestones

Although process stages are typically defined at design time in Designer, you can modify and delete those stages, or create new stages in the process model. For additional information, see ["Editing a Process" on page 103](#).

After you save your changes on the Edit Process page, you can synchronize these changes to your process model in Designer, as described in ["About Synchronizing Stages and Events with Software AG Designer" on page 125](#).

Note: You must have a BPM or a BPM and BAM server environment selected in the Server list at the top of the Process Instances page before you can add stages in the Edit Process page.

For detailed information about stages and milestones, see the *webMethods BPM Process Development Help*.

Adding a Stage

Note: If you leave the **Stages** tab while adding a stage and before you have clicked **Save**, your changes will be discarded.

Note: Before you make modifications to process model stage settings, be aware of the interaction of these settings between Designer and the Edit Process page. For more information, see ["About Synchronizing Stages and Events with Software AG Designer" on page 125](#).

To add a stage

1. On the Business Processes page, locate the process model that you want to work with and click  **Edit**.
2. In the Process Stages and EDA Events window, click the **Stages** tab.
3. Click **Add Stage**. A new row appears in the stage list, populated with default information.
4. Configure the following fields to define the stage:

Note: Any data entry validation errors are displayed within the stage row.

Column	Description
Name	<p>Type a name for the stage.</p> <p>Note: The Name is not editable after you click Save. If you want to rename a stage, you must delete it and then recreate it with the new name.</p> <p>Note: There is an 80-character limit for the stage name when double-byte characters are used in an IBM DB2 database. If you are not using DB2, or if your characters are single byte, then the stage name is limited to 255 characters.</p>
Description	Optional. Type a description of the stage.
Start Milestone	<p>Click the list and select a milestone. Optionally, you can type characters in the text box to filter the list. The Start Milestone and End Milestone selections must be different.</p> <p>Click the list to the right of the milestone selection, and click Start or Complete to specify the start or the completion of the selected milestone.</p>
End Milestone	<p>Click the list and select a milestone. Optionally, you can type characters in the text box to filter the list. The Start Milestone and End Milestone selections must be different.</p> <p>Click the list to the right of the milestone selection, and click Start or Complete to specify the start or the completion of the selected milestone.</p>
Condition	<ul style="list-style-type: none"> ■ Select < (less than) or > (greater than). Default is <. ■ Enter a positive whole number. The maximum supported values are as follows: <ul style="list-style-type: none"> ■ 2,777,777 hours ■ 166,666,666 minutes ■ 9,999,999,999 seconds ■ 9,999,999,999,999 milliseconds ■ Default is 1. ■ Select hours, minutes, seconds, or milliseconds. Default is hours.

Column	Description
	<p>The result is a condition. If the condition specifies <, then the stage is breached when the cycle time exceeds the specified time period. If the condition specifies >, then the stage is breached when the cycle time is less than the specified time period. For example:</p> <p>< 1 hours means that the stage must complete in less than 1 hour or a <code>ProcessStageBreached</code> event will be emitted.</p>
Stop Tracking On Breach	<p>Stops stage processing for all remaining stages in the process instance when a stage breach occurs in this stage, and only one stage breached EDA event is emitted. Remaining stages are not tracked and will be shown as Incomplete. The check box is cleared by default.</p>

5. Click **Save**.

For information about synchronizing the new stage with the process model in Designer, see ["About Synchronizing Stages and Events with Software AG Designer "](#) on page 125.

Modifying a Stage

- Note:** If you leave the **Stages** tab while modifying a stage and before you have clicked **Save**, your changes will be discarded.
- Note:** Before you make modifications to process model stage settings, be aware of the interaction of these settings between Designer the Edit Process page. For more information, see ["About Synchronizing Stages and Events with Software AG Designer "](#) on page 125.

You cannot modify a stage name. If you want to rename a stage, you must delete it and then recreate it with the new name.

Otherwise, all other stage and milestone information can be modified as described in ["Adding a Stage" on page 120](#).

Deleting a Stage

- Note:** If you leave the **Stages** tab after deleting a stage and before you have clicked **Save**, the deletion will be discarded.
- Note:** Before you make modifications to process model stage settings, be aware of the interaction of these settings between Designer and the Edit Process page. For more information, see ["About Synchronizing Stages and Events with Software AG Designer "](#) on page 125.

To delete a stage

1. On the Business Processes page, locate the process model that you want to work with and click  **Edit**.
2. In the Process Stages and EDA Events window, click the **Stages** tab.
3. Click the option button next to stage name for the stage you want to delete. To clear your selection, click the option button again
4. Click **Delete**.
5. Click **Save**.

Viewing Stages in the Process Diagram

You can display a stage's start milestone  and stage end milestone  in the Process Diagram window. Only one stage can be displayed in the Process Diagram at any time.

To view a stage in the process diagram, select any row in the stage list as follows:

- Click the option button next to stage name to select a stage row. The  icon becomes active and the milestone pins are rendered in diagram.
- To disable the pin display in the diagram for a stage row and keep the row selected, click the  icon.
- To disable the pin display in the diagram for a stage row and clear the row selection, click the option button .
- For a selected stage row with pin display disabled, click to display the pins again.
- Click the option button of another stage row to switch stage selection. The milestone pins on diagram are updated for the newly selected stage

Enabling and Disabling EDA Event Emission

You can enable and disable the emission of predefined EDA events for individual process models on an event-by-event basis. When event type emission is disabled, no predefined events are emitted by the model for the disabled event type. EDA event emission for all predefined event types is disabled by default.

To enable event emission, you must manually enable the predefined EDA events you want to emit for each individual process model. The event emission enablement settings are maintained with the process through the build and upload procedure. They are part of the process asset description, and are sent with the process when deployed with webMethods Deployer.

Note: Disabling these EDA events has no effect on custom event types applied to the process model or steps within it.

In addition to enabling EDA event emission as described below, you can also enable events in Software AG Designer. For more information about enabling and disabling

EDA event emission in Designer, and about EDA events in general, see *webMethods BPM Process Development Help*.

Note: Before you modify process model EDA event settings, be aware of the interaction of these settings between Designer and the Edit Process page. For more information, see "[About Synchronizing Stages and Events with Software AG Designer](#)" on page 125.

To enable or disable EDA event emission

1. On the Business Processes page, locate the process model you want to work with and click  **Edit**.
2. In the Process Stages and EDA Events window, click the **Events** tab. The following predefined EDA event types appear in the **Available EDA Events** list:
 - **Process Instance** controls whether to emit an event when the status of a process instance changes, for example from started to completed.
 - **Process Instance Log Message** controls whether to emit an event when a message is logged from a process model step, for example, warnings, messages, and errors.
 - **Process Instance Error** controls whether to emit an event when an error occurs in a process instance.
 - **Process Instance Log Custom ID** controls whether to emit an event when the service `pub.prt.log:logCustomId` is invoked in a process model step to associate a custom ID with the process model instance.
 - **Step Instance** controls whether to emit an event when the status of a step changes, for example, from started to waiting.
 - **Step Loop Instance** controls whether to emit an event when a step loop starts or completes.
 - **Step Instance Transition** controls whether to emit an event when one step transitions to another step.
 - **Step Instance Error** controls whether to emit an event when an error occurs during a process step.

Note: For detailed information about each of these EDA event types, see the topic "EDA Event Types" in *webMethods BPM Process Development Help*.

3. Select the EDA event you want to enable or disable for emission.
4. Use the available buttons to move EDA events between the **Available EDA Events** list and the **Selected EDA Events** list:
 - Click  or  to move a single event selection or a multiple, non-contiguous selection made by selecting events with the CTRL key pressed.
 - Click  or  to move all EDA events in a list to the other list.
5. Click **Save**.

About Synchronizing Stages and Events with Software AG Designer

You can create, modify, and delete stages, and enable/disable EDA events, in two locations:

- On the Edit Process page in My webMethods, as described in the accompanying topics.
- In Software AG Designer, as part of the process model development functionality. For more information, see the *webMethods BPM Process Development Help*.

In both cases, any changes to the stage or event definitions in a process model can be saved to the Process Audit database. The saved changes overwrite whatever previous stage information was present in the database. As a best practice, you should ensure that your process model stage and event settings are always synchronized between the two locations. For the Edit Process page, the following conditions apply:

- When you open the Edit Process page to view the process model settings in the Process Stages and EDA Events window, the stage and event settings saved in the Process Audit database are retrieved and displayed.
- When you save the process model stage and event settings on Edit Process page, the settings are written to the Process Audit database, overwriting whatever stage and event settings are stored there.
- Designer writes stage and event settings to the database when the process model is built and uploaded, overwriting whatever stage and event settings are stored there.

Therefore, be aware that it is possible for someone working in Designer to modify the database stage or event settings after you have opened Edit Process page. In this case, when you click **Save** on the Edit Process page, the stage and event settings in the Edit Process page will overwrite the settings written to the database by the person working in Designer.

Similarly, if you save stage and event settings to the database from the Edit Process page, it is possible for a Designer user to overwrite those settings at any time.

A **Synchronize** button is available in Designer for both stages and events, enabling the Designer user to apply the current database stage settings to the process model in Designer. You are advised to establish procedures to ensure that stage settings event settings are managed without conflict between Designer users and Edit Process page users.

To help ensure that you are working with the latest settings, you are advised to either refresh the Edit Process page or re-open the Edit Process page immediately before modifying and saving stage and event settings.

Working with the Overview Configuration Page

The Overview Configuration page in the Administration section of My webMethods enables you to set up and manage multiple customized data overviews, referred to as

configurations. These configurations can include both system and business process data, and they can be enabled for viewing on the Analytics Overview page. See the *webMethods Optimize User's Guide* for more information about the Analytics Overview page.

There are two default configurations available on the Overview Configuration page for all systems. These are explained below.

- Analytics System View: This configuration contains all system components.
- Analytics Business View: This configuration contains all business components.

These default configurations are only editable by administrators, though normal users can rename them.

If your system had a custom Process Overview configuration saved in Optimize for Process 7.1, that Process Overview configuration automatically becomes the first custom overview configuration available to all users in Optimize for Process 8.2.

Important: The Analytics Overview page is not automatically updated when changes are made to process models or when KPIs are added and removed. These changes must be maintained manually by the system administrator.

Enabling and Disabling Configurations for Viewing on the Analytics Overview Page

The following procedure enables you to control which configurations are available for viewing on the Analytics Overview page (**Navigate > Applications > Monitoring > Analytics Overview**).

To enable or disable a configuration for display on the Analytics Overview page

1. In My webMethods select: **Navigate > Applications > Administration > Analytics > Overview Configuration**

Optimize displays the Overview Configuration page. The first time this page is displayed on a freshly installed system with no data available, it contains only the following message: **No Configurations Found**.

2. To enable or disable a configuration for viewing, click the icon in the configuration's **Enabled** column.
 - A Green icon () in Enabled column indicates that the configuration is available for viewing on the **Navigate > Applications > Monitoring > Analytics Overview** page.
 - A Red icon () in the Enabled column indicates that the configuration is not available for viewing on the **Navigate > Applications > Monitoring > Analytics Overview** page.

Creating, Editing, Copying, and Deleting Configurations

The Add/Edit Overview Configuration page which is accessible from the Overview Configuration page enables you to add, edit, or copy a configuration. Also, you can delete configurations from the Overview Configuration page.

To create or edit a configuration on the Overview Configuration page

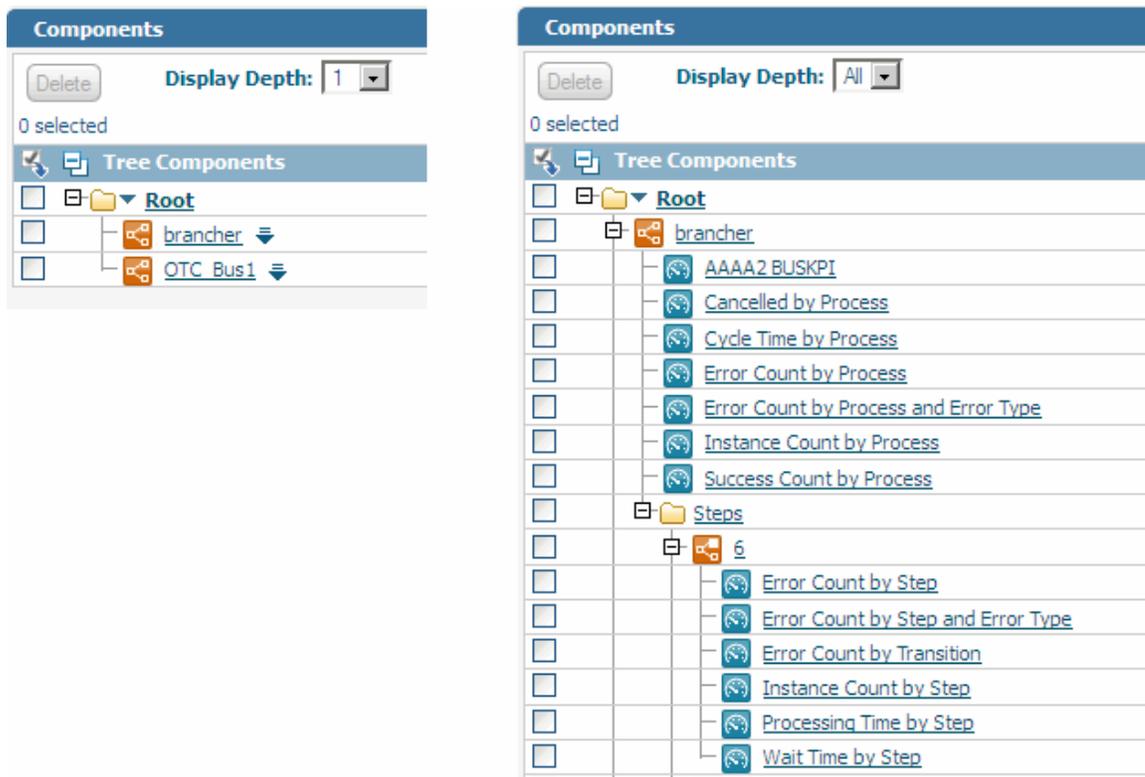
1. In My webMethods, select: **Navigate > Applications > Administration > Analytics > Overview Configuration**
2. To create an overview configuration, click **Add Configuration**. If you want to edit an existing configuration, click the configuration name, or click the Edit icon () to the right of the configuration name.

The Add/Edit Overview Configuration page is displayed.

3. In the **Configuration Information** panel, complete the following fields:
 - **Configuration Name** (required): Type a unique name for the configuration.
 - **Description**: Provide a description of the configuration.
 - **Shared**: Select this check box if you want the configuration to be available for viewing by all users. If this check box is unchecked, the configuration is available only to the user who created the configuration and to system administrators.
4. In the **Components** panel, you can click the question mark icon (across from the **Tree Components** heading) to display a Help dialog box.
5. Also in the **Components** panel, you can specify the **Display Depth** for the component tree. The default setting for **Display Depth** is 2.

The **Display Depth** setting controls the number of levels displayed on the component tree, and it affects the performance of the page display. The more levels that are displayed, the longer the page requires to display, so you should configure the **Display Depth** to show only the levels that you need to see. If the **Display Depth** is configured so that node levels are hidden, a Down arrow icon appears adjacent the applicable nodes to indicate that some node levels are hidden.

The following image shows two views of the component tree. In the view on the left, the **Display Depth** is configured as "1," so only one level of the component tree is visible. In the view on the right, the **Display Depth** is configured as "All," and the entire tree is visible. Note the Down arrow icon beside the tree components on the left view, which indicates that there are additional undisplayed levels.



In the component tree, the target node is the base node from which the tree is expanded according to the Display Depth setting. By default, the Root node is the target node. However, you can designate any node in the component tree as the target node by clicking the desired component name. If you designate a new node as the target node, the component tree is redrawn around that node according to the Display Depth setting. Note that parent nodes *above* a target node display only the direct path between the root node and the target node.

Note: A filter icon on the tree () identifies a node that is a KPI filter.

6. Also in the **Components** panel, you can click the component menu icon () that appears across from the name of each node to access the component menu for that node. The component menu enables you to add any of the following node types to an overview configuration:
 - Folders
 - Processes
 - Process steps
 - Process stages
 - KPIs
 - KPI Filter

In addition, the component menu enables you to perform several editing functions on nodes.

For more information about adding each node type to the component menu, see the appropriate procedure in the "[Working with Configuration Components](#)" on [page 129](#) section that follows. For more information about editing nodes, refer to "[Editing Components on the Component Tree](#)" on [page 133](#).

7. When you complete your changes, save the configuration by clicking **Save**.

Important: If you refresh the page or leave this page without clicking **Save**, everything you've done will be lost.

Deleting a Configuration

Use the following procedure to delete a configuration from the Overview Configuration page.

To delete a configuration on the Overview Configuration page

1. Click the check box to the right of the configuration you want to delete.
2. Click the **Delete** button on the Overview Configuration page.

A dialog box is displayed asking if you want to delete the selected configuration.

3. Click **Yes** to delete the configuration.
4. When you complete your changes, save the configuration by clicking **Save**.

Copying a Configuration

Use the following procedure to copy a configuration on the Overview Configuration page. Note that only shared configurations can be copied.

To copy a configuration on the Overview Configuration page

1. Click the Copy icon () to the right of the configuration you want to copy.

The Add/Edit Configuration page is displayed with the information for the configuration that you copied. The name of the configuration in the **Configuration Name** field is the original name with the word "COPY" appended.

2. Rename the configuration as appropriate, and edit it if desired. Refer to the procedure on the preceding page if you need instructions to edit a configuration.
3. When you complete your changes, save the configuration by clicking **Save**.

Working with Configuration Components

The component menu enables you to manipulate components on the Add/Edit Configuration page configuration tree. The component menu is displayed when you click the component menu icon () beside an applicable node on the component tree.

The component menu is context-sensitive, which means that the component type associated with the specific component menu icon dictates which options appear when that component menu is displayed. The component menu can include the following types of options:

- **Add**, to add components (such as folders, processes, or KPIs) to the component tree. You can add components beneath the **Root** node or beneath another (already existing) node. There is also an **Add KPI Filter** selection to create a filter for the display of KPI instances on the Analytics Overview page.
- **Rename**, to rename the selected node (including the **Root** node).
- **Cut** and **Paste**, to rearrange the position of nodes (along with their children) in the component tree. (You also can use the drag-and-drop method to rearrange the position of nodes in the component tree.)
- **Sort**, to alphabetically sort the selected node's children.
- **Delete**, to delete nodes (along with their children) from the component tree.

Each of these component menu options is described in detail in the specific procedures that follow. In addition, you can find some information on the Help window that is accessible from the user interface.

The procedures in the following sections describe how to add components to the component tree. These procedures assume that you are working from the Add/Edit Overview Component page.

Adding a Folder to the Component Tree

You can add a folder to any existing node on the component tree. Folders can help you to further organize processes and KPIs in the component tree.

To add a folder to the component tree

1. Click  beside an existing node to activate the component menu.
2. Select **Add Folder** from the component menu.

A dialog box is displayed in which you can type a name for the new folder.

3. Type a name for the new folder.
4. Click **OK** to add the folder to the component tree.
5. Repeat these same steps to add as many folders as you desire.
6. When you complete your changes, save the configuration by clicking **Save** or **Apply**.

Important: If you refresh or leave this page without clicking **Save**, everything you've done will be lost.

Adding a Process, Stage or Process Step to the Component Tree

You can add a Process or a Stage to any node on the component tree. Process steps can be added only to processes, within a Steps folder.

This procedure assumes that you are working from the Add/Edit Configuration page. Navigate to **Navigate > Applications > Administration > Analytics > Overview Configuration** and click **Add Configuration**.

To add a process to the component tree

1. Click  beside an existing node to activate the component menu.
2. Select **Add Process** from the component menu.

Optimize displays the Add Process dialog box, listing only those processes in this folder that are not already being tracked.

3. From the list of available processes, select the process you want to add to the component tree.

Optimize populates this process list using a Web service call to retrieve the list of defined processes from the Process Audit database.

4. Click **OK** to add the process to the component tree.

Optimize adds to the component tree a process node and a hierarchy of related components that includes the process steps and intrinsic KPIs.

Note: User-defined process KPIs must be added separately via the **Add KPI** option on the component menu.

5. Repeat the preceding steps to add as many processes as desired.
6. When you complete your changes, save the configuration by clicking **Save** or **Apply**.

Important: If you refresh or leave this page without clicking **Save**, everything you've done will be lost.

Adding a Stage to a Process on the Components Tress

Once you have added processes to a configuration, you can add stages.

To add a stage to a process on the component tree

1. Click  on a node in the component tree that corresponds to a process and choose **Add Process Stages** from the component menu.

Optimize displays the Select Process Stages to Add dialog box, listing only those stages configured for the applicable process that are not already included in the process.

2. From the list of available process stages, select the stages that you want to add to the component tree and click **OK**.

Adding a Process Step to the Components Tree

You also can add specific process steps beneath another existing process steps folder on the components display.

To add a process step to the component tree

1. Click  on a node in the component tree that corresponds to a process and choose **Add Process Step** from the component menu.

Optimize displays the Add Process Step dialog box, listing only those steps for the applicable process that are not already being tracked.

2. From the list of available process steps, select the steps that you want to add to the component tree and click **OK**.

Optimize adds a node for the process step with its intrinsic KPIs to the component tree.

3. When you complete your changes, save the configuration by clicking **Save** or **Apply**.

Important: If you refresh or leave this page without clicking **Save**, everything you've done will be lost.

Adding a KPI or KPI Filter to the Component Tree

You can add a KPI to any node on the component tree. A KPI Filter can be added only to a KPI node.

To add a KPI to the component tree

1. Click  beside the component to which you want to add a KPI.
2. From the component menu, select **Add KPI**.

The Select KPI to Add dialog box is displayed. Note that no KPIs are listed on this dialog until you select a **Category** and **Category Type**.

3. Choose the appropriate category type in the **Category Type** drop-down list. The options are System and Business. Note that business KPIs are specific to Optimize for Process.
4. Choose the appropriate category in the **Category** drop-down list.

The dialog is populated with the KPIs that correspond to the specified **Category Type** and **Category**.

5. Click the appropriate check boxes to select one or more of the KPIs from the list to add to the component tree.
6. Click **OK** to add the KPI(s) to the component tree.

Optimize adds the KPI(s) beneath the selected node in the component tree. You might have to expand the node to see the added KPIs.

7. When you complete your changes, save the configuration by clicking **Save** or **Apply**.

Adding a KPI Filter to the Component Tree

You can create KPI filters and add them to KPI nodes on the component tree. Optimize uses the filters at runtime to select particular KPI instances to display on the Analytics Overview page. You can add as many filters to a KPI node as desired. Note that filters can be shared among sibling KPI nodes on the component tree.

To create a KPI filter

1. Click  beside the KPI node in the component tree to which you want to add a KPI filter and select **Add KPI Filter**.

The Add KPI Filter dialog box is displayed.

2. Type a name for the filter you are creating in the **Filter Name** field.
3. Click the Down arrow to the right of the **Filter Type** field and select Instance Name. This is the only option currently available for this field.
4. If desired, use the **Keyword** text box and drop downs to narrow the list of available KPI instances. This text box accepts standard search keyword notation, which includes the "*" wildcard and quotes.
5. Click the **Preview** button to view the list of KPI instances that are returned by the filter.
6. Click the **Add Filter to Sibling KPIs** check box to automatically share this filter with all other sibling KPIs in the component tree.

Note: Sibling KPIs are KPIs that exist at the same level within the component tree. They are not necessarily adjacent to each other within the tree.

7. Click **OK** to add the KPI Filter to the component tree.
8. When you complete your changes, save the configuration by clicking **Save** or **Apply**.

Important: If you refresh or leave this page without clicking **Save**, everything you've done will be lost.

Editing Components on the Component Tree

You can rename any folder node in the component tree. Other types of nodes cannot be renamed.

To rename a folder node in the component tree

1. Click  beside the desired folder node and choose **Rename** from the component menu.
2. In the dialog box that appears, type a new name for the node and then click **OK**.

- When you complete your changes, save the configuration by clicking **Save** or **Apply**.

Important: If you refresh or leave this page without clicking **Save**, everything you've done will be lost.

- You can move nodes up or down within the component tree either by cutting and pasting or by using drag and drop.

Rearranging Nodes in the Component Tree by Cutting and Pasting

The cut-and-paste method enables you to move a node and all its children to a new location in the component tree. The drag-and-drop method enables you to reorganize “sibling” nodes in the component tree. Regardless of which method you choose, you can move only one node at a time.

To rearrange the position of a node in the component tree by cutting and pasting

- Click  beside the node you want to move.
- Select **Cut** from the component menu; a “cut” icon is displayed next to that node's name in the component tree.
- Click  beside the destination parent node in the component tree (the parent node to which you want to move the node you just cut).
- Select **Paste** from the component menu.

The component tree will be refreshed with the moved node (and its children) displaying in the new location.

Note: Some node types cannot be moved using a cut and paste operation, because they are bound to a particular parent. Also, all destinations in the component tree are verified by Optimize, and an error message is displayed if you attempt to paste a node on an inappropriate parent.

Rearranging Nodes in the Component Tree by Dragging and Dropping

To rearrange the position of a node in the component tree by dragging and dropping

- Position the mouse pointer over the desired node until the “move” cursor () is displayed.
- When the move cursor is visible, hold down the left mouse button while you drag the node to its new location in the component tree. When the node is at the desired location, release the mouse button to “drop” the node and its children into the component tree.
- When you complete your changes, save the configuration by clicking **Save** or **Apply**.

Important: If you refresh the page or leave this page without clicking **Save**, everything you've done will be lost.

Sorting a Node's Children or Descendants in the Component Tree

Use the **Sort** option on the component menu to alphabetically sort a node's children and, optionally, to sort all of its descendants.

To sort a node's children or a node's descendants in the component tree

1. Click  beside the node under which you want to perform a sort and choose **Sort** from the component menu.

A dialog box appears with two sorting options: **Immediate Children** or **All Descendants**. These options are explained below.

- **Immediate Children** sorts only the nodes one level down from the selected node.
 - **All Descendants** sorts all levels of nodes below the selected node.
2. Select the desired sorting option.
 3. When you complete your changes, save the configuration by clicking **Save** or **Apply**.

Important: If you refresh or leave this page without clicking **Save**, everything you've done will be lost.

Deleting Nodes from the Component Tree

Two methods are available for deleting nodes from the component tree. With either method, the node and all of its children are removed from the component tree.

To delete a node from the component tree

1. Use one of the procedures below:
 - For the first deletion method, click  beside the node you want to delete and select **Delete** from the component menu.
 - For the second deletion method, select the check box beside the node you want to delete and then click the **Delete** button.

Note: You cannot delete the **Root** node from the component tree. Also, process step folders cannot be removed by themselves, because the process step folder is required for the addition of process steps.

2. When you complete your changes, save the configuration by clicking **Save** or **Apply**.

Important: If you refresh or leave this page without clicking **Save**, everything you've done will be lost.

Granting User Access to Business Processes

You can control access to business process data in Optimize for Process. With Optimize's data-level security features, you can use roles to specify data-access parameters for users and groups.

Granting a role access to a business process allows anyone in that role to view business process data in Optimize. If a user or group belongs to more than one role, that user or group will have access to the business process data as long as access is granted to at least one of the roles to which the user or group belongs.

After a role is created, access must be assigned. By default, roles do not have access to business processes. Data-level security must be enabled for users and groups to view business process data. If data-level security is enabled, users must be assigned access to view business processes. For instructions for enabling data-level security, see "[Enabling Data-Level Security](#)" on page 75.

To manage data-level security for business processes

1. In My webMethods, navigate the following path: **Navigate > Applications > Administration > System-Wide > User Management > Roles**

Optimize displays the Roles page containing a list of roles. You can add roles and edit or delete existing roles from this page. For information about adding, editing, and deleting roles, see the PDF publication *Administering My webMethods Server*.

2. Do one of the following:

- Enter the name of the role into the **Search Keyword** field and click **Go**.
- Leave the **Search Keyword** field blank and click **Go**.

3. Click the role name, or click  **Edit** beside the role name.

Optimize displays the Edit Role page for the role.

4. On the **Data Level Security** tab, click **Business Process**.

Optimize displays the business processes accessible to this role.

5. From the Edit Role page, you can do any of the following:

<u>To...</u>	<u>Do this...</u>
Grant access to a business process	Click the Add Processes button and continue to " Granting Access to Business Processes " on page 137 below.
Remove a business process from the list	Select the business process and then click the Delete button.

<u>To...</u>	<u>Do this...</u>
Export the list of attributes to a comma-delimited text file	Click the Export Table button. For more information about exporting table data, see <i>Working with My webMethods</i> .

Granting Access to Business Processes

When granting access to a business process, you grant access to a role. Users or groups assigned to that role are subsequently granted access.

To grant access to business processes

1. On the **Data Level Security** tab, click **Business Process**.
2. Click the **Add Processes** button. Optimize displays the Add Business Processes page.
3. From the **Available Business Processes** list, select the business processes to which you would like to grant access for the role. Click the appropriate arrow symbol to move your selection to the **Can Access** list.

To select multiple business processes, hold down the CTRL key while you click the business processes you want to select.

To add all of the available business processes to the **Can Access** list, click the appropriate double-arrow symbol.

To grant access to all business processes currently in the list as well as to all future business processes available for this role, select the option **Access to All Processes** in the **Available Business Processes** list, and click the appropriate arrow symbol to move your selection to the **Can Access** list.

4. Click **OK** to save your choices.

Optimize adds the business processes in the **Business Process** list.

5 Configuring KPIs

■ Overview	140
■ Configuring Business Data	140
■ Configuring System Data	142
■ Defining Dimensions	143
■ Deleting Dimensions	149
■ Mapping Events	150
■ Defining Dimension Hierarchies	164
■ Defining KPIs	167
■ KPI Back Population	172
■ Intelligent Links	173
■ Subscribing to webMethods Events	176
■ Granting User Access to KPIs	177
■ Deleting Corrupted KPI Definitions	179
■ Extracting Optimize Assets from My webMethods Server	179

Overview

When you configure key performance indicators (KPIs) for business and system data, you must complete the following high-level tasks, in sequence:

1. Configure business and system data.
2. Define dimensions, such as customer, region, or host.
3. Map dimensions, measures (formerly facts), transactional data, and intrinsic metrics to a particular business process or system component.
4. Define Dimension hierarchies (formerly referred to as KPI hierarchies) to indicate how dimensions are related. Optimize uses these dimension hierarchical relationships to aggregate data, to drill down for more detail, and to organize data on the business process and system monitoring pages.
5. Define KPIs to associate measures with dimensions. You also can define KPIs that combine business data with process and step metrics, such as Process Cycle Time by Customer.

Important: When Analytic Engine runs in Static DB Schema mode, you cannot create, edit, or delete dimensions, event maps, dimension hierarchies or KPI definitions. In addition, all respective buttons in the user interface will be unavailable. For more information about enabling and disabling Static DB Schema mode for the Analytic Engine, see the PDF publication *Configuring BAM*.

Note: Process and business-related tasks and information are specific to Optimize for Process. System-related tasks and information are generic and apply to both Optimize for Infrastructure and Optimize for Process.

Configuring Business Data

If you are using Optimize for Process, before you can configure business data, you must first do one of the following:

- Define a business process using Software AG Designer. You must then mark the process as “analysis enabled” on the Edit Process page. For more information, see ["Configuring Business Processes" on page 99](#).
- Configure the WmOptimize package. For more information, see ["Installing and Configuring the WmOptimize Package" on page 235](#).
- Configure the Web Service Data Collector to collect business process data. For more information, see ["Configuring and Using the Web Service Data Collector" on page 253](#).

You also must have data for one or more process instances sent to Optimize before you can configure business data.

Important: When Analytic Engine runs in Static DB Schema mode, you cannot create, edit, or delete dimensions, event maps, dimension hierarchies or KPI definitions. In addition, all respective buttons in the user interface will be unavailable. For more information about enabling and disabling Static DB Schema mode for the Analytic Engine, see the PDF publication *Configuring BAM*.

To configure business data

1. In My webMethods, navigate the following path: **Navigate > Applications > Administration > Analytics > KPIs > Business Data**

Optimize displays the Business Data page, which shows event map information for processes and related KPIs.

Specifically, the Business Data page shows the following information:

- The dimensions, measure attributes, event maps, and intrinsic metrics that are associated with a business process, in a hierarchical tree structure.
 - Unassociated event maps that show new types of events that have been collected but are not currently associated with a process.
2. If you have not previously defined dimensions for a business process, continue to "[Defining Dimensions](#)" on page 143. Otherwise, do any of the following:

To...

- Filter business processes in the list
- Search for a particular item
- Search for items that have a particular deployment status

Create an event map for business data that has not yet been associated with a business process

Do this...

Use the **Search** bar. For more information about using the **Search** bar, see *Working with My webMethods*.

Click **Add Event Map** and continue to "[Mapping Events](#)" on page 150.

Note If Optimize has collected business data that is not yet associated with a business process, Optimize lists that data in the hierarchical tree beneath a heading that reads **Event Types not associated with an Event Map**. To associate this business data with a process, click the **Edit** icon to the right of any event type listed under the "unassociated" heading.

To...	Do this...
Add a KPI	Click Add KPI and continue to "Defining KPIs" on page 167 .
Edit an existing event map	Click the Edit icon to the right of the event map and continue to "Mapping Events" on page 150 .
Edit an existing KPI	Click the Edit icon to the right of the KPI and continue to "Defining KPIs" on page 167 .
Delete an event map or a KPI associated with a process	Select the check box beside the event map or the KPI and then click Delete .

Caution Exercise extreme caution when deleting a KPI or especially an event map. Deleting a KPI deletes all associated KPI instances, all rules set on that KPI and all associated historical rule violations. Deleting an event map deletes all associated KPIs, rules and rule violations as well as all data collected for that event map.

Configuring System Data

Before you can configure system data, you must first have in place the Infrastructure Data Collector or a Web Service Data Collector that collects system data.

Important: When Analytic Engine runs in Static DB Schema mode, you cannot create, edit, or delete dimensions, event maps, dimension hierarchies or KPI definitions. In addition, all respective buttons in the user interface will be unavailable. For more information about enabling and disabling Static DB Schema mode for the Analytic Engine, see the PDF publication *Configuring BAM*.

To configure system data

1. In My webMethods: **Navigate > Applications > Administration > Analytics > KPIs > System Data**

Optimize displays the System Data page. The System Data page shows the dimensions, measure attributes, and event maps that are associated with a system component for which system data is being collected. Information on the System Data page is presented in a hierarchical tree structure. The System Data page also displays

unassociated event types, which are new types of events that have come in but that currently are not associated with any event map.

2. Do any of the following:

To...	Do this...
<ul style="list-style-type: none"> ■ Filter system components in the list ■ Search for a particular item ■ Search for items with a particular deployment status 	Use the Search bar. For more information about using the Search bar, see <i>Working with My webMethods</i> .
Add a KPI	Click Add KPI and continue to "Defining KPIs" on page 167 .
Add an event map	Click Add Event Map and continue to "Mapping Events" on page 150
Edit an existing event map	Click the Edit icon next to the event map and continue to "Mapping Events" on page 150 .
Edit an existing KPI	Click the Edit icon next to the KPI and continue to "Defining KPIs" on page 167 .
Delete an event map or a KPI associated with a system component	Select the check box beside the event map or the KPI and then click Delete .

Defining Dimensions

You define dimensions on the Dimensions page.

Important: When Analytic Engine runs in Static DB Schema mode, you cannot create, edit, or delete dimensions, event maps, dimension hierarchies or KPI definitions. In addition, all respective buttons in the user interface will be unavailable. For more information about enabling and disabling Static DB Schema mode for the Analytic Engine, see the PDF publication *Configuring BAM*.

Viewing Existing Dimensions

To view existing dimensions

1. In My webMethods: **Navigate > Applications > Administration > Analytics > KPIs > Dimensions**

Optimize displays the Dimensions page, with the name of each dimension and its internal process or system component name.

Note: Optimize populates the Dimensions page based on the dimensions and attributes declared either in a document or in a business process created using Software AG Designer. In addition, dimensions and attributes can be added when you connect to an Infrastructure Data Collector or to the Web Service Data Collector.

2. Do any of the following:

To...	Do this...
Add a dimension	Click Add Dimension and continue to "Adding a New Dimension" on page 145 .
Edit an existing dimension	Click  Edit beside the dimension name and continue to "Editing an Existing Dimension" on page 147 .
<ul style="list-style-type: none"> ■ Filter dimensions in the list ■ Search for a dimension ■ Search for dimensions that have a particular base internal name or deployment status 	Use the Search bar. For more information about using the Search bar, see <i>Working with My webMethods</i> .
Delete a dimension	Select the check box beside the dimension name and click Delete .
Export the list of dimensions to a *.csv file	Click Export Table . For more information about exporting table data, see <i>Working with My webMethods</i> .

Adding a New Dimension

To add a new dimension

1. In My webMethods: **Navigate > Applications > Administration > Analytics > KPIs > Dimensions.**

Optimize displays the Dimensions page.

2. Click **Add Dimension.**

Optimize displays the Add Dimension page.

3. In the **Name** field, type a name for the dimension (maximum 25 alphanumeric characters).
4. Do one of the following:
 - If data has already come into the system for any dimension type, then dimension types will already exist. From the **Dimension Type** list, select an existing dimension type. If the dimension type you need is absent from the **Dimension Type** list, click the **Other** button next to the **Dimension Type** list and enter a new dimension-type name in the **Other value** field (in the dialog box).

When you select an existing dimension type, any attributes defined for that dimension type will appear in the **Attributes** panel.
 - If no dimension-type data exists yet in the system, then you must create a new dimension type. Simply enter the new dimension-type name in the **Dimension Type** text box.
5. To associate an intelligent link with this dimension, select an option from the **Intelligent Link** list box.

For more information about intelligent links, see ["Intelligent Links" on page 173](#). The **Intelligent Link** list box appears here only if intelligent links already have

been configured. Once configured, intelligent links appear on the following My webMethods pages:

- Problems page
- KPI Instance Detail page, on the data version of the page (as opposed to the graph/detail version of the page)
- Process Instance Detail page
- Step Instance Detail page

6. To add attributes in the **Attributes** panel, refer to the following procedure:

Note: Attributes that are automatically imported with a dimension type cannot be renamed.

- a. In the **Attribute Name** field, type a name for the attribute (maximum 25 alphanumeric characters). Each dimension attribute must have a unique name, and case is *not* a differentiator. In other words, the database sees `Attribute1` and `attribute1` as the same name.

Caution: Optimize databases do *not* distinguish between upper- and lowercase alphabetic characters, which means `Attribute1` and `attribute1` are the same name to the databases.

- b. If you selected an intelligent link to associate with the dimension, you can optionally associate an intelligent-link parameter with this attribute by selecting from the **Link Parameter** drop-down menu. Position the mouse pointer over a parameter name to display the description of that parameter. If you did not select an intelligent link, the text **Intelligent Link has no Parameters** is displayed.
- c. If you want to indicate that the attribute is a unique identifier for the dimension, select the **Unique** check box. Because unique attributes are by definition required, the **Required** check box is automatically selected when you select the **Unique** check box. Clear the **Unique** check box to indicate that the attribute is not a unique identifier for the dimension. For example, a Customer Number attribute might be unique but a Region attribute might not. You must identify at least one unique attribute for the dimension.
- d. If the attribute is required, select the **Required** check box. Clear the check box to indicate that the attribute is not required for the dimension.

Keep in mind that if you set an attribute to **Required**, a new record for that dimension will be rejected if that attribute is not available. A dimension can have more than one **Required** attribute. For example, assume you have a Customer dimension that has the attributes Customer ID and Customer Name. Customer ID is set to both **Unique** and **Required**, and Customer Name is set only to **Required**. Whenever you create a new record for the Customer dimension, the record will be rejected unless both Customer ID and Customer Name are supplied. However, only the Customer ID attribute must be unique.

- e. To add more attributes, click **Add Attribute** to add a new row at the bottom of the **Attributes** list and repeat the preceding steps.
7. To assign more attributes to the dimension, or to remove any attributes, go to the Monitored Components page. See "[Discovering Assets and Monitoring Components](#)" on page 195 for details.
8. If you are ready for Optimize to create a database table and begin collecting dimension data, click **Save**.

When you click **Save**, Optimize allocates database tables to store the dimension attributes in the database and also begins to process incoming data for the dimension. When you save a dimension, any changes to dimension attributes will cause previously collected data to be discarded.
9. If you have not previously mapped dimensions to a business process or system component, continue to "[Mapping Events](#)" on page 150.

Editing an Existing Dimension

Important: Exercise extreme caution when editing a dimension. Changing a dimension's attributes will cause the destruction of all associated definitions: event map definitions, dimension hierarchies, KPI, rules and all their collected runtime data. Only dimension name and intelligent link information can be safely modified.

To edit an existing dimension

1. In My webMethods: **Navigate > Applications > Administration > Analytics > KPIs > Dimensions**.

Optimize displays the Dimensions page.

2. Click  **Edit** beside the dimension you want to edit.

The Edit Dimension page is displayed.

Note: Only dimensions that have been configured through the user interface can be modified at the Edit Dimension page. You can view system-defined dimensions such as Step or Database, but you can make only limited modifications.

Dimensions > Edit Dimension Save Cancel

Dimension Information

* Name:

Dimension Type: AA Dim Type

Intelligent Link:

Attributes

Attribute Name	Link Parameter	Unique	Required
<input type="checkbox"/> AA Attribute	No Intelligent Link Selected	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Save Cancel

3. You can change the name of the dimension or enter a new name.
4. You cannot change the **Dimension Type** from its original setting.
5. To associate an intelligent link with this dimension, select an option from the **Intelligent Link** list box.

For more information about intelligent links, see "[Intelligent Links](#)" on page 173. The **Intelligent Link** list box appears here only if intelligent links already have been configured. Once configured, intelligent links appear on the following My webMethods pages:

- Problems page
 - KPI Instance Detail page, on the data version of the page (as opposed to the graph/detail version of the page)
 - Process Instance Detail page
 - Step Instance Detail page
6. In the **Attributes** panel, do the following:
 - a. To change the name of an existing attribute, enter a new unique name in the **Attribute Name** field. Attributes that are imported from a dimension type cannot be renamed.
 - b. To add a new attribute for this dimension, click **Add Attribute**. Optimize inserts a new row at the bottom of the **Attributes** list. In the **Attribute Name** field, type a unique name for the attribute.
 - c. If an intelligent link is associated with the dimension, you can specify a parameter to use with the intelligent link for this attribute by selecting from the **Link Parameter** drop-down menu.
 - d. To indicate that the attribute is a unique identifier for the dimension, select the **Unique** check box. Clear the check box to indicate that the attribute is not a unique identifier for the dimension. For example, a Customer Number attribute might be unique but a Region attribute might not. You must identify at least one unique attribute for the dimension.

- e. If the attribute is required, select the **Required** check box. Clear the check box to indicate that the attribute is not required for the dimension.

Keep in mind that if you set an attribute to **Required**, a new record for that dimension will be rejected if that attribute is not available. A dimension can have more than one **Required** attribute. For example, assume you have a Customer dimension that has the attributes Customer ID and Customer Name. Customer ID is set to both **Unique** and **Required**, and Customer Name is set only to **Required**. Whenever you create a new record for the Customer dimension, the record will be rejected unless both Customer ID and Customer Name are supplied. However, only the Customer ID attribute must be unique.

- f. To delete an attribute, select the check box beside the attribute name and click **Delete**.
7. If you are ready for Optimize to create a database table and begin collecting dimension data, click **Save**.

When you click **Save**, Optimize allocates database tables to store the dimension attributes in the database and also begins to process incoming data for the dimension. When you save a dimension, any changes to dimension attributes will cause previously collected data to be discarded.

Deleting Dimensions

You can delete any user-defined dimensions. System dimensions cannot be deleted. In addition, you cannot delete any dimensions that are referenced by any of the following:

- An event map
- A dimension hierarchy
- A rule filter

To delete a Dimension

1. In My webMethods navigate the following path: **Navigate > Applications > Administration > Analytics > KPIs > Business Data** (Optimize for Process only) or **Navigate > Applications > Administration > Analytics > KPIs > Dimensions**

The Dimensions page is displayed, showing all dimensions configured for your system.

2. Select the dimensions you want to delete by clicking the appropriate boxes located to the left of the dimension name.
3. Click the **Delete** button to delete the selected dimensions.

A dialog is displayed asking if you are sure that you want to delete the selected dimensions.

4. Click **OK** to proceed with the dimension deletion.

A Dimension Deletion Results dialog is displayed. This dialog shows all dimensions that were deleted, based on your selections, as well as any dimensions that could not be deleted due to system dependencies. Deleted dimensions are marked with a square green icon, and dimensions that could not be deleted are marked with a red circle icon.

Mapping Events

When configuring Optimize KPIs, you must map incoming events to data types that are meaningful within Optimize on the Business Data page or the System Data page. These event mappings identify event data elements that are of interest, such as dimensional data, business data and numeric data (measures), which can be used by Optimize analytics. Optimize enables you to work with both intrinsic events (those that Optimize is designed specifically to collect) as well as events from other applications that are available via the webMethods Event Server. Note that when configuring events from the webMethods Event Server, you must use the Event Subscription page to subscribe to the desired events in order for them to be available within Optimize.

Important: When Analytic Engine runs in Static DB Schema mode, you cannot create, edit, or delete dimensions, event maps, dimension hierarchies or KPI definitions. In addition, all respective buttons in the user interface will be unavailable. For more information about enabling and disabling Static DB Schema mode for the Analytic Engine, see the PDF publication *Configuring BAM*.

Adding a New Event Map

You can add a new event map by clicking the **Add Event Map** button on the **KPIs > Business Data** page or the **KPIs > System Data** page. There are two ways to create an event map in Optimize. The first is to build an event map from scratch by specifying a new event type and manually adding each event field. The second approach is to select a known event type which will automatically add the event fields.

Note: Note that it is not recommended that you create an event map using fields which generate a unique value with every transaction. If you use such a field in your event map, Optimize creates a new KPI instance every time a new event of this type is received. This results in database tables with extremely large size and makes any further data processing cumbersome.

To create an event map from scratch, click the **Add Event Map** button and enter a new event type on the Event Map page. You must ensure that the event type matches the incoming event type name exactly.

To create an event map from a known event type, you can either select the event type from a listing of unmapped event types, which will open the Event Map page, or you can click the **Add Event Map** button and select an event type from the drop-down list. Both methods will pre-populate the mapped fields table from the fields in the event type.

Note that unmapped event types are events that have been sent to the Analytic Engine prior to event mapping. To view all unmapped event types for your system, you can search for “unassociated” from the Business Data or System Data pages, and the unmapped event types will be displayed under an **Unmapped Event Types** heading in the KPIs region on the lower part of the page.

When you add an event map, you must configure some or all of the following information:

- Name the event map.
- Specify the event type.
- Associate a business process to the event.
- Map fields to the event.

When you map fields to an event, you might add new fields, remove existing fields, and choose or remove intrinsic fields (for intrinsic metrics). In addition, working with fields for events can involve specifying the data type for the field, associating an intelligent link with a dimension, and specifying a calculation type and a unit of measure for a measure data type. Note that fields cannot be added or removed from a map that is based on a subscribed event type.

Also, when working with event maps, you can add intrinsic process metrics to define KPIs that combine business data with process and step metrics (such as Process Cycle Time by Customer).

The following procedure describes the specific steps to create a basic event map.

To add a basic event map

1. In My webMethods navigate the following path: **Navigate > Applications > Administration > Analytics > KPIs > Business Data** (Optimize for Process only) or **Navigate > Applications > Administration > Analytics > KPIs > System Data**
2. Click **Add Event Map**. Alternatively, you can search for unassociated to display Unmapped Event Types and then click on the desired event type.

The Add Event Map page is displayed.

3. In the **Event Information** panel, complete the following steps:
 - a. In the **Name** text box, type a name for the event map.
 - b. Select or enter an event type for this event map:
 - If the system has already collected data for an event type, then an event type will already exist. From the **Event Type** list, select an existing event type for this event map.
 - If the system has not collected data for this event type, then you must create a new event type. Click **Other**, and then enter the name of the new event type in the **Other value** pop-up field.

- If you want to create an event map based on a subscribed event type, you must first use the Event Subscription page to subscribe to the desired event type. After you subscribe to events, they are available for use in Optimize.
- c. If you are working with Business Data, select a business process with which you want to associate business data for this event map from the **Business Process** list. If the data is not associated with a process, leave **None** in the **Business Process** list.

Associate an event map with a business process if the business data that is sent to Optimize for that event map is sent as part of process instance control operations (step control operations). Associating an event map with a business process allows for process and step metrics to be associated with business data that is stored against that event map.

Note: The **Business Process** list is not displayed if you click **Add Event Map** on the System Data page.

4. If you selected an existing event type, each field known to that event type is automatically added to the **Field Mapping** panel. You can edit each field as needed, and you delete any fields that are not needed by selecting them and clicking the **Delete** button. Remember that fields cannot be added for subscribed event types.
5. To assign more field mappings (for example, if you are using the Web Service Data Collector and you start collecting new types of data), click **Add Field** and complete the fields in the **Field Mapping** dialog box as many times as necessary.

For each field, you must configure the mapping by assigning the field type and various related information that will be used by the Analytic Engine. To configure a field click either the field name or the **Edit** icon. Either action activates a dialog with configuration options.

6. In the **Field Mapping** dialog box, complete the following steps for each new field you want to map:
 - a. Enter a name for the field in the **Field** text box. (Fields automatically imported from an existing event type cannot be renamed.)
 - b. In the **Display Name** field, enter a user-friendly name to display for the field on Optimize pages. If the field name is user friendly, you can also use it as the display name.
 - c. From the **Type** list, select a data type. Depending on your selection in the **Type** list, the following lists or text boxes appear in the bottom half of the dialog box. Make selections from these lists or enter text as appropriate:

Data Type	Description
Dimension	A <i>dimension</i> represents data that you can categorize, query, and associate with a KPI (such as region or host name). Choose between an Existing Dimension or a New Dimension .

Data Type	Description
Transaction	<ul style="list-style-type: none"> <li data-bbox="618 327 1312 394">■ Select Existing Dimension to link the event map to a dimension that already exists. Select the name of the existing dimension from the Dimension list, and select a value for the dimension in the Attribute list. <li data-bbox="618 537 1268 604">■ Select New Dimension to link the event map to a dimension that does not yet exist. Enter a name for the dimension in the Dimension text box, and enter a value for the dimension in the Attribute text box. <p data-bbox="618 768 1304 1073">A <i>transaction</i> is an attribute that you do not want to measure but that you would like to store in the database to use in a rule expression or to associate with a fact that arrives with the same event. Select the Transaction type to specify additional fields in a document, such as document identifiers, order numbers, or error messages. Optimize displays these fields on the KPI Instance Detail page and in alerts from rules defined for the event.</p> <p data-bbox="618 1100 1308 1335">To link the transaction data to a URL specified in an intelligent link (see "Intelligent Links" on page 173), select the name of the link from the Intelligent Link list, and select an optional parameter for the link from the Parameter list. Position the mouse pointer over a parameter name to display a description of the parameter.</p>
Measure	<p data-bbox="618 1381 1336 1549">A <i>measure</i> (formerly known as a fact) is a measurable value (such as revenue or queue length) that the Analytic Engine can use for calculation or analysis. Select a calculation type and enter a unit of measure for the value:</p> <ul style="list-style-type: none"> <li data-bbox="618 1577 867 1606">■ Calculation Type: <li data-bbox="618 1629 1279 1696">■ Select Sum to total the data collected during the collection interval. <li data-bbox="618 1719 1367 1787">■ Select Average to average the data collected during the collection interval. <li data-bbox="618 1810 1260 1877">■ Select Last Value to use only the last data point collected.

Data Type	Description
Date	<ul style="list-style-type: none"> ■ Select State to aggregate the binary state of the data point (such as online or offline). ■ Select Count to count the data points during the collection interval. ■ In the UOM text box, type a unit of measure (such as Dollars or Bytes). If you want the unit of measure to be automatically formatted when the KPI results are displayed, use one of the keywords described in the following step (keywords are not case sensitive). <p>Choose this data type to specify that the field is a date. Date fields are used for process deadlines (see "Configuring Business Processes" on page 99, the section titled Deadline Standardized Error Types, for more information).</p>

7. In the **Field Mapping** dialog box, if you chose **Measure** as the data type, you can use one of the following keywords in the **UOM** field so the unit of measure is automatically formatted when the KPI results are displayed:

Unit of Measure Description	Keyword(s) for UOM Field	Display in KPI Results
Dollars	"dollars" or "\$"	<value to two decimals>\$
Thousands of dollars	"\$1000" Multiplies the result by 1000.	<value>,000.00\$
Percentage	"% " or "percent" Using "% " multiplies the result by 100.	<value to two decimals>%
Size in bytes	bytes	<value to two decimals> bytes
Size in kilobytes	k kbytes kilobytes	<value to two decimals>k
Size in megabytes	mb megabytes	<value to two decimals> MB
clients	clients	<value to one decimal >
connection count	connection count	<value to one decimal >

<u>Unit of Measure Description</u>	<u>Keyword(s) for UOM Field</u>	<u>Display in KPI Results</u>
connections	connections	<value to one decimal >
Item count	count	<value to one decimal >
number of documents	documents	<value to one decimal >
number of errors	errors	<value to one decimal >
number of files	files	<value to one decimal >
insert row count	insert row count	<value to one decimal >
number of instances	instances	<value to one decimal >
number of items	items	<value to one decimal >
monitor count	monitor count	<value to one decimal >
number of records	records	<value to one decimal >
row count	row count	<value to one decimal >
threads	threads	<value to one decimal >
thread count	thread count	<value to one decimal >
traces	traces	<value to one decimal >
time in milliseconds	ms millis milliseconds	<hh:mm:ss> Hours <mm:ss> Minutes <ss:mm> Seconds <mmm> Milliseconds
time in seconds	sec seconds	<hh:mm:ss >
Online/offline	state	0 = offline 1 = online

- To assign more field mappings (for example, if you are using the Web Service Data Collector and you start collecting new types of data), click **Add Field** again and complete the fields in the **Field Mapping** dialog box as many times as necessary.

9. To delete a field mapping, select the check box beside the field name and then click **Delete**.
10. If you are ready for Optimize to create a database table and begin collecting event-map data, click **Save**.

When you click **Save**, Optimize allocates database tables to store the event map in the database and also begins to process incoming data for the event map. When you save an event map, any changes to the event map will cause previously collected data to be discarded.

11. To define a KPI hierarchy for this business process or system component, continue to ["Defining Dimension Hierarchies" on page 164](#).

Adding an Event Map with New Fields

To add an event map with new fields

1. In My webMethods: **Navigate > Applications > Administration > Analytics > KPIs > Business Data** (Optimize for Process only) or **Navigate > Applications > Administration > Analytics > KPIs > System Data**.

2. Click **Add Event Map**.

The Add Event Map page is displayed.

3. To map a new field, click **Add Field** in the **Field Mapping** panel.

The **Add Intrinsic Fields** is applicable only if you want to combine business data with process and step metrics. See ["Adding an Event Map that Combines Business Data with Process and Step Metrics" on page 160](#) for more information.

4. In the **Field Mapping** dialog box, complete the following steps for each new field you want to map:
 - a. Enter a name for the field in the **Field** text box. (Fields automatically imported from an existing event type cannot be renamed.)
 - b. In the **Display Name** field, enter a user-friendly name to display for the field on Optimize pages. If the field name is user friendly, you can also use it as the display name.
 - c. From the **Type** list, select a data type. Depending on your selection in the **Type** list, the following lists or text boxes appear in the bottom half of the dialog box. Make selections from these lists or enter text as appropriate:

Data Type	Description
Dimension	A <i>dimension</i> represents data that you can categorize, query, and associate with a KPI (such as region or host name). Choose between an Existing Dimension or a New Dimension .

Data Type	Description
Transaction	<ul style="list-style-type: none"> <li data-bbox="618 327 1312 394">■ Select Existing Dimension to link the event map to a dimension that already exists. Select the name of the existing dimension from the Dimension list, and select a value for the dimension in the Attribute list. <li data-bbox="618 537 1268 604">■ Select New Dimension to link the event map to a dimension that does not yet exist. Enter a name for the dimension in the Dimension text box, and enter a value for the dimension in the Attribute text box. <p data-bbox="618 768 1304 1073">A <i>transaction</i> is an attribute that you do not want to measure but that you would like to store in the database to use in a rule expression or to associate with a fact that arrives with the same event. Select the Transaction type to specify additional fields in a document, such as document identifiers, order numbers, or error messages. Optimize displays these fields on the KPI Instance Detail page and in alerts from rules defined for the event.</p> <p data-bbox="618 1100 1312 1335">To link the transaction data to a URL specified in an intelligent link (see "Intelligent Links" on page 173), select the name of the link from the Intelligent Link list, and select an optional parameter for the link from the Parameter list. Position the mouse pointer over a parameter name to display a description of the parameter.</p>
Measure	<p data-bbox="618 1381 1338 1549">A <i>measure</i> (formerly known as a fact) is a measurable value (such as revenue or queue length) that the Analytic Engine can use for calculation or analysis. Select a calculation type and enter a unit of measure for the value:</p> <ul style="list-style-type: none"> <li data-bbox="618 1577 867 1606">■ Calculation Type: <li data-bbox="618 1629 1279 1696">■ Select Sum to total the data collected during the collection interval. <li data-bbox="618 1719 1367 1787">■ Select Average to average the data collected during the collection interval. <li data-bbox="618 1810 1260 1877">■ Select Last Value to use only the last data point collected.

Data Type	Description
Date	<ul style="list-style-type: none"> ■ Select State to aggregate the binary state of the data point (such as online or offline). ■ Select Count to count the data points during the collection interval. ■ In the UOM text box, type a unit of measure (such as Dollars or Bytes). If you want the unit of measure to be automatically formatted when the KPI results are displayed, use one of the keywords described in the following step (keywords are not case sensitive). <p>Choose this data type to specify that the field is a date. Date fields are used for process deadlines (see "Configuring Business Processes" on page 99, the section titled Deadline Standardized Error Types, for more information).</p>

5. In the **Field Mapping** dialog box, if you chose **Measure** as the data type, you can use one of the following keywords in the **UOM** field so the unit of measure is automatically formatted when the KPI results are displayed:

Unit of Measure Description	Keyword(s) for UOM Field	Display in KPI Results
Dollars	"dollars" or "\$"	<value to two decimals>\$
Thousands of dollars	"\$1000" Multiplies the result by 1000.	<value>,000.00\$
Percentage	"% " or "percent" Using "% " multiplies the result by 100.	<value to two decimals>%
Size in bytes	bytes	<value to two decimals> bytes
Size in kilobytes	k kbytes kilobytes	<value to two decimals>k
Size in megabytes	mb megabytes	<value to two decimals> MB
clients	clients	<value to one decimal >
connection count	connection count	<value to one decimal >

<u>Unit of Measure Description</u>	<u>Keyword(s) for UOM Field</u>	<u>Display in KPI Results</u>
connections	connections	<value to one decimal >
Item count	count	<value to one decimal >
number of documents	documents	<value to one decimal >
number of errors	errors	<value to one decimal >
number of files	files	<value to one decimal >
insert row count	insert row count	<value to one decimal >
number of instances	instances	<value to one decimal >
number of items	items	<value to one decimal >
monitor count	monitor count	<value to one decimal >
number of records	records	<value to one decimal >
row count	row count	<value to one decimal >
threads	threads	<value to one decimal >
thread count	thread count	<value to one decimal >
traces	traces	<value to one decimal >
time in milliseconds	ms millis milliseconds	<hh:mm:ss> Hours <mm:ss> Minutes <ss:mm> Seconds <mmm> Milliseconds
time in seconds	sec seconds	<hh:mm:ss >
Online/offline	state	0 = offline 1 = online

- To assign more field mappings (for example, if you are using the Web Service Data Collector and you start collecting new types of data), click **Add Field** again and complete the fields in the **Field Mapping** dialog box as many times as necessary.

7. To delete a field mapping, select the check box beside the field name and then click **Delete**.
8. If you are ready for Optimize to begin collecting event data, click **Save**.
When you click **Save**, Optimize allocates database tables to store the event data in the database and also begins to process incoming data for the event map.
9. To define a dimension hierarchy for the dimensions defined in this business process or system component, continue to ["Defining Dimension Hierarchies" on page 164](#).

Adding an Event Map that Combines Business Data with Process and Step Metrics

To add an event map that combines business data with process and step metrics in Optimize for Process

1. In My webMethods navigate the following path: **Navigate > Applications > Administration > Analytics > KPIs > Business Data**
2. Click **Add Event Map**.
The Add Event Map page is displayed.
3. Complete the fields in the **Event Information** panel, making sure a business process is selected in the **Business Process** field.
4. To combine business data with process and step metrics, click **Add Intrinsic Fields** in the **Field Mapping** panel.
5. In the popup list, select the intrinsic process metrics you want to include with this event map. The list shows only those intrinsic process metrics not already defined in the event map.

The following intrinsic process metrics are available:

<u>Field (Process Metric) Name</u>	<u>Display Name and Type</u>	<u>Description</u>
Process	Name: Process Type: Dimension	The process model external ID, from the BAM_PT_PROCESS_META table (process key from WMPROCESSDEFINITION). This is the unique identifier of the process model.
ProcessInstanceId	Name: Process Instance Id Type: Transaction	The unique identifier of a process instance. The primary key of the process instance record in the BAM_PT_PROCESS_

<u>Field (Process Metric) Name</u>	<u>Display Name and Type</u>	<u>Description</u>
		INST table.
Step	Name: Step Type: Dimension	A unique step identifier within a process. The step external ID in the BAM_PT_STEP_META table (step ID in WMSTEPDEFINITION).
StepIteration	Name: Step Iteration Type: Dimension	The step iteration.
ProcessError	Name: Process Error Type: Dimension	The error type of a process or step error. This maps to the <code>ErrorType</code> attribute of the <code>ProcessError</code> dimension.
process_instance_count	Name: Process Instance Count Type: Fact (SUM)	The indication of a new process instance. This metric is always set to either 0 or 1.
process_cycle_time	Name: Process Cycle Time Type: Fact (AVG) in milliseconds	The duration of a process instance (the sum of the step processing times for all step iterations of all steps).
process_success_count	Name: Process Success Count Type: Fact (SUM)	The indication of a process instance that completed without error.
process_error_count	Name: Process Error Count Type: Fact (SUM)	The indication of one or more errors associated with a process instance.

Note: Process Engine executes the process, the process instance is cancelled shortly after an error occurs. This generates an additional event with

<u>Field (Process Metric) Name</u>	<u>Display Name and Type</u>	<u>Description</u>
		the following readings: process_error_count=0 and process_cancelled_count=1.
process_cancelled_count	Name: Process Cancellation Count Type: Fact (SUM)	The indication of a process instance that either was cancelled, or failed when executed by Process Engine.
step_error_count	Name: Step Error Count Type: Fact (SUM)	The number of step errors.
step_processing_time	Name: Step Processing Time Type: Fact (AVG) in milliseconds	The duration of a step (the time between the start and stop operations of a step).
step_wait_time	Name: Step Wait Time Type: Fact (AVG) in milliseconds	The duration of the transition between two steps (the time between the stop operation of the source step and the start operation of the target step, where the Step field is the target step of the transition).
step_instance_count	Name: Step Instance Count Type: Fact (SUM)	The indication of a new step instance for a given process instance.

Editing an Existing Event Map

You can edit a previously created event map using the Edit Event Map page.

Important: You can edit and save a user defined event map, but certain changes may cause the loss of dependent KPI and Rule definitions as well as any related event and analytical data collected. When saving, Optimize will analyze your changes and warn you if any loss of data may occur. You then have the option to either cancel the changes or proceed with the save.

Note: You can view system-defined event maps such as StationHealth or ProcessMetrics on the Edit Event Map page, but you cannot modify them.

Note: You can edit event maps that are based on subscribed event types but you cannot add or remove event fields from them.

To edit an existing event map

1. In My webMethods, navigate the following path: **Navigate > Applications > Administration > Analytics > KPIs > Business Data** (Optimize for Process only) or **Navigate > Applications > Administration > Analytics > KPIs > System Data** (Optimize for Process and Optimize for Infrastructure).

2. Click the name of an event map, or click  **Edit** beside the event map you want to edit.

The Edit Event Map page is displayed.

3. In the **Event Information** panel, make any desired changes as appropriate for your selection.
 - If you are editing a System Data event map, the **Event Information** panel displays the **Name** and **Event Type** for the Event. Neither of these fields can be edited.
 - If you are editing a Business Data event map the **Event Information** panel displays the **Name** and **Event Type** for the Event as well as the **Business Process** associated with the map. You can edit the business process association by clicking the Down arrow to the left of the **Business Process** field and selecting a different business process.
4. In the **Field Mapping** panel, click **Add Field** to add a new field, or click  **Edit** beside a field name to reconfigure an existing field and make any desired changes in the **Field Mapping** dialog box. For more details about any of these fields, see ["Adding a New Event Map" on page 150](#).

Note: You cannot add or remove fields for event maps based on subscribed event types.

5. If you want to add intrinsic fields, click **Add Intrinsic Fields**. For more information about intrinsic fields, see ["Adding an Event Map that Combines Business Data with Process and Step Metrics" on page 160](#).
6. To add a new field mapping (for example, if you are using the Web Service Data Collector and you start collecting new types of data), click **Add Field** and complete the fields in the **Field Mapping** dialog box as many times as necessary.
7. If you are ready for Optimize to begin collecting event data, click **Save**.

When you click **Save**, Optimize allocates database tables to store the event data in the database and also begins to process incoming data for the event map. You can edit and save a user defined event map, but certain changes may cause the loss of dependent KPI and Rule definitions as well as any related event and analytical data collected. When saving, Optimize will analyze your changes and warn you if any

loss of data may occur. You then have the option to either cancel the changes or proceed with the save.

Note: You can edit and save a user defined event map, but certain changes may cause the loss of dependent KPI and rule definitions as well as any related event and analytical data collected. When saving, Optimize will analyze your changes and warn you if any loss of data may occur. You then have the option to either cancel the changes or proceed with the save.

Deleting an Event Map

You can delete an event map at the KPIs > Business Data page or the KPIs > System Data page.

Note: You cannot delete system-defined event maps such as StationHealth or ProcessMetrics.

To delete an event map

1. In My webMethods, navigate the following path: **Navigate > Applications > Administration > Analytics > KPIs > Business Data** (Optimize for Process only) or **Navigate > Applications > Administration > Analytics > KPIs > System Data**.
2. Select the check box beside the event map(s) you want to delete.
3. Click **Delete**.

Defining Dimension Hierarchies

You define KPI hierarchies from the Dimension Hierarchies page. Dimension hierarchies illustrate relationships between KPIs. Optimize uses Dimension hierarchies to complete the following tasks:

- Evaluate KPI relationships when performing trend analysis to determine how a lower-level, or child, KPI might affect its parent KPI.

For example, suppose you define a profit KPI as a high-level KPI in a hierarchy and you create a rule that alerts you when that profit KPI is below normal. Then you define a second KPI to monitor revenue by customer region. You define that second (revenue) KPI as a child of the profit KPI, and you create a rule that alerts you when that revenue KPI is below normal. If a rule violation occurs for the parent (profit) KPI, Optimize then will evaluate the children of that KPI. Based on the relationship you defined, if Optimize discovers that a child KPI also has gone out of compliance with a rule (for example, if the revenue of a specific region is below normal), Optimize can suggest that the child KPI is a probable cause of the rule violation for the parent KPI.

- Display KPI relationships on the Business Data (Optimize for Process) and System Data pages.

- Build a monitor tree, which presents a structural view of the monitored systems and business processes on the Analytics Overview page.

Important: When Analytic Engine runs in Static DB Schema mode, you cannot create, edit, or delete dimensions, event maps, dimension hierarchies or KPI definitions. In addition, all respective buttons in the user interface will be unavailable. For more information about enabling and disabling Static DB Schema mode for the Analytic Engine, see the PDF publication *Configuring BAM*.

Viewing Existing Dimension Hierarchies

To view existing Dimension hierarchies

1. In My webMethods, navigate the following path: **Navigate > Applications > Administration > Analytics > KPIs > Dimension Hierarchies**
Optimize displays the Dimension Hierarchies page.
2. Do any of the following:

To...	Do this...
Search or filter hierarchies in the list	Use the Search bar. For more information about using the Search bar, see <i>Working with My webMethods</i> .
Edit an existing hierarchy	Click the Edit icon next to the hierarchy and continue to "Adding or Editing a Dimension Hierarchy" on page 165.
Add a hierarchy	Click the Add Hierarchy button and continue to "Adding or Editing a Dimension Hierarchy" on page 165.
Delete a hierarchy	Select the check box beside the hierarchy and then click Delete .

Adding or Editing a Dimension Hierarchy

To add or edit a Dimension hierarchy

1. In My webMethods, navigate the following path: **Navigate > Applications > Administration > Analytics > KPIs > Dimension Hierarchies**
The Dimension Hierarchies page is displayed.
2. Do one of the following:

<u>To...</u>	<u>Do this...</u>
Add a new hierarchy	Click the Add Hierarchy button.
Edit an existing hierarchy	Click  Edit beside the hierarchy.
Optimize displays the Add/Edit Hierarchy page.	

Note: System-defined hierarchies such as Station cannot be modified using the Add/Edit Hierarchy page.

3. On the **Hierarchy Information** panel, in the **Name** field, type a name for the hierarchy.
4. On the **Hierarchy** panel, do the following steps:
 - a. From the first list on a row, select a dimension (such as Customer).
 - b. From the second list on a row, select an attribute (such as Region).
5. To add a child beneath a row, click **Add Child**. Repeat this step and the preceding one as often as necessary until you have created all of the desired children.

For example, if you wanted to define a hierarchy stating that Region is a parent to Branch and Branch is a parent to SalesRep, your hierarchy would look like this:

Dimension / Attribute

Customer / Region

Customer / Branch

Sales Rep/ Name

6. Click **Save**.
7. If you have not previously defined a KPI for this business process or system component, continue to ["Defining KPIs" on page 167](#).

Deleting Dimension Hierarchies

You can delete Dimension hierarchies as desired, with some restrictions. System-bound hierarchies are grayed out and cannot be selected for deletion. Also, you cannot delete a Dimension hierarchy if there are KPIs dependent upon that hierarchy.

To delete Dimension hierarchies

1. In My webMethods, navigate the following path: **Navigate > Applications > Administration > Analytics > KPIs > Dimension Hierarchies**
The Dimension Hierarchies page is displayed.

- Click the check boxes to the left of the hierarchies you want to delete to select them, and then click the **Delete** button.

A dialog is displayed asking if you are sure that you want to delete the selected hierarchies.

- Click **OK** to delete the selected hierarchies.

A Dimension Hierarchy Deletion Results dialog is displayed. This dialog shows all hierarchies that were deleted, based on your selections, as well as any hierarchies that could not be deleted due to system dependencies. Deleted hierarchies are marked with a square green icon, and hierarchies that could not be deleted are marked with a red circle icon.

Defining KPIs

The Add/Edit KPI page in Optimize enables you to create and edit KPIs. You can activate this page from the Business Data page in Optimize for Process or from the System Data page in either version of Optimize. For more information about the elements that make up KPIs and a description of how KPIs function within Optimize, see the “Business Visualization Concepts” chapter in *webMethods Optimize User’s Guide*. Also, when creating KPIs it is helpful to understand how event maps function in Optimize. For more information about event maps, refer to ["Mapping Events" on page 150](#).

Important: When Analytic Engine runs in Static DB Schema mode, you cannot create, edit, or delete dimensions, event maps, dimension hierarchies or KPI definitions. In addition, all respective buttons in the user interface will be unavailable. For more information about enabling and disabling Static DB Schema mode for the Analytic Engine, see the PDF publication *Configuring BAM*.

Optimize enables you to define two types of KPIs: *individual* KPIs and *composite* KPIs.

- **Individual KPIs** associate one measure with one dimension attribute, such as revenue by region or queue length by host. When you define an individual KPI, Optimize creates a KPI instance for each attribute of the dimension defined for that KPI. For example, if you define a KPI to measure revenue by region and you have three regions, Optimize creates three separate KPI instances to collect revenue data for the three regions.
- **Composite KPIs** consist of the result of a mathematical operation performed against two other KPIs. For example, if you defined individual KPIs for a product's operating costs and revenue, you can define a composite KPI for the product's profit by subtracting the operating cost KPI from the revenue KPI. A composite KPI must be set up on individual KPIs that have the same collection interval, and dimensionality. Also, composite KPIs should be set up only on two “Sum” calculation or “Average” calculation KPIs. Although it is possible to set up composite KPIs involving a combination of “Sum” or “Average” calculation KPIs, the resulting calculations may be incorrect.

Note: When you define an individual or composite KPI for which applicable data has already been collected, the KPI instances created for that KPI are automatically populated with that existing data. This enables you to determine a KPI's effectiveness without having to wait for new data to be collected. See "[KPI Back Population](#)" on page 172 for more information.

There are two sections on the Add/Edit KPI page that enable you to define or update the characteristics of a KPI. These sections are **KPI Information** and **KPI Definition**. These sections and their related fields are explained below.

- **KPI Information:** This section enables you to assign a name to the KPI and define a metric (including an appropriate unit of measure, a calculation type appropriate for data aggregation and a data collection interval). In addition, you can define a naming template which helps you to identify data on graphs and tables throughout Optimize. Finally, in this section you also set Empty Data Set Behavior (EDSB) for the KPI.
- **KPI Definition:** The fields in this section enable you to define the KPI type (individual or composite) and to associate a fact with a *dimension* (a means for breaking out your data into slices), such as revenue by region or queue length by host. This creates a KPI Instance. In addition, you associate event maps and attributes with the KPI and you define a hierarchy.

Because of the inherent complexities involved in creating KPIs, you typically move back and forth between these two sections when defining a new KPI.

The fundamental factor to consider when creating a KPI is the event map selection. The event map you select defines the available dimensions, measures, naming templates, and other variables for the KPI. So, when creating a KPI it is important that you are familiar with the event maps configured on your system and know which one is appropriate for the KPI that you wish to create. For information about creating and editing event maps, refer to "[Mapping Events](#)" on page 150.

When creating a KPI, you must define Empty Data Set behavior (EDSB) for that KPI. EDSB behavior defines how the system handles intervals where no data is collected, and EDSB settings can have implications that ripple throughout the system. For an individual KPI with a calculation type of Sum or Count, the Empty Data Set Behavior can be defined as either "No Reading Provided" or "Value is Set To Zero." For all other calculation types (Average, Last Value, and State), the EDSB is set to "No Reading Provided" and cannot be edited.

After the first KPI is defined on a given measure, all subsequent individual and composite KPIs on that measure must have the same EDSB setting. The EDSB setting for subsequently defined KPIs is defaulted to the setting specified for the first KPI defined on the measure and is not editable.

The following table lists EDSB settings and summarizes their effects on calculations and data displays.

EDSB Setting	Description
No reading provided	Applies only to Sum KPI instances. If applicable, when calculations are performed, zero readings are counted and factored into calculations and are displayed on graphs and tables. Null readings are not factored into calculations and are not displayed on graphs and tables.
Value is set to Zero	Returns zero when zero or null readings are provided. The zero/null readings are factored into calculations, and they are displayed on graphs and tables; so they can potentially have a significant effect on values displayed throughout the system.

Also when you create an individual KPI, you can construct a naming template to use for that KPI using the **Naming Template** field. A naming template enables you to customize the name displayed for each KPI instance. By default, a KPI instance name is composed of the base KPI name plus the dimension hierarchy description. This naming structure is unique but can result in a lengthy display name that is difficult to interpret. Constructing an appropriate naming template enables you to customize the display name in a manner that suits your needs.

The event map and hierarchy you select for the KPI you are creating determine the variables available in the **Naming Template** field. Variables are constructed using the following format:

```
[$[DimensionName].[AttributeName]]
```

The naming template can contain any descriptive text in combination with one or more variables you select. When creating a naming template, you should select variables that make the derived name distinctive for each KPI instance. When working with composite KPIs, the variables available for use in the **Naming Template** field are those that are common across both KPIs specified.

Adding Individual KPIs

To add or edit an individual KPI

1. In My webMethods, select: **Navigate > Applications > Administration > Analytics > KPIs > Business Data** (Optimize for Process only) or **Navigate > Applications > Administration > Analytics > KPIs > System Data**
2. Click **Add KPI**.
Optimize displays the Add/Edit KPI page.
3. Ensure that the **Type** field displays the appropriate type of KPI (System or Business). System KPIs monitor intrinsic system information, and Business KPIs monitor process data.

4. On the **KPI Information** panel, do the following:
 - a. In the **Name** field, type a name for the KPI.
 - b. For the moment, disregard the **Unit of Measure** and **Calculation** fields. They are populated automatically when you make a selection in the **Fact** field below.
 - c. In the **Collection Interval** drop-down list, specify how often Optimize should collect data for this KPI. The default is five minutes. The minimum collection interval is one minute, and the maximum collection interval is eight hours.
5. On the **KPI Definition** panel, do the following:
 - a. From the **Definition Type** list, select **Individual**.
 - b. From the **Event Mapping** list, select an appropriate event map for this KPI.
 - c. From the **Measure** list, select a measure to associate with the KPI. One or more measures associated with the event map you selected in the previous step appear in the **Measure** list. The appropriate selections appear in the **Unit of Measure (UOM)** and **Calculation** text boxes in the **KPI Information** panel after you have selected a measure.
 - d. From the **Hierarchy** list, select a dimension hierarchy, or select **[No Hierarchy]**. Optimize displays only those dimension hierarchies that include one or more dimension attributes associated with the event map you selected in the **Event Mapping** list.
 - e. From the **Dimension** list, select the dimension to associate with the KPI. Optimize displays only those dimensions that are associated with the KPI hierarchy you selected in the previous step.
 - f. From the **Attribute** list, select an attribute associated with the dimension you selected in the previous step.
6. If you are creating a Sum KPI and it is the first KPI on the specified measure, ensure that the **Empty Data Set Behavior** field displays the appropriate selection for the behavior you want.
7. Go back to the **KPI Information** panel and ensure that the **Unit of Measure (UOM)** and **Calculation** text boxes are populated appropriately.
8. Also in the **KPI Information** panel, enter or select the appropriate dimension variables for the **Naming Template** field, if applicable to your system. Dimension variables enable you to ensure that dimension instances are displayed with meaningful names on various data displays. Click the icon to view a list of dimension variables for the selected event map and process.

Tip: Click the magnifying glass icon to see a sample KPI Instance name using the specified naming template.

9. Click **Save**.

Adding Composite KPIs

Observe the following rules when working with Composite KPIs. Note that not all of these rules are enforced through the user interface, so users must be aware of and observe them manually. Failure to observe these rules may result in system problems and/or invalid data.

- Only individual KPIs with calculation types of SUM-SUM and AVERAGE-AVERAGE are supported for use in Composite KPIs.
- EDSB for composite KPIs is inherited from the underlying KPIs if the underlying KPIs use the same setting. If the underlying KPIs use different settings, the EDSB for the composite KPI defaults to “Value is set to Zero”.
- Individual KPIs used in a Composite KPI must have the same dimensionality (including hierarchy).
- Individual KPIs used in a Composite KPI must all have the same collection interval.

To add or edit a composite KPI

1. In My webMethods select: **Navigate > Applications > Administration > Analytics > KPIs > Business Data** (Optimize for Process only) or **Navigate > Applications > Administration > Analytics > KPIs > System Data**
2. Click **Add KPI**.
Optimize displays the Add/Edit KPI page.
3. In the **KPI Information** panel, type a name for the KPI in the **Name** field.
4. In the **KPI Definition** panel, do the following:
 - a. From the **Definition Type** list, select **Composite**.
 - b. From the first **KPI** list, select a KPI. In the **KPI Information** panel, the **Collection Interval** field reflects the collection interval for this KPI.
 - c. From the **Operator** list, select the mathematical operation you want Optimize to perform on the two KPIs. Valid operators are **Add**, **Subtract**, **Multiply**, and **Divide**.
 - d. From the second **KPI** list, select a KPI. This list displays all eligible KPIs based on the first KPI selected.

Note: The collection interval of the second KPI must match that of the first KPI. Individual KPIs with the same collection interval, event map, and dimensionality are pre-populated, offering a list of KPIs that can be combined with the first KPI selected.
5. In the **KPI Information** panel, do the following:
 - a. In the **Unit of Measure** field, type the unit of measure that Optimize should use to evaluate data associated with this KPI (for example, Dollars or Bytes). If you would like the unit of measure to be automatically formatted when the KPI

results are displayed, use one of the following keywords (keywords are not case sensitive):

<u>Unit of Measure</u>	<u>Keyword(s)</u>	<u>Display</u>
Dollars	"Dollars" or "\$"	\$<value to two decimals>
Thousands of Dollars	"\$1000"	\$<value>, 000.00
Percentage	"%" or "Percent"	<value to two decimals> %
Time in Milliseconds	"MS", "Millis", or "Milliseconds"	<dd> Days <hh:mm:ss.mmm> Hours

- Also in the **KPI Information** panel, enter or select the appropriate dimension variables for the **Naming Template** field, if applicable to your system. Dimension variables enable you to ensure that dimension instances are displayed with meaningful names on various data displays. Click the icon to view a list of dimension variables for the selected event map and process. Also, type the appropriate free text to create a name that is meaningful for you.

Tip: Click the magnifying glass icon to see an sample KPI Instance name using the specified naming template.

- Click **Save**.

To view the data collected for a KPI, you can click  or search for the name of the KPI in the **Search** panel on the Business Data page (Optimize for Process only) or the System Data page. For multi-dimensional KPIs (as with the KPI for the three revenue regions described in the paragraph above), navigation will be to the KPI Summary page, where all of the KPI instances associated with the KPI are displayed. To view data for a specific instance, click a KPI icon to go to the KPI Instance Detail page.

For single-dimension KPIs, clicking the monitor icon or the name of the KPI activates the KPI Instance Detail page.

For more information on the KPI Instance Detail page, see "Using the KPI Instance Detail Page" in *webMethods Optimize User's Guide*.

KPI Back Population

If you create a new KPI or modify an existing one, Optimize invokes a back-population feature that automatically generates the appropriate KPI instances and back-populates baseline statistics from any existing data. In this case, back-populating means that the relevant data displays are re-calculated in light of the new KPI(s). During back population, data is re-parsed according to the new KPI specifications, but the fundamental contents of the underlying data set are not altered by this feature.

Note that back-population occurs only if there is data relevant to the KPI; if there is no relevant data, back-population does not occur. KPI back population is managed by the Optimize Job Manager and can be monitored through the Job Manager interface.

Customers can use the KPI back-population feature to update their data displays in a couple of scenarios as described below.

- If you want to recast your existing data in light of new or altered KPIs, you can make the desired KPI changes and Optimize will automatically invoke the back-population feature to recast your statistics. This means that both historical and current data is re-evaluated in light of the new or updated KPIs.
- If you have added back-dated events to your Optimize data set, or if you have injected legacy data from another data set into your Optimize data set, you can delete and recreate your existing KPIs to view graphs and statistics in light of the newly added data. Deleting and recreating existing KPIs is the same as creating new KPIs from an Optimize perspective.

When you create a new KPI or update an existing KPI, Optimize does the following:

- Displays all KPI Instances for that KPI on the Overview and Summary pages.
- Displays all KPI Instance detail graphs for all KPI instances.
- Displays baseline statistics (historical data) on KPI Instance Detail graphs for each KPI instance.
- Displays all diagnosis (trending up/down, sloping up/down, above/below mean, etc.) icons for the KPI.

Note: If you have a large data set, KPI back-population can take some time to complete. Be aware of this whenever you make a change that initiates back population.

When you create a new KPI, Optimize displays a message indicating that back-population is being initiated, if applicable. You can check the Job Manager Audit page to confirm that back-population has been completed. To view the Job Manager Audit page, click **Analytics > Job Management > Job Audit** in My webMethods.

Intelligent Links

Intelligent links provide the ability to configure dynamic links to Web pages inside or outside My webMethods and to display those links in appropriate pages based on the data. Data containing intelligent links displays as blue underlined text in the Problem Event Detail, KPI Instance Detail, Process Instance Detail, or Step Instance Detail pages. Clicking an intelligent link opens the designated URL with any configured parameters in a new browser window.

Intelligent links are managed from the Intelligent Link Definitions page, where you can view available links, add a new link and test it, and edit or delete an existing link.

To display the list of intelligent links:

In My webMethods: **Navigate > Applications > Administration > Analytics > KPIs > Intelligent Links**

All currently available intelligent links are displayed on the Intelligent Link Definition page.

Adding Intelligent Links

To add an intelligent link:

1. In My webMethods: **Navigate > Applications > Administration > Analytics > KPIs > Intelligent Links**

2. Click **Add Link Definition**.

The Add Intelligent Link Definition page is displayed.

3. Add the following information for your intelligent link. An asterisk (*) indicates a required field.

Field	Description
* Name	Name that will appear in the Intelligent Link Definitions list. No more than 80 alphanumeric characters (no punctuation).
Description	Description of the purpose of the intelligent link. No more than 255 alphanumeric characters (no punctuation).
* Base URL	URL for the intelligent link. Include only static parameters in this URL. Use the URL Parameters field (see below) for dynamic parameters.
URL Parameters	To define optional dynamic parameters for this intelligent link, in the URL Parameters panel, follow these instructions: <ol style="list-style-type: none"> a. Click Add Parameter. b. A blank line is displayed in the URL Parameters list. A new line is added to the URL Parameters list each time you click Add Parameter. c. In the Name text box, specify a name for the parameter. No more than 80 alphanumeric characters (no punctuation). Parameters that are not named are deleted when the intelligent link is saved.

Field	Description
	<ul style="list-style-type: none"> d. In the Description text box, provide an optional description of the parameter. Description text can be helpful when assigning an intelligent link parameter to a dimension or an event map because the text is displayed whenever you position the mouse pointer over a parameter name. e. To delete a parameter you have added, select the check box for that parameter and click Delete.
4.	<p>Validate the intelligent link by completing the following steps:</p> <ul style="list-style-type: none"> a. Click Test. The Test Intelligent Link pop-up window is displayed. b. Enter any parameter values you want to test with the URL in the URL Parameters panel. Position the mouse pointer over a parameter name to view the parameter description. c. Click Test. The URL target is displayed in the default browser. If the desired result is not displayed, you can do either of the following: <ul style="list-style-type: none"> ■ Change the parameter values. ■ Close the Test Intelligent Link pop-up window, and edit to adjust the link URL and parameters in the Intelligent Link Definition page. d. Click Done to close the Test Intelligent Link pop-up window and return to the Add/Edit Intelligent Link page. e. Make any necessary changes to the intelligent link, and do one of the following: <ul style="list-style-type: none"> ■ Click Save to save your changes, and click Test to test again. ■ Click Cancel to close the page without saving.
5.	<p>Click Save to save your intelligent link definition, or click Cancel to close the page without saving.</p>

Editing Intelligent Links

To edit an intelligent link:

1. On the Intelligent Link Definitions page, do one of the following:
 - Click the name of the intelligent link you want to edit.
 - Click  **Edit**.
The Edit Intelligent Link Definition page is displayed.

2. Make any changes to the link, test the edited link, and click **Save**. To close the page without saving, click **Cancel**.

Deleting Intelligent Links

To delete an intelligent link:

1. On the Intelligent Link Definitions page, select the check box beside the intelligent link you want to delete.
2. Click **Delete**.

Subscribing to webMethods Events

The Optimize Analytic Engine has the capability to consume subscribed events. You can subscribe to specific event types and use the business data of the received events to construct event maps and KPIs.

The event types you can subscribe to correspond to the list of XSD schemata currently available in the Event Type Store. The Event Type Store provides a central location where predefined and user-defined event types are stored. By default, it is installed under *Software AG_directory/common/EventTypeStore*. For more information about the Event Type Store, see the PDF publication *webMethods Event Processing Help*.

Note: If your Optimize system operates in a EDA enabled environment using NERV, then you must configure the NERV JMS Provider setting in the profile for the Platform Manager (SPM) that is running on the same machine as your Analytic Engine according to the instructions in the "Modifying Transport Layer Configuration" section of the *Implementing Event-Driven Architecture with Software AG Products* guide. See "[Configuring EDA/NERV Settings for Optimize](#)" on page 94 for more information. This configuration is important for all systems in order in order to support event subscription correctly, but it is especially critical for systems that use a Universal Messaging server that is configured remotely from the Optimize installation.

Filtering the List of Subscribed Event Types

The **Event Subscription** page contains the list of available event types in your Event Type Store. You can filter that list or modify the number of displayed items per page.

To filter the list of displayed event types

1. In My webMethods: **Navigate > Applications > Administration > Analytics > KPIs > Event Subscription**.
2. On the **Event Subscription** page, do one of the following:
 - Enter the name of the Event Type into the **Filter** field and click **Go** to filter the event types available on the page.

You can filter both by event name or by namespace.

- Leave the **Filter** field blank and click **Go** to display all available event types.

Note: By default, event types are displayed in batches of 30 items per page. If you want to modify that number, use the **Items per page** property in the upper right corner of the page to set a different number, and click **Go**.

Optimize displays the list of event types. You can subscribe to any of the event types from this page to receive events and use the data to create event maps and KPIs.

For more information about creating event maps, see ["Mapping Events" on page 150](#).

For more information about defining KPIs, see ["Defining KPIs" on page 167](#).

Subscribing to Event Types

The **Event Subscription** page displays a list of event types you can subscribe to.

To subscribe to events

1. In My webMethods: **Navigate > Applications > Administration > Analytics > KPIs > Event Subscription**.
2. On the **Event Subscription** page, select the check box beside the event type you want to subscribe to, and clear the check box beside the event type you want to unsubscribe from.
3. Click **Save** to save the event subscription definition, or click **Delete** to cancel your choice and continue to use the event subscription definition you last made.

Granting User Access to KPIs

You can control access to KPIs in Optimize. The **Data-Level Security** panel on the Edit Role page enables you to specify access parameters for users and groups by role.

Granting a role access to a KPI allows anyone in that role to view KPI data in Optimize. If a user or group belongs to more than one role, the user or group will have access to the KPI data as long as access is granted to at least one of the roles.

After a role is created, access must be assigned. By default, roles do not have access to KPIs. For users and groups to view KPI data, data-level security must be enabled. If data-level security has been enabled, users must be assigned access to view KPIs. For instructions for enabling data-level security, see ["Enabling Data-Level Security" on page 75](#).

To open the data level security tab for KPIs

1. In My webMethods: **Navigate > Applications > Administration > User Administration > Roles**.

Optimize displays the Roles page containing a list of roles. You can add new roles and edit or delete existing roles from this page. For information about adding, editing, and deleting roles, see *Working with My webMethods*.

2. Do one of the following:
 - Enter the name of the role into the **Search Keyword** field and click **Go**.
 - Leave the **Search Keyword** field blank and click **Go**.
3. Click the **ROLE NAME**, or click the **Edit** button (✎) to the right of the role name. Optimize displays the Edit Role page for the role.
4. In the **Data Level Security** tab, click **KPI**. Optimize displays the KPI page for data level security, which contains a list of all accessible KPIs.
5. From this page, you can do any of the following:

To...	Do this...
Grant access to a KPI	Click Add KPI and continue to " Granting Access to KPIs " on page 178 below.
Remove a KPI from the list	Select the KPI and then click Delete .
Export the list of attributes to a comma-delimited text file	Click Export Table. For more information about exporting table data, see <i>Administering My webMethods Server</i> .

Granting Access to KPIs

When granting access to a KPI, you grant access to a role. Users assigned to that role are subsequently granted access. You grant access to a role from the Edit Role page.

To grant access to KPI

1. On the **Data Level Security** tab, click **KPI**.
2. Click the **Add KPI** button. Optimize displays the **Add KPI Privileges** page.
3. Do one of the following:
 - Type the name of the KPI into the **Search Keyword** field and click **Go**.
 - Leave the **Search Keyword** field blank and click **Go**.
4. Select the check box next to the KPI for which you want to grant access, and then click **Select** at the top of the KPI tab. Optimize displays the Edit Role page with the KPI displayed at the bottom of the **Data Level Security** tab.

5. To specify one or more instances for a KPI, click the **Edit** button (✎) beside the name of the KPI. Optimize displays the Edit KPI Privileges page, which contains the **KPI Instances** tab. To display a list of KPI instances available for the KPI, select the **Restricted By KPI Instances** check box in the **KPI Instances** tab.
6. Select the check box next to the KPI instances for which you want to grant access, and then click **Select**.

Important: If no KPI instances are selected in the list, then the role can access all instances for the KPI.

Deleting Corrupted KPI Definitions

In rare cases, when you have initiated a deletion process of one or more KPI definitions using the Business Data or System Data pages and that process did not complete successfully, you might have some corrupted assets in the database. You can use the KPI definition deletion script to delete those corrupted KPI definitions and related KPI instances. The KPI definition deletion script uses the database configuration specified during the installation of the Analytic Engine, and cleans up any corrupted KPI assets found in the database.

Note: Please note that any KPI instances related to the corrupted KPI definitions will be deleted as well. Also, keep in mind that the KPI definition deletion script is not aware of cache, so it must not be used in a Terracotta cluster environment.

To run the KPI definition deletion script

1. Enter one of the following at the command line:
 - (Windows) `Optimize_directory\analysis\bin\cleanupKPIs.bat`
 - (UNIX) `Optimize_directory/analysis/bin/cleanupKPIs.sh`

Important: UNIX users who receive a Java security exception when they run the `cleanupKPIs.sh` script must modify the script as follows: Delete the following line (`-Djava.security.manager=''\`) at this location in the script:

```
$JAVA_HOME/bin/java\ $GLUE_OPTS\  
com.webmethods.optimize.cleanup.KPICleanup$
```

Extracting Optimize Assets from My webMethods Server

The webMethods product suite offers tools for storing, building, and deploying Optimize assets (such as custom trees, data filters, dimensions, event maps, dimension hierarchies, intelligent links, process configurations, and rules) from one environment to another using a common composite repository system, thus enabling users to share business data throughout the entire organization.

You can extract Optimize assets from My webMethods Server, save them to your file system, check them in to a version control system, and later deploy them to another Optimize Analytic Engine.

To extract Optimize assets

1. In My webMethods: **Navigate > Administration > Analytics > Data Management > Asset Extraction**

My webMethods displays the **Asset Extraction** page.

2. Expand the Optimize root and select the assets you want to extract.
3. Click **Select Dependencies** to select all assets dependent on the ones you selected in step 2.

This ensures that all necessary assets are extracted in one archive.

4. Click **Export Assets** to export all selected Optimize assets to your file system.
5. In the dialog that appears, specify a project name for the exported assets. In case you do not provide a valid name, an error message appears.

The assets are exported in a folder with the specified project name. The project contains an archive with separate .xml files that you can save and later deploy to another instance of Optimize Analytic Engine. The archive includes an .acdl file in the project's directory. The .acdl file has the same name as the project and contains a description of the exported assets and their dependencies.

For more information about deploying the extracted assets to other instances of Optimize Analytic Engine, see *webMethods Deployer User's Guide*.

6 Configuring and Using the Infrastructure Data Collector

■ Overview	182
■ Monitoring Infrastructure Data Collector	182
■ Configuring Integration Servers and webMethods Brokers	182
■ Configuring the SNMP Data Collector	184
■ Changing Infrastructure Data Collector Service Name	190
■ Changing Infrastructure Data Collector Ports	190
■ Defining HTTPS Connectors	192
■ Configuring Infrastructure Data Collector for SSL	193

Overview

The SNMP data collector works with the webMethodsInfrastructure Data Collector to monitor non-webMethods devices using the Monitoring API. The Monitoring API passes operational data and monitoring model data to the Analytic Engine. The Monitoring API is designed as a programmatic interface to the systems monitoring functionality of a business integration platform. It communicates over JMS, utilizing a JMS queue, to provide secure and reliable communications between Optimize and Infrastructure Data Collector.

The Infrastructure Data Collector lets you identify webMethods component resources that you want to monitor and analyze. For information on discovering and monitoring resources, see ["Discovering Assets and Monitoring Components" on page 195](#)

For information on starting and stopping the Infrastructure Data Collector, see ["Starting and Stopping Infrastructure Data Collector " on page 46.](#)

For information on configuring the Infrastructure Data Collector, see *Configuring BAM*.

Information on using SNMP data collector, and configuring webMethods Broker and webMethods Integration Server to use Infrastructure Data Collector, follows.

Monitoring Infrastructure Data Collector

You can monitor the status of a remote Infrastructure Data Collector using the following web page.

`http://<hostname>:<port>/infradc`

Configuring Integration Servers and webMethods Brokers

The current version of Infrastructure Data Collector can monitor the current version of webMethods Integration Servers and webMethods Brokers without any special configuration. In order to monitor older versions of Integration Server, ensure that you are using the latest available service pack and apply the latest Infrastructure Data Collector fixes to Integration Server.

Configuring IS Packages

Once you have configured your Integration Server(s) for monitoring with Infrastructure Data Collector, you can configure parameters for the following packages:

WmISExtDC

The following parameters can be configured. Changes take place immediately; you do not need to restart Integration Server for the changes to take effect.

To configure parameters for WmiSExtDC

1. Open Integration Server Administrator for your Integration Server.
2. In Integration Server Administrator: **Packages > Management > WmiSExtDC**.
3. Click **Browse services in WmiSExtDC**, click the name of the parameter you want to configure, and click **Test <parameter>**.
4. Enter the value in the appropriate field.

<u>Parameter</u>	<u>Description</u>
------------------	--------------------

setMaxNotificationQueueLength	
-------------------------------	--

	Changes the default Integration Server Notification queue length. The default queue length value is 6000.
--	---

setOnlySendMonitoredObjectNotification	
--	--

	Changes how notifications will be sent to Infrastructure Data Collector.
--	--

	Choose from the following values
--	----------------------------------

<u>Value:</u>	<u>Definition:</u>
True	Only notifications sent by monitored components will be sent to the Infrastructure Data Collector
False	All Integration Server notifications will be sent to the Infrastructure Data Collector

	Default = True
--	-----------------------

sendNotification	
------------------	--

	Used for defining your own notifications. Choose from the following key value pairs:
--	--

<u>Key</u>	<u>Value</u>
messages	String[] msgsg

Parameter	Description
notificationType	"FAILURE", "LOGMESSAGE", "OPERATIONAL", "STOPPED"
severity	"CRITICAL", "MAJOR", "MINOR", "OK", "WARNING", "INFO"
correctiveMessage	String msg

WmARTExtDC

The polling interval for adapter model changes and model status can be configured.

To configure the polling interval in WmARTExtDC

1. Open Integration Server Administrator for your Integration Server.
2. In Integration Server Administrator: **Packages > Management > WmARTExtDC**.
3. Click Browse services for WmARTExtDC, click setAdapterPollInterval, and click **Test setAdapterPollInterval** to configure how often Adapter monitoring polls ART for changes.
4. Enter a value in minutes in the field.

Configuring the SNMP Data Collector

The SNMP data collector for Infrastructure Data Collector monitors non-webMethods devices such as bridges, hubs, routers, or network servers that are accessible with Simple Network Management Protocol (SNMP) agents.

Configuring the SNMP data collector is a three-step process:

1. Identify the devices to monitor.

Configure the SNMP data collector configuration files to identify which devices that you want to monitor. The SNMP data collector configuration files are defined by an MIB configuration file.
2. Discover the SNMP data collector configuration files using the asset discovery process as described in ["Discovering Assets and Monitoring Components" on page 195](#).
3. Select the SNMP objects to monitor in My webMethods.

Monitoring begins when you select the SNMP objects and attributes from the My webMethods Monitored Components page.

Creating and Defining the MIB Configuration File

A *Management Information Base (MIB)* is a hierarchical database of information that defines properties of objects that are associated with an SNMP-managed device. It contains elements that map to your existing MIB hierarchy. The SNMP data collector uses this configuration file to identify the devices to monitor. The SNMP data collector also uses the MIB file to create the hierarchical structure that determines how Optimize displays information about the devices on the Monitored Components page.

The webMethods SNMP MIB configuration file has the following basic structure:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE MIBConfig SYSTEM
"
Software AG_directory
/profiles/InfraDC/configuration/com.softwareag.infracd
.wmsnmpdc/config/MIBConf
ig.dtd">
<MIBConfig>
<ConfigType>
<MIBNode/>
<Key>
<.../>
</Key>
<Display>
<.../>
</Display>
<Monitor>
<.../>
</Monitor>
</ConfigType>
</MIBConfig>
```

Element	Description
<!DOCTYPE>	Reference to the document type definition file that defines the MIB configuration.
<MIBConfig>	The root element. A <MIBConfig> element contains one or more <ConfigType> elements.
<ConfigType>	A configuration type for an SNMP MIB object or sequence table node.

Note: The webMethods SNMP data collector cannot process a SEQUENCE as a parent type node. The Agent does not expect a SEQUENCE type to have any other ConfigType entries as a child. That is, a node that is a SEQUENCE cannot contain other ConfigType entries, and if it does, an error will be generated. Instances of the node you specify for this element appear on the My webMethods Monitored Components page.

Element	Description										
<MIBNode>	The MIB node to which the configuration type maps.										
	<table border="1"> <thead> <tr> <th>Attribute</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>The name of the node.</td> </tr> <tr> <td>OID</td> <td>The node's object ID.</td> </tr> <tr> <td>Type</td> <td>The data type of the node. Valid types are INTEGER, STRING, OBJECT, SEQUENCE, NULL, IPADDRESS, PHYSADDRESS, COUNTER, GAUGE, and TIMETICKS.</td> </tr> <tr> <td>Unit</td> <td>For monitors, the unit associated with the monitor value in the System tree on the Analytics Overview page (for example, Byte). This attribute is required and can be any logical string. An empty string("") is not allowed. If this attribute is not provided, the default unit, Unit, is used.</td> </tr> </tbody> </table>	Attribute	Description	Name	The name of the node.	OID	The node's object ID.	Type	The data type of the node. Valid types are INTEGER, STRING, OBJECT, SEQUENCE, NULL, IPADDRESS, PHYSADDRESS, COUNTER, GAUGE, and TIMETICKS.	Unit	For monitors, the unit associated with the monitor value in the System tree on the Analytics Overview page (for example, Byte). This attribute is required and can be any logical string. An empty string("") is not allowed. If this attribute is not provided, the default unit, Unit, is used.
Attribute	Description										
Name	The name of the node.										
OID	The node's object ID.										
Type	The data type of the node. Valid types are INTEGER, STRING, OBJECT, SEQUENCE, NULL, IPADDRESS, PHYSADDRESS, COUNTER, GAUGE, and TIMETICKS.										
Unit	For monitors, the unit associated with the monitor value in the System tree on the Analytics Overview page (for example, Byte). This attribute is required and can be any logical string. An empty string("") is not allowed. If this attribute is not provided, the default unit, Unit, is used.										
<Key>	A unique string to identify an instance of the object node. This string consists of fields constructed from one or more of the following elements:										
	<table border="1"> <thead> <tr> <th>Element</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><MIBNode></td> <td>The value of a child MIB node.</td> </tr> <tr> <td><FixedNode></td> <td>The specified text string.</td> </tr> </tbody> </table>	Element	Description	<MIBNode>	The value of a child MIB node.	<FixedNode>	The specified text string.				
Element	Description										
<MIBNode>	The value of a child MIB node.										
<FixedNode>	The specified text string.										
	<p>Note: When using a ConfigType of SEQUENCE, the MIBnode is expected to contain index information. The index information is used to discover node instances, which appear on the My webMethods Monitored Components page.</p>										
<Display>	(Optional) A single MIB variable to appear in the Monitored Components page.										
<Monitor>	(Optional) The child node to monitor. Optimize displays the node you specify for this element in the following places:										

Element	Description
	<ul style="list-style-type: none"> ■ In the Monitored Components page. ■ As a monitor in the system tree on the Analytics Overview page if you select it for monitoring. <p>The <Monitor> element consists of one or more <MIBNode> elements. The <MIBNode> elements must be of Integer type and they must represent child nodes of the configuration type with which the <Display> element is associated.</p>

The following example illustrates a sample MIBConfig.xml properties file and the hierarchy it creates in the system tree on the Analytics Overview page:

Table 1. Example sample_MIBConf.xml file

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE MIBConfig SYSTEM
"file:///C:/<INSTALLDIR>/profiles/InfraDC/configuration/com.softwareag.infradc
.wmsnmpdc/config/MIBConfig.dtd#">
<MIBConfig>
<ConfigType Name="HostGroup">
  <MIBNode Name="host" OID="1.3.6.1.2.1.25" Type="OBJECT"/>
  <Key>
    <FixedNode Value="HostInformation"/>
  </Key>
  <ConfigType Name="HostSystem">
    <MIBNode Name="hrSystem" OID="1.3.6.1.2.1.25.1" Type="OBJECT"/>
    <Key>
      <FixedNode Value="hrSystem"/>
    </Key>
    <Monitor>
      <MIBNode Name="hrSystemUptime" OID="1.3.6.1.2.1.25.1.1"
Type="INTEGER" Unit="Second"/>
      <MIBNode Name="hrSystemNumUsers" OID="1.3.6.1.2.1.25.1.5"
Type="INTEGER" Unit="User Session"/>
      <MIBNode Name="hrSystemProcesses" OID="1.3.6.1.2.1.25.1.6"
Type="INTEGER" Unit="Process"/>
      <MIBNode Name="hrSystemMaxProcesses" OID="1.3.6.1.2.1.25.1.7"
Type="INTEGER" Unit="Process"/>
    </Monitor>
  </ConfigType>
  <ConfigType Name="HostStorage">
    <MIBNode Name="hrStorage" OID="1.3.6.1.2.1.25.2" Type="OBJECT"/>
    <Key>
      <FixedNode Value="hrStorage"/>
    </Key>
    <Monitor>
      <MIBNode Name="hrMemorySize" OID="1.3.6.1.2.1.25.2.2"
Type="INTEGER" Unit="KBytes"/>
    </Monitor>
    <ConfigType Name="HostStorageTable">
      <MIBNode Name="hrStorageTable" OID="1.3.6.1.2.1.25.2.3"
Type="SEQUENCE"/>
      <Key>
        <FixedNode Value="hrStorageTable"/>
        <MIBNode Name="hrStorageIndex" OID="1.3.6.1.2.1.25.2.3.1.1"
```

```
Type="INTEGER"/>
    </Key>
    <Display>
        <MIBNode Name="hrStorageDescr" OID="1.3.6.1.2.1.25.2.3.1.3"
Type="STRING"/>
    </Display>
    <Monitor>
        <MIBNode Name="hrStorageAllocationUnits"
OID="1.3.6.1.2.1.25.2.3.1.4" Type="INTEGER" Unit="Byte"/>
        <MIBNode Name="hrStorageSize" OID="1.3.6.1.2.1.25.2.3.1.5"
Type="INTEGER" Unit="AllocationUnits"/>
        <MIBNode Name="hrStorageUsed" OID="1.3.6.1.2.1.25.2.3.1.6"
Type="INTEGER" Unit="AllocationUnits"/>
        <MIBNode Name="hrStorageAllocationFailures"
OID="1.3.6.1.2.1.25.2.3.1.7" Type="INTEGER" Unit="Failure"/>
    </Monitor>
</ConfigType>
</ConfigType>
</ConfigType>
<ConfigType Name="NetworkInterfaces">
    <MIBNode Name="interfaces" OID="1.3.6.1.2.1.2" Type="OBJECT"/>
    <Key>
        <FixedNode Value="NetworkInterfaces"/>
    </Key>
    <Monitor>
        <MIBNode Name="ifNumber" OID="1.3.6.1.2.1.2.1" Type="INTEGER"/>
    </Monitor>
</ConfigType>
</MIBConfig>
```


At the top of the file, be sure to include a `<!DOCTYPE>` definition that specifies the `MIBConfig.dtd` file located in the `SoftwareAG-directory\profiles\InfraDC\configuration\com.softwareag.infracd.wmsnmpdc\config` directory.

Tip: Use the `sample_MIBConfig.xml` file in the `SoftwareAG-directory\InfrastructureDC\samples\snmp` directory as a guide.

4. Save the text file.

Changing Infrastructure Data Collector Service Name

If you have installed Infrastructure Data Collector as a service, you can change the service name using the following procedure.

To change the Infrastructure Data Collector service name:

1. Open a command prompt and remove previous service by running the following command:

```
install_dir\profiles\InfraDC\bin\service.bat -remove
```

2. Open the `SoftwareAG-directory\profiles\InfraDC\bin\setenv.bat` file and change the following lines:

```
SET SERVICE_NAME=saginfrastructuredatacollector82_1
SET WINDOW_TITLE=Software AG Optimize Infrastructure Data Collector (1)
```

3. Save the `setenv.bat` file and close it.
4. Open the `SoftwareAG-directory/profiles/InfraDC/configuration/wrapper.conf` file and edit the appropriate lines to read as follows:

```
wrapper.ntservice.name=saginfrastructuredatacollector82_1
wrapper.ntservice.displayname=Software AG Optimize Infrastructure Data
Collector (1)
wrapper.ntservice.description=Software AG webMethods Optimize Infrastructure
Data Collector 8.2
```

5. Save the `wrapper.conf` file and close it.
6. Register the new service name by running the following command:

```
Software AG_directory\profiles\InfraDC\bin\service.bat -install.
```

Changing Infrastructure Data Collector Ports

In some cases, you may need to change the Infrastructure Data Collector default port assignments. Users may wish to change ports in the case of conflicts, if multiple Infrastructure Data Collectors are installed on the same machine, or to accommodate specific configuration issues. Infrastructure Data Collector uses two ports: one serves as the primary port to invoke web services, and the other is the configuration port used for deploying central configuration metadata. To change the port assignment, you must edit the Infrastructure Data Collector `config.ini` file.

Changing the Infrastructure Data Collector Primary Port

To change the Infrastructure Data Collector Primary Port

1. Open the <install dir>\profiles\InfraDC\configuration\config.ini file.
2. Edit the following property `org.eclipse.equinox.http.tomcat.http.port` to point to an unused port.

The edited property should have the following

format:`org.eclipse.equinox.http.tomcat.http.port=<Primary Port>`

3. Save your changes to the file.
4. Open the <install dir>\profiles\InfraDC\configuration\tomcat\conf\server.xml file, and edit the primary port as shown in the following example:

```
<Connector acceptCount="100"
  connectionTimeout="20000"
  description="CTP HTTP port"
  disableUploadTimeout="true"
  enableLookups="false"
  maxHttpHeaderSize="8192"
  maxSpareThreads="75"
  maxThreads="150"
  minSpareThreads="25"
  port=<Primary Port>
  redirectPort="6740"/>
```

5. Save your changes to the file.

Changing the Infrastructure Data Collector Configuration Port

To change the Infrastructure Data Collector Configuration port

1. Open the <install dir>\profiles\InfraDC\configuration\config.ini file.
2. Edit the following property `com.softwareag.infradc.ccs.port"=16000` to point to an unused port.

The edited property should have the following

format:`com.softwareag.infradc.ccs.port=<Config Port>`

3. Save your changes to the file.
4. Open the <install dir>\profiles\InfraDC\configuration\tomcat\conf\server.xml file, and edit the config connector port as shown in the following example:

```
<Connector port="<Config Port>
  maxHttpHeaderSize="8192"
  maxThreads="150"
  minSpareThreads="25"
  maxSpareThreads="75"
  enableLookups="false"
  acceptCount="100"
  connectionTimeout="20000"
  disableUploadTimeout="true"
```

```
redirectPort="8084"/>
```

5. Save your changes to the file.

Note: Configuration port changes must also be reflected in the Central Configuration settings for the Infrastructure Data Collection logical server on the Map Endpoints tab for deploying a configuration. See *Configuring BAM* for more information.

Defining HTTPS Connectors

If you want to use secure web service or browser connections to communicate with Infrastructure Data Collector, you must set up server and client side HTTPS connectors.

Server-side Configuration

To define an HTTPS connector on the server side

1. Obtain an appropriate server certificate and place it in the desired folder.
2. Define the location of the server certificate with the parameter `keystoreFile` (replace the default value).
3. Set the CN of the certificate to be identical to the URL of the server, without the "https://".

For example, for a server under `https://MyWebServer:8443/`, the CN is "MyWebServer".

4. Set the keystore password parameter (`keystorePass`).

Infrastructure Data Collector supports both types of Java keystores:

- `KeystoreType="JKS"` (default)
- `KeystoreType="PKCS12"` (PKCS#12 keystore)

Client-side Configuration

To accept an HTTPS connection on the client side

Import the server certificate into your browser trust store (or in the case of a PKI, preferably the CA certificate that has issued the server certificate).

Note: If you are accessing UDDI from a Java client using Oracle JSSE, you must also set a truststore via `Djavax.net.ssl.trustStorePassword=<your_trust_store_here>`. Alternatively, if you have not imported the server certificate and the browser asks if you trust the certificate, select **Yes** at the prompt if you trust the certificate.

Configuring Infrastructure Data Collector for SSL

This section describes how to manually enable the Tomcat SSL connector which secures Infrastructure Data Collector for SSL. Note that to secure an entire system, you must also secure all other Optimize components as well.

To manually enable the Tomcat SSL Connector

1. Open `/profiles/InfraDC/configuration/config.ini` file and add the following property:

```
org.eclipse.equinox.http.tomcat.https.port=<port>
```

2. Append the property to the `"com.softwareag.platform.port.properties.list"` property as shown:

```
com.softwareag.platform.port.properties.list=org.eclipse.equinox.http.tomcat.http.port,com.softwareag.platform.startup.jmx.rmi.agent.port,org.eclipse.equinox.http.tomcat.https.port
```

3. Open the `/profiles/InfraDC/configuration/tomcat/conf/server.xml` file, and add the following `<Connector>`:

```
<Connector description="Default CTP HTTPS Connector"
port="${org.eclipse.equinox.http.tomcat.https.port}"
maxHttpHeaderSize="8192" maxThreads="150" minSpareThreads="25"
maxSpareThreads="75" enableLookups="false" disableUploadTimeout="true"
acceptCount="100" scheme="https" secure="true" SSLEnabled="true"
keystoreFile="<path to keystore>" keystoreType="JKS" keystorePass="<keystore
password>" clientAuth="false" sslProtocol="TLS" algorithm="SunX509" />
```

Make sure to type the KeyStore file and password as specified in the Tomcat documentation. Refer to the Tomcat documentation (available from the Tamino website for more information.)

4. Open the `/profiles/InfraDC/workspace/wsstack/repository/conf/axis2.xml` file and find the place where `<transportReceiver>` commands are configured (there should be just one transport receiver for HTTP). Add the HTTPS transport receiver as shown in the following example:

```
<transportReceiver name="http"
class="org.apache.axis2.osgi.tx.HttpListener"> <parameter
name="portProperty">org.eclipse.equinox.
http.tomcat.http.port</parameter></transportReceiver><transportReceiver
name="https" class="org.apache.axis2.osgi.tx.HttpsListener"> <parameter
name="portProperty">org.eclipse.equinox.
http.tomcat.https.port</parameter></transportReceiver>
```

5. Save the file and close it.

7 Discovering Assets and Monitoring Components

■ Overview	196
■ Discovering Assets	199
■ Monitoring Components	225
■ Using the Monitor and Suspend Services	230

Overview

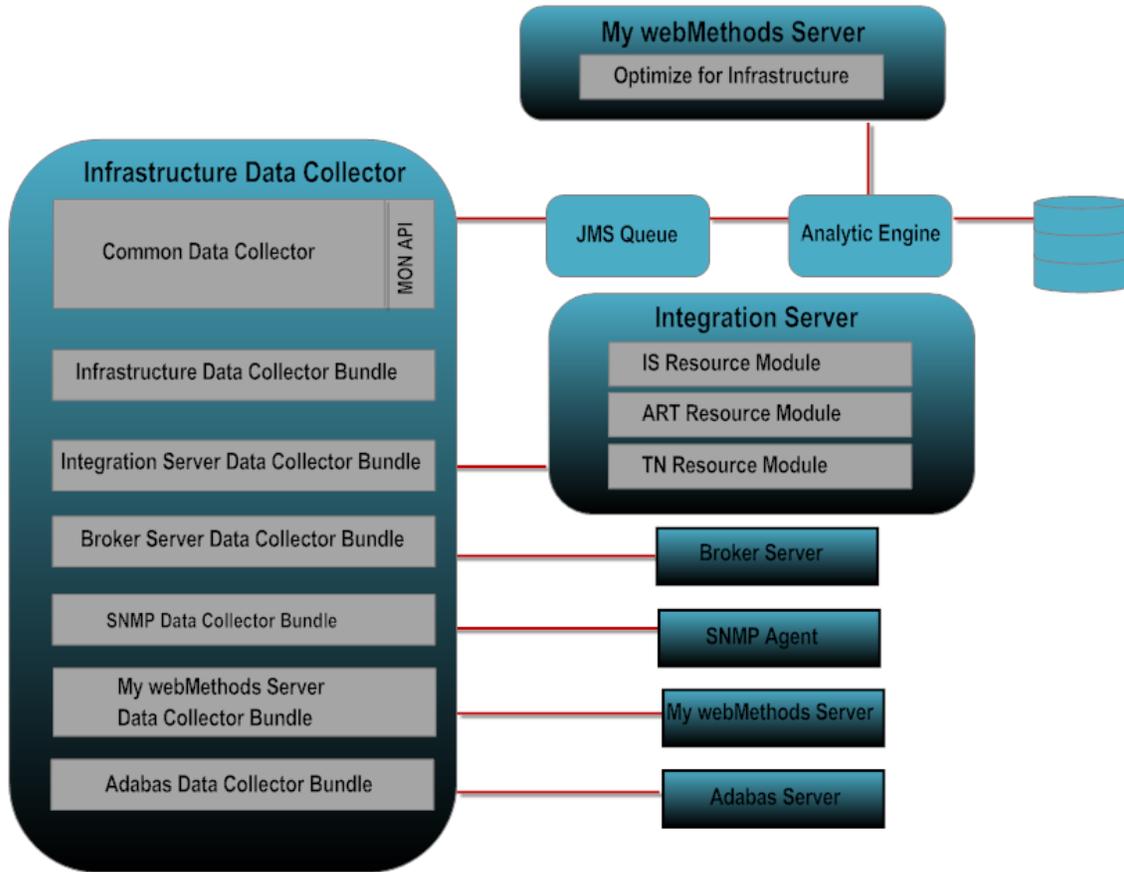
Optimize enables you to discover and monitor Software AG component resources on your network. You can discover and monitor Integration Servers (including Integration Server clusters), Broker Servers, My webMethods Servers, Universal Messaging servers, Terracotta Management Consoles, and SNMP components to your system configuration as well as EntireX, ApplinX, Adabas, Adabas SOA Gateway, Com-plete, Natural and Natural Ajax components. These components are generally referred to as assets. By discovering and monitoring these assets, you can enhance the monitoring capabilities of your system.

This chapter provides information about how Infrastructure Data Collector and Optimize work together as well as instructions and information for discovering, adding, monitoring, and deleting the described assets.

Interaction between Infrastructure Data Collector and Optimize

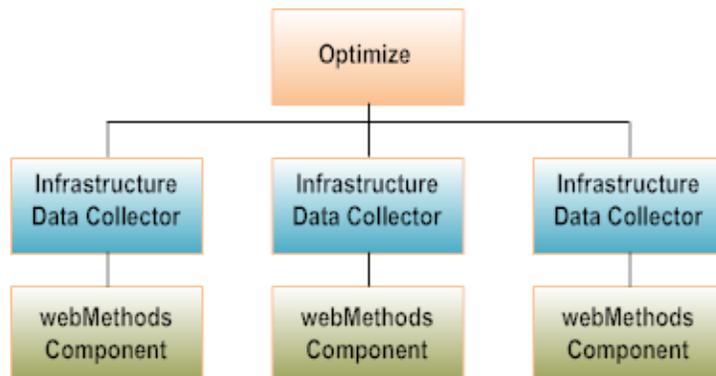
In an Optimize installation, Infrastructure Data Collector uses the Monitoring API to pass operational data and monitoring model data to the Analytic Engine. The Monitoring API serves as a programmatic interface to the systems monitoring functionality of a business integration platform. It communicates over the Java Message Service (JMS), utilizing a JMS queue, to provide secure and reliable communications between Optimize and Infrastructure Data Collector.

The diagram below provides an overview of Optimize and Infrastructure Data Collector. As shown, Infrastructure Data Collector connects to the Analytic Engine in Optimize through JMS.



Scaling Your Architecture

Each Infrastructure Data Collector supports up to 15,000 components or 30,000 KPI instances whichever is reached first. One Infrastructure Data Collector connected to one Analytic Engine should be sufficient for most needs, but you can scale your architecture by deploying multiple Infrastructure Data Collectors. The following diagram illustrates the suggested architecture for scaling Optimize.



To further increase scalability, My webMethods can support multiple installations of Optimize. For information about hardware requirements depending on the number of

objects you are monitoring, see *Installing webMethods and Intelligent Business Operations Products*.

Infrastructure Data Collector

Infrastructure Data Collector enables you to identify Software AG component resources to monitor and analyze. Using Infrastructure Data Collector, you can monitor the following:

- webMethods Universal Messaging servers

When you start monitoring a Universal Messaging Server, Infrastructure Data Collector makes related items available for monitoring. Such items include Universal Messaging channels, queues, and datagroups that are running on the Universal Messaging Server.

- webMethods Broker Servers.

When you start monitoring a Broker Server, Infrastructure Data Collector makes related items available for monitoring. Such items include webMethods Broker clients, document types, and gateways and territories that are running on the Broker Server.

- Integration Servers.

When you start monitoring an Integration Server, Infrastructure Data Collector makes related items available for monitoring. Such items include the Integration Server clusters (if enabled) and all Integration Server ports; Trading Networks Server, if you have that component installed; and packages that are running on the Integration Server. To lessen the load on Infrastructure Data Collector, you can choose to monitor only certain services on an Integration Server.

If clustering is enabled on an Integration Server, the logical cluster and all nodes of the cluster are shown on the Discovered Assets and Monitored Components pages. That is, there are three or more components displayed on the Monitored Components page: the Integration Server asset, the IS Cluster asset, and one or more IS Cluster Server assets. For Integration Server clusters, there are several intrinsic KPI available that enable you to monitor cluster components. Refer to "[Monitored KPI Definitions](#)" on [page 349](#) for more information about KPIs that are available for Integration Server cluster monitoring.

- webMethods Adapters.

Infrastructure Data Collector lets you monitor adapters installed on Integration Servers. When you start monitoring an Integration Server, Infrastructure Data Collector makes all adapter components that are installed on the Integration Server available for monitoring.

- SNMP Agents.

The Infrastructure Data Collector lets you specify how often SNMP data is collected and delivered to the analytic engine for processing. By selecting which monitored

components and KPIs you want to monitor on the Optimize Monitored Components page, you can instruct Infrastructure Data Collector to collect monitored SNMP data.

- webMethodsEntireX, webMethodsApplnX, Adabas, Adabas SOA Gateway, Complete, Natural, and Natural Ajax components.

Infrastructure Data Collector enables you to discover and monitor these components.

- My webMethods Servers.

Infrastructure Data Collector enables you to discover and monitor My webMethods Servers in your environment.

- Terracotta Servers and Server Arrays

Infrastructure Data Collector enables you to discover a Terracotta Server, Server Arrays, and Ehcache.

- Presto Servers

Infrastructure Data Collector enables you to discover and monitor Presto Servers in your environment.

Optimize

Optimize enables you to access the resource information maintained by Infrastructure Data Collector. You use Optimize to monitor the status of individual monitored components as well as the overall status of your system. You can use Optimize to do the following:

- **Set alerts based on a monitored component's KPIs.** Optimize allows you to define a rule that specifies conditions under which a component resource becomes problematic and requires user attention. For example, you can create an object error count rule that triggers when a Integration Server Service Error Count is greater than or equal to 1.

Optimize alerts you through the My webMethods user interface and through any other alert mechanisms that you configure, such as e-mail or SNMP trap.

- **Retrieve notifications generated by a monitored component.** Optimize enables you to access error messages for components that are monitored by Infrastructure Data Collector. You can view resource errors in My webMethods from the KPI Instance Detail page for the monitored component error message.

Discovering Assets

Assets is a generic term for network-attached resources in your environment to which you can connect and which may require periodic administrative attention. Integration Servers, Integration Server clusters, My webMethods Servers, webMethods Universal Messaging, webMethods Brokers, as well as Adabas, Natural, and SNMP components that exist in your network are examples of assets. Note that Broker Servers and SNMP components generally have child components that are typically the specific targets for monitoring.

The Analytical section of My webMethods provides several pages that enable you to discover, configure, and monitor assets. While these pages appear similar, they have specific interactions and restrictions that you must be aware of when discovering and configuring assets.

Before you can monitor resources in your environment, you must discover them. Discovery is the process of locating and establishing communication with resources and their child components. Entries for resources on the Discovery page are called *discoveries*. The Discovery page enables you to specify the configuration used for initially conducting these discoveries. Valid discoveries become assets and are automatically propagated to the Assets page.

Once an asset has been discovered, its connection configuration information is displayed in the Discovered Assets table on the Assets page. Any changes or refinements to the asset's connection credentials (connection parameters) must be made on the Assets page, not on the Discovery page. Also these changes are not reflected on the Discovery page.

After assets are configured, they can be selected and configured for monitoring using the Monitored Components page. See "[Monitoring Components](#)" on page 225 for more information.

Note: It is recommended that only one Infrastructure Data Collector be used to monitor any single Integration Server instance. Optimize events are stored in memory on the monitored Optimize. Using more than one Infrastructure Data Collector on different JMS queues to monitor the same Integration Server instance could result in unexpected KPI behavior.

The Discovery and Asset page functionality is designed to accommodate complexities inherent in typical component configurations. The following example illustrates how and why these pages function as they do:

You know there is a webMethods Broker on a Broker Server somewhere in your network that you want to monitor. Also, the webMethods Broker you want to monitor has a different SSL setting than the parent Broker Server.

1. Using the Discovery page, create a discovery to locate the webMethods Broker Server and all its associated components.
2. Using the Assets page, locate the webMethods Broker you want to monitor in the Discovered Assets table, and modify its SSL settings via the Keystore connection parameters so that you can monitor the webMethods Broker.

The Discovery Page

The Discovery page shows information about all existing discoveries and enables you to create new discoveries. Components that can be discovered include Integration Servers, Terracotta Management Consoles, webMethods Universal Messaging Servers, Broker Servers, My webMethods Servers, and SNMP components as well as EntireX, ApplinX, Adabas, Adabas SOA Gateway, Complete, Natural, and Natural Ajax components. Note that this page does not show the parent SNMP Config file for SNMP components or child webMethods Broker components underneath the parent Broker Servers. When

you discover a component using Asset Discovery on the Discovery page, that discovery runs automatically and the discovered asset is listed on the Assets page. The Assets page shows the child webMethods Brokers and parent SNMP Config files in addition to Integration Servers, webMethods Broker Servers, and SNMP agents.

You may need to edit connection information for a discovery if the parameters used to create that discovery have changed. If you change information about a discovery on the Discovery page, the changes are not reflected on the Assets page unless you update the settings on the Discovery page and rerun the discovery.

The webMethods Broker auto-discovery feature automatically detects when new components have been created on a previously discovered Broker Server asset. For example, auto discovery can find a new webMethods Broker that has been created on a Broker Server. When webMethods Broker auto-discovery is enabled, it runs automatically on all available Broker Servers to discover newly added components. Note that Broker Servers must still be manually discovered, even if webMethods Broker auto-discovery is enabled.

webMethods Broker auto-discovery is disabled by default, because it can have a negative impact on system performance in some configurations. You must use the Central Configuration feature of My webMethods to enable auto-discovery. See the “Defining Logical Server Subcomponents for the Infrastructure Data Collector” section of the Managing webMethods Environments chapter of *Configuring BAM* for more information.

Note: If the Infrastructure Data Collector is down and a new webMethods Broker is added or deleted while auto-discovery is enabled, the new webMethods Broker is discovered or removed when Infrastructure Data Collector comes up. But if a document type/custom adapters/territory/gateway is changed while Infrastructure Data Collector is down then it will not be detected when Infrastructure Data Collector comes back up. In this case, you must manually re-discover the Broker Server to update these components.

Note: The Optimize Analytic Engine does not function with IPv6, so Optimize for Infrastructure must be run on IPv4 while it monitors resources running on IPv6. As a workaround, users need to discover Software AG applications running on IPv6 using the appropriate host name rather than the IPv6 address.

To view the Discovery page, navigate the following path in My webMethods:

Navigate > Applications > Administration > Analytics > Infrastructure Components > Discovery.

The primary Discovery page component is the Assets to Discover table. This table displays a No Discoveries Found message until you create a discovery. Two buttons titled **Add Assets** and **Add Network** enable you to create discoveries.

This table displays the following information about discovered assets:

Column	Description
NAME	<p>The IP address or host name, and the port number for the discovery.</p> <p>Note: Discovered Integration Server assets always consist of the Integration Server's fully qualified DNS name and port number, even if an IP address or range was specified at the Discovery page.</p>
TYPE	The component type (SNMP, Broker Server, Integration Server, My webMethods Server, Adabas, ApplinX, EntireX, Adabas SOA Gateway, Com-plete, Natural or Natural Ajax) of the asset for which you are searching, and the type of discovery (Asset or Network).
DATA COLLECTOR	The network address or host name of the data collector used for this discovery.
STATUS	<p>State of the discovery as listed below.</p> <ul style="list-style-type: none"> ■ Configured: The discovery has been configured but has not been run. This status applies only to a Network search. ■ Running: Indicates a network discovery in progress. ■ Completed: Indicates a discovery that completed without errors. ■ Requested: Indicates that a discovery request has been made. The discovery may be in progress or the Infrastructure Data Collector may not have received the discovery request. ■ Cancelled: Indicates that there has been a request to stop a network discovery. ■ Failed: Indicates that a discovery failed. If status is Failed, position the mouse pointer over the status to display information about the cause of the failure.
LAST ACTION	Time the last action indicated in the STATUS column was performed.
ACTIONS	Selected actions you can perform on the discovery.

In addition, you can do the following on the Discovery page:

- Click the **Add Asset** button to discover an asset for which you have specific addressing information. You can discover Integration Servers, Broker Servers, My webMethods Servers, Terracotta Servers and Server Arrays and SNMP components

as well as ApplinX, EntireX, Adabas, Adabas SOA Gateway, Complete, Natural, or Natural Ajax components using this button. See ["Add Asset" on page 204](#) for more information.

- Click the **Add Network** button to discover network assets using a range of IP addresses. You can discover Integration Servers and Broker Servers using this method. See ["Add Network" on page 212](#) for more information.
- Click the **Delete** button to remove selected discoveries. Note that this button is inactive until an asset is selected. Also, you cannot remove a discovery in progress. Deleting an asset does not affect previously discovered or monitored assets. For instance, deleting an SNMP Discovery does not delete discovered SNMP assets or associated event data.
- Click the **Refresh** button to update the discovery display. Click this button after adding or discovering any assets to ensure that the information displayed is current.
- Click the **Export Table** button to view the contents of the Discovery page as a table.
- Click a discovery name in the **NAME** column to view and/or edit the connection parameters for that asset.
- To view information about an asset with a failed status, position the mouse pointer over the word **Failed** in the **STATUS** column.
- Click ▾ beside a column heading to sort the display by the information in the column.
- Click an icon in the **ACTIONS** column to manipulate a discovery. See the following table for information about specific icons and the actions they facilitate:

If you want to...	Do this...
Cancel a network discovery that is in progress.	Click  in the ACTIONS column beside the network discovery you want to cancel. (This icon does not appear for asset discoveries or for network discoveries that are no longer in progress.)
Run a previously created asset discovery.	Click  in the ACTIONS column beside the discovery you want to run. (You cannot run a discovery that is already in progress.)
Change the connection parameters of a previously created asset discovery.	Click  in the ACTIONS column beside the discovery you want to edit. (You cannot edit an asset discovery that is in progress.)

Add Asset

The **Add Asset** button on the Discovery page enables you to add asset discoveries to your environment when you have all of the information required to connect to those assets. You must have specific information about the host and port(s) used by the target asset to complete this procedure. Also, if you are adding an SNMP component, you must have the name and location of the SNMP Configuration file for the component. Use the following procedure and tables to add assets using the **Add Asset** button.

Any changes or refinements to the asset's connection credentials (connection parameters) must be made on the Assets page, not on the Discovery page. Also, these changes are not reflected on the Discovery page.

To add a discovery using the Add Asset button

1. In My webMethods: **Navigate > Applications > Administration > Analytics > Infrastructure Components > Discovery**

The Discovery page is displayed.

2. Click **Add Asset**.

The Add Asset Discovery dialog is displayed.

3. Click the Down arrow to the right of the **Asset Type** field, and select the desired discovery type.

Note: If you add an asset to your system or start an existing asset that was previously not running, you may need to refresh the Add Asset Discovery dialog in order for that asset to be displayed in the **Asset Type** list.

The fields on the Add Asset Discovery dialog change to match those required by the selected discovery type.

4. Complete the fields on the Add Asset Discovery dialog.

Refer to the tables that follow this procedure for information about the optional and required fields for each valid discovery type.

5. Click **OK** to add the specified asset, or click **Cancel** if you want to cancel the procedure without adding the discovery.
6. Click **Refresh** to update the information on the Discovery page.

To discover an Integration Server, enter the following information on the Add Asset Discovery dialog (required fields are identified with *) Note that if clustering is enabled on the Integration Server, the Discovery page is the same as for a standalone Integration Server. Following discovery of an Integration Server cluster, however, the logical cluster and individual cluster components are displayed on the Assets page, and you can monitor cluster components on the Monitored Components page.

Field	Description
*Asset Type	Select the type of asset (Integration Server) to discover.
*Data Collector	Select the Infrastructure Data Collector to use for the discovery. <div style="border: 1px solid gray; padding: 5px; background-color: #f0f0f0;"> Tip: Only one Infrastructure Data Collector can be used to monitor the same Integration Server instance on different JMS queues, so if you use multiple Infrastructure Data Collectors, be careful not to point more than one to the same Integration Server. </div>
*Host	Enter the host name or IP address for the discovery. The name must be unique.
*Port	Enter the port number for the discovery.
SSL	(Optional) Select if an SSL connection is required.
KeyStore File	This field appears only if SSL is selected. Specify the location of the KeyStore file.
*Username	User ID needed to log in to the asset. Required if displayed but not displayed if a KeyStore file is listed.
*Password	Password needed to log in to the asset. Required if displayed but not displayed if a KeyStore file is listed.
KeyStore Password	This field appears only when KeyStore file information is supplied. Specify the password for the KeyStore file.
KeyStore Alias	This field appears only when KeyStore file information is supplied. Specify the alternate name for the KeyStore file.
Include System Packages/ Services	(Optional) Select if you want to include webMethods-supplied packages and services in the data returned from the Integration Server. Disabled by default.

To discover a webMethods Universal Messaging Server, enter the following information on the Add Asset Discovery dialog (required fields are identified with *). Note that there is a unique method of monitoring a Universal Messaging server with SSL, which is invoked via the appropriate protocol selection.

Field	Description
*Asset Type	Select the type of asset (Universal Messaging) to discover.
*Data Collector	Select the Infrastructure Data Collector to use for the discovery.
*Host	Enter the host name or IP address for the discovery. The name must be unique.
*Server Port	Enter the server port number for the discovery.
Username	User ID needed to log in to the asset.
Password	Password needed to log in to the asset.
*Protocol	Protocol used to connect to the Universal Messaging Server being discovered. Supported protocols are nsp, nsps, nhp, and nhps. The default is nsp. To monitor a Universal Messaging server with SSL, select nsps as the protocol. Note that when using SSL, the Universal Messaging Server must be configured to support SSL and the Infrastructure Data Collector must be configured to use the appropriate key and trust stores using the JNDI Configuration setting on the Configure Servers tab of the My webMethods Define Environments page. See the <i>Configuring BAM</i> guide for more information about JNDI Configuration. Also, the InfraDC config.ini file must point to the appropriate key and trust stores. If you need to edit the config.ini file to point to key and trust stores other than the defaults, see " Editing the Config.ini file for Universal Messaging SSL Configuration " on page 216 for more information.
Client Authentication	Select the client authentication type appropriate for the Universal Messaging Server being discovered. Currently, the only available option is "None".

To discover a Broker Server, enter the following information on the Add Asset Discovery dialog (required fields are identified with *). Note that some of the fields displayed vary according to the Broker Server's client authentication level ("None", "Basic", or "Certificate"):

Field	Description
*Asset Type	Select the type of asset (Broker Server) to discover.
*Data Collector	Select the Infrastructure Data Collector to use for the discovery.
*Host	Enter the host name or IP address for the discovery. The name must be unique.
*Monitor Port	Enter the Broker Server monitor port number. Default = 6850.
*Server Port	Enter the webMethods Broker Server port number. Default= 6849.
Client Authentication	Select the client authentication type appropriate for the Broker Server being discovered. Available options are "None", "Basic" and "Certificate". Note that additional fields may be displayed on the Discovery dialog based on your selection as explained below.
Encryption	Select this check box if the Broker Server being discovered requires encryption. Note that TrustStore File and TrustStore Type fields are displayed and are required if encryption is selected.
Username	User ID required to log in to the asset. Displayed only when client authentication is "Basic."
Password	Password required to log in to the asset. Displayed only when client authentication is "Basic."
KeyStore File	Type the name and path of the Broker Server KeyStore file. This field is displayed, and required, only when client authorization is "Certificate".
KeyStore Type	Type the KeyStore file type. This field is displayed, and required, only when client authorization is "Certificate".
KeyStore Password	Type the password for the Broker Server certificate file. This field is displayed, and required, only when client authorization is "Certificate".

Field	Description
TrustStore File	Type the TrustStore filename if required to identify the owner or issuer of the certificate file. This field is displayed only when either the client authorization is "Certificate" or the Encryption box is checked. If the Encryption box is checked, the field is required.
TrustStore Type	Type the TrustStore file type. This field is displayed only when either the client authorization is "Certificate" or the Encryption box is checked. If the Encryption box is checked, the field is required.

To discover a My webMethods Server, enter the following information on the Add Asset Discovery dialog (required fields are identified with *):

Field	Description
*Asset Type	Select the type of asset (My webMethods Server) to discover.
*Data Collector	Select the Infrastructure Data Collector to use for the discovery.
*Host	Enter the host name or IP address for the discovery. The name must be unique.
*Port	Enter the port number for the discovery.
*Username	User ID needed to log in to the asset.
*Password	Password needed to log in to the asset.

To discover an SNMP component, enter the following information on the Add Asset Discovery dialog (required fields are identified with *):

Field	Description
*Asset Type	Select the type of asset (SNMP) to discover.
*Data Collector	Select the Infrastructure Data Collector to use for the discovery.

Tip: Only one Infrastructure Data Collector can be used to monitor the same SNMP component on different JMS queues, so if you use multiple Infrastructure Data

Field	Description
	Collectors, be careful not to point more than one to the same SNMP component.
*SNMP Version	Select the appropriate SNMP version (SNMPV1 or SNMP V2c) to use when connecting to the SNMP agent for the discovery.
*SNMP Host	Enter the host name or IP address for the discovery. The name must be unique.
*Port	Enter the port number for the discovery. The default is 161.
*Community Password	Password needed to log in to the asset. This is the community password used for secure connections to the SNMP Agent.
*SNMP Config	Location of the SNMP MIB Configuration file on the Infrastructure Data Collector host for this SNMP component discovery. <p>Note: Note that a single MIB filename cannot be used in two different locations on the same host; however, the same filename (with exactly the same path) can be used to discover SNMP agents located on different hosts.</p>

To discover an ApplinX, EntireX, Adabas SOA Gateway, Com-plete, or Natural Ajax component, enter the following information on the Add Asset Discovery dialog (required fields are identified with *):

Field	Description
*Asset Type	Select the type of asset (ApplinX , Adabas SOA Gateway , Com-plete , EntireX , or Natural Ajax) to discover.
*Data Collector	Select the Infrastructure Data Collector to use for the discovery. <p>Tip: Only one Infrastructure Data Collector can be used to monitor the same component on different JMS queues, so if you use multiple Infrastructure Data Collectors, be careful not to point more than one to the same component.</p>
*Host	Enter the host name or IP address for the discovery. The name must be unique.
*Port	Enter the ApplinX or EntireX port number. For ApplinX components the default is 2380, and for EntireX components

Field	Description
	the default is 1971. The default port for Adabas SOA Gateway components is 56100, and the default for Complete components and Natural Ajax components is 80.
SSL	(Optional for some components) Select if an SSL connection is required.
*Username	User ID needed to log in to the asset.
*Password	Password needed to log in to the asset.
KeyStore File	(Optional) Applies only to EntireX components. Enter the name and path of the EntireX KeyStore file.

To discover an Adabas or Natural component, enter the following information on the Add Asset Discovery dialog (required fields are identified with *):

Field	Description
*Asset Type	Select the type of asset (Adabas or Natural) to discover.
*Data Collector	Select the Infrastructure Data Collector to use for the discovery. <div style="background-color: #f0f0f0; padding: 5px; margin-top: 5px;"> <p>Tip: Only one Infrastructure Data Collector can be used to monitor the same component on different JMS queues, so if you use multiple Infrastructure Data Collectors, be careful not to point more than one to the same component.</p> </div>
*Host	Enter the host name or IP address for the EntireX Broker host. The name must be unique.
*Port	Enter the EntireX Broker port number. The default for both Adabas and Natural components is 1971.
*RPC Server	Enter the host name of the Natural RPC server.
*Username	User ID needed to log in to the asset.
*Password	Password needed to log in to the asset.
KeyStore File	(Optional) Enter the name and path of the EntireX Broker KeyStore file.

Field	Description
RPC User	User ID needed for secure connections to the Natural RPC server.
RPC Password	Password needed for secure connections to the Natural RPC server.

To discover Terracotta Servers and Server Arrays, enter the following information on the Add Asset Discovery dialog (required fields are identified with *). Following discovery of an Terracotta Server Array, however, the logical server and individual array components are displayed on the Assets page, and you can monitor individual components on the Monitored Components page.

Note: You can monitor only Terracotta 4.1 and newer servers.

Field	Description
*Asset Type	Select the type of asset (Terracotta) to discover.
*Data Collector	Select the Infrastructure Data Collector to use for the discovery. <div data-bbox="511 1050 1377 1249" style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;"> <p>Tip: Only one Infrastructure Data Collector can be used to monitor the same Integration Server instance on different JMS queues, so if you use multiple Infrastructure Data Collectors, be careful not to point more than one to the same Integration Server.</p> </div>
*Host	Enter the host name or IP address for the Terracotta Management Console. The name must be unique.
Connection Type	Select <code>TMS</code> or <code>Direct</code> as the type of an existing connection. <div data-bbox="511 1449 1377 1753" style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;"> <p>Note: When you select the <code>TMS</code> connection type, in order to monitor Terracotta servers and server components, you must have an appropriately configured Terracotta Management Console. For information about configuring a Terracotta Management Console, see the Terracotta Management Console section of the Terracotta 4.1 documentation that is available from the Terracotta website.</p> </div>
*Port	Enter the port number for the applicable Terracotta Management Console.

Field	Description
*Group Name	(Available if Connection Type is set to TMS) Enter the name of an existing Terracotta Server Array group.
*TSA Name	(Available if Connection Type is set to Direct) Enter the name of the relevant Terracotta Server.
Username	User ID needed to log in to the asset.
Password	Password needed to log in to the asset.

To discover Presto Servers, enter the following information on the Add Asset Discovery dialog (required fields are identified with *):

Field	Description
*Asset Type	Select the type of asset (Presto) to discover.
*Data Collector	Select the Infrastructure Data Collector to use for the discovery. <div style="background-color: #f0f0f0; padding: 5px; margin-top: 5px;"> <p>Tip: Only one Infrastructure Data Collector can be used to monitor the same Integration Server instance on different JMS queues, so if you use multiple Infrastructure Data Collectors, be careful not to point more than one to the same Integration Server.</p> </div>
*Host	Enter the host name or IP address for the Presto Server. The name must be unique.
*Port	Enter the port number for the applicable Presto Server.
*Username	User ID needed to log in to the asset.
*Password	Password needed to log in to the asset.

Add Network

The **Add Network** button on the Discovery page enables you to add assets to your environment when you do not know the exact connection information for the desired assets on your network. You can locate and add Integration Server and Broker Server assets using a range of IP addresses. Use the following procedure and tables to add assets using the **Add Network** button.

Note: Only Integration Server and Broker Server components can be discovered using the **Add Network** button.

To add a discovery using the Add Network button

1. In My webMethods: **Navigate > Applications > Administration > Analytics > Infrastructure Components > Discovery**

The Discovery page is displayed.

2. Click **Add Network**.

The Add Network Discovery dialog is displayed.

3. Click the Down arrow to the right of the **Asset Type** field, and select the desired discovery type.

The fields on the Add Network Discovery dialog change to match those required by the selected discovery type.

4. Complete the fields on the Add Network Discovery dialog.

Refer to the tables that follow this procedure for information about the optional and required fields for each valid discovery type.

5. Click **OK** to add the specified asset, or click **Cancel** if you want to cancel the procedure without adding the discovery.
6. Click **Refresh** to update the information on the Discovery page.

Note: Network discovery sends UDP echo and NETBIOS Name Server query requests to each address within the specified address range. When a host responds to these requests, it is added to the list of servers. The UDP echo service (port 7) and NETBIOS Name Server (port 137) must be available on the remote hosts for them to be discovered through Network Search.

To discover an Integration Server, enter the following information on the Add Network Discovery dialog (required fields are identified with *):

Field	Description
*Asset Type	Select the type of asset (Integration Server) to discover.
*Discovery Name	Enter a name for this discovery.
*Data Collector	Select the Infrastructure Data Collector to use for the discovery.

Tip: Only one Infrastructure Data Collector can be used to monitor the same Integration Server instance on different JMS queues; so if you use multiple

Field	Description
	Infrastructure Data Collectors, be careful not to point more than one to the same Integration Server.
*Start IP	Enter the IP address at the beginning of the range to scan.
*End IP	Enter the IP address at the end of the range to scan.
*Port	Enter the port number for the discovery.
SSL	(Optional) Select if an SSL connection is required.
KeyStore File	This field appears only if SSL is selected. Specify the location of the KeyStore file.
*Username	User ID needed to log in to the asset. Required if displayed but not displayed if a KeyStore file is listed.
*Password	Password needed to log in to the asset. Required if displayed but not displayed if a KeyStore file is listed.
KeyStore Password	This field appears only when KeyStore file information is supplied. Specify the password for the KeyStore file.
KeyStore Alias	This field appears only when KeyStore file information is supplied. Specify the alternate name for the KeyStore file.
Include System Packages/Services	(Optional) Select if you want to include webMethods-supplied packages and services in the data returned from the Integration Server. Disabled by default.

To discover a Broker Server, enter the following information on the Add Network Discovery dialog (required fields are identified with *):

Field	Description
*Asset Type	Select the type of asset (Broker Server) to discover.
*Discovery Name	Enter a name for this discovery.
*Data Collector	Select the Infrastructure Data Collector to use for the discovery.

Field	Description
*Start IP	Enter the IP address at the beginning of the range to scan.
*End IP	Enter the IP address at the end of the range to scan.
*Monitor Port	Enter the webMethods Broker monitor port number.
*Server Port	Enter the Broker Server port number.
Client Authentication	Select the client authentication type appropriate for the Broker Server being discovered. Available options are "None", "Basic" and "Certificate". Note that additional fields may be displayed on the Discovery dialog based on your selection as explained below.
Encryption	Select this check box if the Broker Server being discovered requires encryption. Note that TrustStore File and TrustStore Type fields are displayed and are required if encryption is selected.
KeyStore File	Type the name and path of the Broker Server KeyStore file. This field is displayed, and required, only when client authorization is "Certificate".
KeyStore Type	Type. the KeyStore file type. This field is displayed, and required, only when client authorization is "Certificate".
KeyStore Password	Type the password for the Broker Server certificate file. This field is displayed, and required, only when client authorization is "Certificate".
TrustStore File	Type the TrustStore filename if required to identify the owner or issuer of the certificate file. This field is displayed only when either the client authorization is "Certificate" or the Encryption box is checked. If the Encryption box is checked, the field is required.
TrustStore Type	Type the TrustStore file type. This field is displayed only when either the client authorization is "Certificate" or the Encryption box is checked. If the Encryption box is checked, the field is required.

Editing the Config.ini file for Universal Messaging SSL Configuration

If you want to discover a Universal Messaging server that uses SSL, you must ensure that the config.ini file contains the correct location for the KeyStore and TrustStore certificates.

To update the TrustStore/KeyStore location in the config.ini file

1. Open the <installdir>/profiles/InfraDC/configuration/config.ini file using an appropriate text editor.
2. Locate and edit the TrustStore and KeyStore locations so that they are accurate for the Universal Messaging server being discovered.
3. Save the file and close it.
4. To implement your changes, restart the Infrastructure Data Collector.

The Assets Page

The Assets page enables you to view information about discovered assets and, in some cases, modify or refine their connection credentials. It also enables you to delete assets, subject to certain restrictions and limitations, particularly in regard to Integration Server clusters and SNMP assets. In addition, this page also enables you to suspend monitoring of selected components.

If webMethods Broker auto-discovery is enabled, brokers and related components are automatically updated for all Broker Servers, though only webMethods Brokers and Broker Servers are displayed on the Assets page.

To view the Assets page, navigate the following path in My webMethods: **Navigate > Applications > Administration > Analytics > Infrastructure Components > Assets.**

The primary component of the Assets page is the Discovered Assets table. This table displays all discovered SNMP assets, Integration Servers, Integration Server clusters, My webMethods Servers, and Broker Servers/webMethods Brokers arranged in a hierarchical order. For webMethods Broker assets the parent Broker Server is added as a node in the tree view of assets, and all related webMethods Broker components appear beneath that node. Similarly, for SNMP assets the parent SNMP file is added as a node in the tree view of assets, and all related SNMP components appear beneath that node. The name, asset type, data collector used, connection status, and when the asset information was last updated are listed.

If a cluster enabled Integration Server is discovered, the IS Cluster asset is added as a parent node in the tree view of assets and all IS Cluster Server members appear beneath that node. Note that there are some restrictions on suspending and deleting these components as discussed below.

The Discovered Assets table provides the following information about each asset:

Column	Description
NAME	Click a name to edit connection parameters for an asset. Click  to expand a parent asset to view its children.
TYPE	Asset type: Universal Messaging Server, Broker Server, Broker, SNMP Configuration, Integration Server, IS Cluster, or IS Cluster server, My webMethods Server, SNMP, SNMP Agent, Adabas, ApplinX, EntireX, Natural, Natural Ajax, Presto, or Terracotta.
DATA COLLECTOR	The name of the Infrastructure Data Collector associated with this asset. Note: If you have discovered the same Integration Server asset with two or more Infrastructure Data Collectors on different JMS queues, you must delete the asset here and then create a new discovery at the Discovery page that uses a single Infrastructure Data Collector.
ENABLED	The Enabled icon indicates the status of asset monitoring: <ul style="list-style-type: none"> ■ A green enabled icon means monitoring is enabled for all components of the asset. ■ A red suspended icon means monitoring is suspended for all components of the asset.
CONNECTION STATUS	Connection status (Accepted, Refused, or Connecting) of the asset. If the status is Refused , you can move your mouse over the status indicator to view a description of the associated error.
LAST UPDATED	The date/time when this asset was last updated.
ACTIONS	Click  to edit communication parameters for this asset.

You can do the following on the Assets page:

- Click the **Delete** button to remove selected assets. Note that the **Delete** button is inactive until at least one asset is selected. Deleting an Integration Server or Broker Server asset removes it from monitoring but does not remove the asset itself, nor does it remove historical data collected for the asset. Note that check boxes for individual webMethods Brokers are not available because they cannot be removed.

Note: In the case of IS Clusters, you can only delete at the IS Cluster asset level. Deleting an IS Cluster asset will delete all IS Cluster Server

assets associated with the specified IS Cluster. The IS Cluster and IS Cluster Server KPIs use a connection with any of the participating Integration Server to poll for KPI values and status, and thus at least one of the participating Integration Server must be discovered so that an Infrastructure Data Collector can establish a connection. When all Integration Servers that participate in an IS Cluster are deleted from the Assets page, the IS Cluster and IS Cluster Server KPIs will no longer be updated. Delete the IS Cluster asset from the Assets page if you no longer need to monitor this component.

Note also that users are not notified when an Integration Server leaves an IS Cluster. When an Integration Server instance leaves a cluster, Infrastructure Data Collector is not notified of the change, and the Integration Server remains on the Assets page. As a workaround, delete the Integration Server cluster asset from the Assets page and rediscover the IS Server. Then, re-select the component for monitoring.

Similarly, if KPI values are not being generated for an IS Cluster because the associated Integration Server asset has been suspended or deleted, then you should delete and re-discover the Integration Server asset associated with the cluster.

- Click the **Refresh** button to update connection status information for discovered assets.
- Click the **Export Table** button to view the contents of the Assets page as a table.
- See "[Deleting SNMP Component Assets](#)" on page 223 for special instructions for deleting SNMP component assets.
- Click ▾ beside a column heading to sort the display by the information in the column.
- Click  to expand the hierarchical tree to view all children, if applicable.
- Click the name of an asset, or click , in the **EDIT** column to edit the connection parameters used to communicate with the asset. See "[Editing Connection Parameters](#)" on page 218 for more information. Note that changes to these parameters are not reflected on the Discovery page.

Editing Connection Parameters

You can edit connection parameters for assets from the Assets page. These parameters are a subset of the connection information specified when creating a discovery, though changes to these parameters are not reflected on the Discovery page.

Note: Connection parameters for IS Cluster and IS Cluster Server asset types cannot be edited.

To edit the connection parameters for an asset

1. Click the name of the asset, or click .

The Edit Connection Parameters dialog appears.

2. You can edit the following information:

- For a Universal Messaging Server.

Parameter	Description
Data Collector	Specifies the Infrastructure Data Collector to be used by the Universal Messaging Server.
Host	Specifies the host machine for the Universal Messaging Server.
Server Port	Specifies the communication port for the Universal Messaging Server.
Username	User ID required to login on the Universal Messaging Server.
Password	Password for logging in to the Universal Messaging Server.
Protocol	Specifies the protocol for communicating with the Universal Messaging Server.
Client Authentication	Specifies the client authentication type appropriate for the Universal Messaging Server. Currently, the only available option is "None".

- For an Integration Server, My webMethods Server, Adabas SOA Gateway, or Complete asset. Note that not all parameters are applicable for all asset types. For instance, only the Username and Password can be edited for a My webMethods Server asset:

Parameter	Description
SSL	(Optional) Select if SSL is required.
*Username	User name for logging in to the Integration Server. Required if displayed but not displayed if a KeyStore file is listed.

Parameter	Description
*Password	Password for logging in to the Integration Server. Required if displayed but not displayed if a KeyStore file is listed.
KeyStore File	This field appears only if SSL is selected. Specify the location of the KeyStore file.
KeyStore Password	This field appears only when KeyStore file information is supplied. Specify the password for the KeyStore file.
KeyStore Alias	This field appears only when KeyStore file information is supplied. Specify the alternate name for the KeyStore file.

- For a webMethods Broker or Broker Server asset:

Parameter	Description
Client Authentication	Select the client authentication type appropriate for the Broker Server being discovered. Available options are "None", "Basic" and "Certificate". Note that additional fields may be displayed on the Edit Connection Parameters dialog based on your selection as explained below.
Username	User ID required to log in to the asset. Displayed only when client authentication is "Basic."
Password	Password required to log in to the asset. Displayed only when client authentication is "Basic."
KeyStore File	Type the name and path of the Broker Server KeyStore file. This field is displayed, and required, only when client authorization is "Certificate".
KeyStore Type	Type the KeyStore file type. This field is displayed, and required, only when client authorization is "Certificate".
KeyStore Password*	Type the password for the Broker Server certificate file. This field is displayed, and required, only when client authorization is "Certificate".

Parameter	Description
TrustStore File	Type the TrustStore filename if required to identify the owner or issuer of the certificate file. This field is displayed only when either the client authorization is "Certificate" or the Encryption box is checked. If the Encryption box is checked, the field is required.
TrustStore Type	Type the TrustStore file type. This field is displayed only when either the client authorization is "Certificate" or the Encryption box is checked. If the Encryption box is checked, the field is required.

- For an SNMP component asset:

Parameter	Description
Password	Password for the MIB file.

- For an Adabas or Natural asset:

Parameter	Description
Username*	User ID needed for secure connections to the EntireX Broker.
Password*	Password needed for secure connections to the EntireX Broker.
Keystore File	Name and path of the EntireX Broker KeyStore file.
RPC User	User ID needed for secure connections to the Natural RPC server.
RPC Password	Password needed for secure connections to the Natural RPC server.

- For an EntireX asset:

Parameter	Description
Username*	User ID needed for secure connections to the EntireX Broker.

Parameter	Description
Password*	Password needed for secure connections to the EntireX Broker.
KeyStore File	Name and path of the EntireX Broker KeyStore file.

- For an ApplinX or Natural Ajax asset:

Parameter	Description
SSL	(Applies only to Natural Ajax). (Optional) Select if SSL is required.
Username*	User ID needed for secure connections to the asset.
Password*	Password needed for secure connections to the asset.

- For a Terracotta asset:

Parameter	Description
*Data Collector	Select the Infrastructure Data Collector to use for the discovery.
	<div style="background-color: #f0f0f0; padding: 10px;"> <p>Tip: Only one Infrastructure Data Collector can be used to monitor the same Integration Server instance on different JMS queues, so if you use multiple Infrastructure Data Collectors, be careful not to point more than one to the same Integration Server.</p> </div>
*Host	Enter the host name or IP address for the Terracotta Management Console. The name must be unique.
Connection Type	Select <code>TMS</code> or <code>Direct</code> as the type of an existing connection.
*Port	Enter the port number for the applicable Terracotta Management Console.
*Group Name	(Available if Connection Type is set to <code>TMS</code>) Enter the name of an existing Terracotta Server Array group.

Parameter	Description
*TSA Name	(Available if Connection Type is set to <code>Direct</code>) Enter the name of the relevant Terracotta Server.
Username	User ID needed to log in to the asset.
Password	Password needed to log in to the asset.

- For a Presto Server.

Parameter	Description
*Data Collector	Select the Infrastructure Data Collector to use for the discovery.
	<p>Tip: Only one Infrastructure Data Collector can be used to monitor the same Integration Server instance on different JMS queues, so if you use multiple Infrastructure Data Collectors, be careful not to point more than one to the same Integration Server.</p>
*Host	Enter the host name or IP address for the Presto Server. The name must be unique.
*Port	Enter the port number for the applicable Presto Server.
*Username	User ID needed to log in to the asset.
*Password	Password needed to log in to the asset.

Deleting SNMP Component Assets

There are several options for deleting SNMP component assets, and each has significant implications that you should be aware of before proceeding. You can delete only selected SNMP assets, or you can delete an entire SNMP Configuration node. The following list explains the results and implications of each option.

- If you delete one or more specific SNMP assets under an SNMP Configuration node, the specified assets are removed from the Assets page, associated components are hidden on the Monitored Components page, and associated KPIs are hidden on the Analytics Overview page. Note that the KPIs are only flagged as inactive; if you rediscover the assets, the associated KPI instances are reactivated.

If you delete an SNMP configuration, all underlying SNMP assets are selected automatically. Deleting an SNMP configuration is handled the same as deleting

specific SNMP assets, meaning the configuration is removed from the Assets page, and associated components are hidden on the Monitored Components page.

This option is appropriate if you have made no changes to the MIB file and want to delete only specific assets and leave the MIB and other related assets intact.

- If you delete a SNMP Configuration node, all SNMP assets and related event data (including historical data) associated with the SNMP Configuration file are deleted, all associated SNMP components are removed from the Monitored Components page, and all associated KPIs are removed from the Analytics Overview page. Note that all monitored instances are actually deleted from the system, and they cannot be recovered.

This option is appropriate if you have made changes, or have added SNMP MIB configuration types to the SNMP MIB configuration file, and want those changes to take effect.

To delete SNMP component assets

1. In My webMethods: **Navigate > Applications > Administration > Analytics > Infrastructure Components > Assets**
2. Do one of the following in the Discovered Assets panel:
 - Select the check box beside the SNMP Configuration to delete all SNMP assets and event data associated with this SNMP MIB configuration file.
 - Select the check box beside one or more SNMP assets to delete only the selected asset(s) without deleting other assets and event data associated with the SNMP MIB configuration file.

When you select an asset, the **Delete** button becomes active.

3. Click **Delete**.
A dialog is displayed asking if you want to delete the specified asset(s).
4. Answer **Yes** to delete the selected assets and configurations, or **No** to take no action.

Note: Deleted assets are not immediately removed from the Monitored Components page. Depending on the number of monitored components, it can take some time for deleted assets to be removed from the Monitored Components and Analytics Overview pages. As a workaround, after deleting the asset, wait a few moments before verifying that the information of the Monitored Components and Analytics Overview pages has been updated.

Suspending Monitoring

The suspend monitoring feature enables you to suspend monitoring of any discovered assets for any period of time. This feature is particularly useful when performing system maintenance to avoid generating inaccurate data in system reports and inappropriate alerts.

To suspend asset monitoring

1. In My webMethods: **Navigate > Applications > Administration > Analytics > Infrastructure Components > Assets**

2. Locate the asset that you want to suspend, and click the  icon in the Enabled column to suspend monitoring of that asset.

A dialog is displayed asking if you want to suspend monitoring.

3. Click **Yes** to suspend monitoring of the selected asset.

The **Status** column displays an  icon indicating that monitoring for the asset is suspended. Rule instances that are associated with this asset will not fire while it is suspended.

Resuming Monitoring

If you previously suspended monitoring, you can resume monitoring.

To resume asset monitoring

1. In My webMethods: **Navigate > Applications > Administration > Analytics > Infrastructure Components > Assets**

2. Locate the asset for which you want to resume monitoring, and click the  icon.

A dialog is displayed asking if you want to resume monitoring for the selected asset.

3. Click **Yes** to resume monitoring.

The Status column displays an  icon indicating that the asset is being monitored.

Monitoring Components

Monitored components are components of discovered assets, such as Integration Servers, Universal Messaging Servers, and Broker Servers, about which Optimize can report information. In order to monitor, or collect information about, a component of a discovered asset, one or more KPIs must be associated with that component. This section describes how to view and configure monitored components.

In an appropriately configured system, the Monitored Components page shows a tree view of component types that can be monitored. By default, a set of monitored component types is added to the Monitored Component page when you start the Infrastructure Data Collector, and these enable you to collect a baseline of information about your system infrastructure using KPIs. In addition, a set of KPIs is also provided for available components, and you must associate these with applicable components in order to initiate monitoring. Also, if you want to monitor additional system components, you must discover the appropriate assets as described earlier in this chapter, and then associate the appropriate KPIs with the appropriate components. An Edit icon is provided on the Monitored Components page to enable you to select those instances and KPIs.

Note that the Monitored Components page is empty until you start an Infrastructure Data Collector in your environment. The default set of component types that you can monitor is loaded when you start the Infrastructure Data Collector; however, the number of *available components* displayed in the Monitored Components page is zero until you have discovered assets of that component type (see ["Discovering Assets" on page 199](#) for instructions).

Tip: If the default set of monitored components does not appear after you have started the Infrastructure Data Collector, first confirm that the Infrastructure Data Collector, Analytic Engine, and My webMethods with the Optimize user interface have been installed and started. If these components have been installed and started, it is likely that the metadata was not properly loaded. Use the following steps to reload the metadata: (1.) Verify that the JMS Queue configuration is the same for both the Infrastructure Data Collector and the Analytic Engine. This can be verified from the **JNDI Configuration** "Connection Uri" setting on the Configure Servers tab in the Edit Environment page (**Navigate > Applications > Administration > System-Wide > Environments > Define Environments**). (2.) Shut down the Infrastructure Data Collector (see ["Starting and Stopping Infrastructure Data Collector " on page 46](#)). (3.) Open *Software AG_directory/profiles/InfraDC/configuration/infracdc/config/infracdc.cnf* in a text editor, and set the `metadataLoaded` property to `false`. (4.) Restart the Infrastructure Data Collector.

After you have discovered assets of a component type in your environment, you can configure the instances of that component type that you want to monitor, as well as the KPIs that you want to use to monitor the instances of that component type. For detailed information about the KPIs associated with each type of monitored component, see ["Monitored KPI Definitions" on page 349](#)

Note: The Broker data collector does not automatically detect when a Broker territory or gateway is deleted. As a result, the Monitored Components and Analytics Overview pages still show that the Territories, Gateways, and related KPIs exist. As a workaround, you should manually remove territories and gateways from the Monitored Components page and delete the related KPIs.

Displaying a List of Monitored Components

To display a list of Monitored components

In My webMethods: **Navigate > Applications > Administration > Analytics > Infrastructure Components > Monitored Components**

The Monitored Components page is displayed.

Monitored Components			
		Refresh	Export Table...
		21 Items	
TYPE ▲	SELECTED ▼	AVAILABLE ▼	ACTION
[-] Broker Server	0	0	
[-] Broker	0	0	
[-] Custom Adapter	0	0	
[-] Document Type	0	0	
[-] Infrastructure Data Collector	0	1	
[-] Integration Server	0	1	
[-] ART Adapter Service	0	0	
[-] ART Connection	0	5	
[-] ART Polling Notification	0	0	
[-] ART Listener	0	9	
[-] ART Listener Notification	0	0	
[-] IS Broker Transport	0	0	
[-] IS Email Port	0	0	
[-] IS FTP Port	0	0	
[-] IS HTTP Port	0	1	
[-] IS HTTPS Port	0	1	
[-] IS Package	0	21	
[-] IS Service	6	1276	
[-] IS Socket Port	0	0	
[-] IS Cluster	0	0	
[-] Repository Server	0	0	

You can do the following from the Monitored Components page:

- Click to expand the hierarchical tree to view all children.
- Click the name of a component type, or click , to edit the components and KPIs selected for this component type.
- Click ▼ beside a column heading to sort the display by the information in the column.

The following information about the available monitored component is displayed in the **Components** table:

Column	Description
TYPE	Monitored component type. If assets have been discovered for this component type, the number of discovered assets of this type appears in the Available column, and the component type is editable. Click the name of the component type, or the Edit icon, to change the selected component instances or KPIs associated with this component type. Click to view children of the component type.

Column	Description
SELECTED	Number of components of this type selected.
AVAILABLE	Number of components of this type available for selection.
ACTIONS	Contains an Edit  icon for each monitored component. This icon is unavailable for components types of which no assets have yet been discovered.

Control	Description
 Edit	Click to edit the associated component in the Edit Monitored Component page. (You can also click the name of the component to display the Edit Monitored Component page.)
Export Table	Click to display the contents of the Components table as a .csv file.

Configuring Monitor Components

The following procedure shows how to configure components for monitoring.

Note: In order to use the Object Status rule on any child component, the root component must also be selected for monitoring. If the root component is not selected, the selected child component will not have the Object Status rule. Without this rule, any online/offline alerts are not processed.

To configure a monitored component

1. Display the list of monitored components (see "[Displaying a List of Monitored Components](#)" on page 226).
2. Do either of the following:
 - Click the component type name. Names of component types for which assets have been discovered are blue and underlined.
 - In the **EDIT** column, click  **Edit** for the associated component type.
The settings for the component are displayed on the Edit Component Type page.

Edit Component Type: Broker



3. You can make the following changes to the component type settings:

If you want to...	Do this...
Add instances of this component type to monitor	<p>In the Selected Components panel:</p> <ol style="list-style-type: none"> Click Add Components. The Add Component Instances dialog displays. To reduce the number of instances displayed, enter matching text in the Filter: field and click Go. Select one or more instances from the list. Click OK to add the selected instances to the Selected Components list.
Remove instances of this component type	<p>In the Selected Components panel:</p> <ol style="list-style-type: none"> Select the component instance(s) you want to delete.

If you want to...	Do this...
Add KPIs to use for this component type	<p>b. Click Delete.</p> <p>In the Selected KPIs panel:</p> <p>a. Click Add KPIs.</p> <p>b. To reduce the number of KPIs displayed, enter matching text in the Filter: field and click Go.</p> <p>c. Select one or more KPIs from the list.</p> <p>d. Click OK to add the selected KPIs to the Selected KPIs list.</p>
Remove KPIs for this component type	<p>In the Selected KPIs panel:</p> <p>a. Select the KPI(s) you want to remove.</p> <p>b. Click Delete.</p>

Note: The Object Status KPI is automatically selected for monitoring once a component type instance has been selected, and it cannot be removed.

- Click **Save** to save changes, or click **Cancel** to close the Edit Component page without saving.

Using the Monitor and Suspend Services

Optimize for Infrastructure includes two services that enable you to activate or suspend asset monitoring remotely via web service calls. The two services are called Suspend Asset and Monitor Asset. These services accomplish the same actions as clicking the Monitor (🔴) and Suspend (🟢) icons on the Assets page. They both use the same two parameters as described below:

- `assetName`: Name of the asset.
- `assetType`: The type of asset. Possible values include Integration Server, IS Cluster, Broker Server, SNMP Agent.

Further details about these services are contained in the WSDL file used to invoke them which can be obtained at the following location:

```
http://<host>:<port>/wsstack/services/InfraDCRemoteServices?wsdl
```

Note that `<host>` is the name on the host machine on which Infrastructure Data Collector is installed, and `<port>` is the port via which Infrastructure Data Collector is accessed.

These services can be implemented using the WSDL file in the appropriate manner for your system configuration.

8 Setting Up Logging for Optimize

- Overview 232
- Setting Up Logging for Analytic Engines and Data Collectors 232
- Setting Up Logging for Infrastructure Data Collectors 233

Overview

Optimize stores log files for each Analytic Engine and Web Service Data Collector in the following locations:

Subcomponent	Location of Log Files
Analytic Engine	<i>Optimize_directory</i> \analysis\logs directory on the Analytic Engine host system
Web Service Data Collector	<i>Optimize_directory</i> \dataCollector\logs directory on the Web Service Data Collector host system

The log files are as follows:

File Type	Contents	File Name
Server	Warning messages generated by the component.	server.log
Error	Errors generated by the component. Check this log first for critical issues with the component.	error.log
Information	Informational messages generated by the component.	info.log

Optimize creates a log file of each type every day. Typically, the data in the log file is collected for a time period ranging from midnight to midnight.

By default, log file size is unlimited. To change these settings, follow the instructions in ["Setting Up Logging for Analytic Engines and Data Collectors"](#) on page 232 below.

Setting Up Logging for Analytic Engines and Data Collectors

This section explains how to locate the logging settings for each Analytic Engine and Web Service Data Collector. For detailed instructions on how to configure logging, see *Configuring BAM*.

To set up logging for Analytic Engines and Data Collectors

1. In My webMethods: **Navigate > Applications > Administration > System-Wide > Environments > Define Environments.**

The Define Environments page displays.

2. Click the name of the configured environment for which you want to set up logging.

The Edit Environment page is displayed.

3. Click the **Configure Servers** tab.
4. On the **CONFIGURATION** tree, find the Analytic Engine or Web Service Data Collector for which you want to set up logging, and click **Journal Logging** under the name of the Analytic Engine or Data Collector.

Note: You can also adjust logging settings for all Analytic Engines and Web Service Data Collectors in your environment by selecting **Journal Logging** in the **Default Settings** section of the Configuration tree.

The Journal Loggers screen displays.

LOGGER	THRESHOLD	TARGET NAME	TARGET TYPE	ASSIGN TARGETS
<input type="checkbox"/> root	INFO	StdErr	Console (System.Err)	
<input type="checkbox"/>		DailyFile	File system (./logs/server.log)	
<input type="checkbox"/>		ErrorFile	File system (./logs/error.log)	
<input type="checkbox"/>		InfoFile	File system (./logs/info.log)	

5. See information about defining journal log settings in *Configuring BAM* for configuration instructions.
6. Click **Save** and **Finish**.

Setting Up Logging for Infrastructure Data Collectors

The instructions for configuring logging settings for Infrastructure Data Collector are located in the Defining Collector Settings section of the Managing webMethods Environments chapter of the *Configuring BAM* guide.

9 Installing and Configuring the WmOptimize Package

■ Installing and Configuring the WmOptimize Package	236
■ Adding Optimize Users to the OptimizeUsers Group	240

Installing and Configuring the WmOptimize Package

WmOptimize is a standard webMethods package that can be installed on Integration Server. It is available to all users who have an Analytic Engine configured on their systems.

To access the WmOptimize package, you must select the **OptimizeSupport X.x** item within the installation tree when installing Integration Server. For specific instructions about installing Integration Server, refer to *Installing webMethods and Intelligent Business Operations Products* for the appropriate version of the product that you want to use.

Enabling the WmOptimize Package

After installation, you must enable the WmOptimize package and verify or configure the default Broker or Universal Messaging URL address for the JMS provider used by the Analytic Engine as described in the following procedures. You must complete these procedures for every Integration Server that runs the WmOptimize package.

To enable the WmOptimize Package on Integration Server

1. Start Integration Server if it is not already running.
2. In the navigation area on the left side of the page, click **Packages > Management**.
The Packages > Management page is displayed, showing a list of available packages.
3. Locate the WmOptimize package in the list and click **No** in the Enabled column.
The **No** changes to a **Yes** to indicate that the package is enabled.

Configuring the WmOptimize Package

If you are using Optimize for Process and processes are enabled for analysis, you must connect the Process Engines running on the Integration Servers in your environment to the Optimize Analytic Engines so that My webMethods can display process metrics.

To configure the WmOptimize package on Integration Server

1. Start Integration Server, if it is not already running.
2. In the navigation area on the left side of the page, click **Packages > Management**.
3. On the Packages > Management page, click  for the WmOptimize package.
The Configuration Settings page enables you to configure the default Analytic Engine host and port, as well as the Broker or Universal Messaging instances to which WmOptimize sends information. Default values appear for the initial Broker instances. If you want WmOptimize to send information to additional Broker or Universal Messaging instances, you must do so by creating new connections using the `pub.optimize.monitoring:createNewConnection` service. For more information, see "[pub.optimize.monitoring:createNewConnection](#)" on page 518.

Note: If your Optimize system operates in a EDA enabled environment using NERV, then you must configure the NERV JMS Provider setting in the profile for the Integration Server that is running on the same machine as your Analytic Engine. By default, the NERV JMS Provider is defined as `nsp://<host_name>:9000`. To configure this setting, you must edit the `COMMON-SYSPROPS-com.softwareag.eda.nerv.properties` file using the Command Central interface. Detailed instructions for configuring the NERV JMS Provider can be found in the "Modifying Transport Layer Configuration" section of the *Implementing Event-Driven Architecture with Software AG Products* guide. For more information about configuring Optimize for subscription and publication of events in an EDA environment, see ["Configuring EDA/NERV Settings for Optimize" on page 94](#).

4. Accept or modify the following settings:

<u>Setting</u>	<u>Description</u>						
Analytic Engine Host	Displays the name of the computer where the WmOptimize package is installed. Accept the default value or change it to the host name of the Analytic Engine you want to send data to.						
Analytic Engine Port	Displays the port number of the Analytic Engine. Accept the default value of 12503 or change it to the port number you want to use.						
Analytic Engine User	Displays the user name for connecting to the Optimize Analytic Engine.						
Analytic Engine Password	Displays the user password for connecting to the Optimize Analytic Engine.						
Analytic Engine Use SSL	Specifies whether you want to use SSL to connect to the Optimize Analytic Engine or not. Selecting the option enables all the other SSL-related fields for the WmOptimize package.						
	<table border="1"> <thead> <tr> <th><u>Set to...</u></th> <th><u>To specify that...</u></th> </tr> </thead> <tbody> <tr> <td>Clear (the default)</td> <td>You do not want to use SSL connection to the Optimize Analytic Engine.</td> </tr> <tr> <td>Selected</td> <td>You want to use SSL connection to the Optimize Analytic Engine. For more</td> </tr> </tbody> </table>	<u>Set to...</u>	<u>To specify that...</u>	Clear (the default)	You do not want to use SSL connection to the Optimize Analytic Engine.	Selected	You want to use SSL connection to the Optimize Analytic Engine. For more
<u>Set to...</u>	<u>To specify that...</u>						
Clear (the default)	You do not want to use SSL connection to the Optimize Analytic Engine.						
Selected	You want to use SSL connection to the Optimize Analytic Engine. For more						

Setting	Description						
	<p>information about SSL configuration for the Analytic Engine, see the PDF publication <i>Configuring BAM</i>.</p>						
JMS Server URL	<p>Displays URL values for the JMS provider host, port, and name. The supported JMS providers are webMethods Broker and webMethods Universal Messaging. Accept the default URL values or substitute the values you want to use. For example:</p> <pre data-bbox="667 636 1138 688">broker://localhost:6849/Broker #1 nsp://localhost:9000</pre>						
Use SSL	<p>Specifies whether you want to use SSL to connect to Broker or Universal Messaging. Selecting the option enables all the other SSL-related fields for the WmOptimize package and renders them required.</p> <div data-bbox="618 867 1365 1003" style="background-color: #f0f0f0; padding: 5px;"> <p>Important To remove the SSL connections for the WmOptimize package after they have been activated, you must reload the package.</p> </div> <table border="1" data-bbox="618 1035 1333 1686"> <thead> <tr> <th data-bbox="618 1035 748 1066">Set to...</th> <th data-bbox="764 1035 1333 1066">To specify that...</th> </tr> </thead> <tbody> <tr> <td data-bbox="618 1108 748 1213">Clear (the default)</td> <td data-bbox="764 1108 1333 1182">You do not want to use SSL connection to Broker or Universal Messaging.</td> </tr> <tr> <td data-bbox="618 1255 748 1287">Selected</td> <td data-bbox="764 1255 1333 1675">You want to use SSL connection to the Broker or Universal Messaging. For more information about SSL configuration for the Broker, see the PDF publication <i>Administering webMethods Broker</i>. For more information about SSL configuration for Universal Messaging, see the Universal Messaging documentation available at Software AG Documentation web site, webMethods Product Suite > webMethods Product Suite version > Documentation by Product > Universal Messaging.</td> </tr> </tbody> </table>	Set to...	To specify that...	Clear (the default)	You do not want to use SSL connection to Broker or Universal Messaging.	Selected	You want to use SSL connection to the Broker or Universal Messaging. For more information about SSL configuration for the Broker, see the PDF publication <i>Administering webMethods Broker</i> . For more information about SSL configuration for Universal Messaging, see the Universal Messaging documentation available at Software AG Documentation web site , webMethods Product Suite > webMethods Product Suite version > Documentation by Product > Universal Messaging .
Set to...	To specify that...						
Clear (the default)	You do not want to use SSL connection to Broker or Universal Messaging.						
Selected	You want to use SSL connection to the Broker or Universal Messaging. For more information about SSL configuration for the Broker, see the PDF publication <i>Administering webMethods Broker</i> . For more information about SSL configuration for Universal Messaging, see the Universal Messaging documentation available at Software AG Documentation web site , webMethods Product Suite > webMethods Product Suite version > Documentation by Product > Universal Messaging .						
Use Encryption	<p>Specifies whether you want to use encryption for the connection or not.</p> <table border="1" data-bbox="618 1829 1333 1860"> <thead> <tr> <th data-bbox="618 1829 748 1860">Set to...</th> <th data-bbox="764 1829 1333 1860">To specify that...</th> </tr> </thead> <tbody> </tbody> </table>	Set to...	To specify that...				
Set to...	To specify that...						

<u>Setting</u>	<u>Description</u>						
	<p>Clear The connection will be authenticated but no (the data will be encrypted. default)</p> <p>Selected All traffic to the JMS server and/or the Optimize Analytic Engine on this connection will be encrypted.</p>						
SSL Keystore File	(Required if Use SSL is selected) Displays the absolute path and file name of the keystore file containing the SSL certificate to be used by the test client to join the client group. Type in the path and file name that you want to use.						
Event Enable	Specifies whether you want to start publishing events to other consumer products, such as PPM.						
	<table border="1"> <thead> <tr> <th><u>Set to...</u></th> <th><u>To specify that...</u></th> </tr> </thead> <tbody> <tr> <td>Clear (the default)</td> <td>No events (push process control operations, push step control operations, process execution errors) will be published to consumers.</td> </tr> <tr> <td>Selected</td> <td>All events will be published to consumers.</td> </tr> </tbody> </table>	<u>Set to...</u>	<u>To specify that...</u>	Clear (the default)	No events (push process control operations, push step control operations, process execution errors) will be published to consumers.	Selected	All events will be published to consumers.
<u>Set to...</u>	<u>To specify that...</u>						
Clear (the default)	No events (push process control operations, push step control operations, process execution errors) will be published to consumers.						
Selected	All events will be published to consumers.						
SSL Truststore File	(Required if Use SSL is selected) Displays the absolute path and file name of the truststore file containing the trusted root for the SSL certificate stored in the specified keystore. Type in the path and file name that you want to use.						
SSL Keystore Password	(Required if Use SSL is selected) Contains the password to the keystore file. Type in the password.						
ISO UTC Time Handling	Defines whether the <i>yyyy-MM-dd'T'HH:mm:ss.SSS'Z'</i> time/date format should use UTC time zone or the locally set time zone.						
	<table border="1"> <thead> <tr> <th><u>Set to...</u></th> <th><u>To...</u></th> </tr> </thead> <tbody> </tbody> </table>	<u>Set to...</u>	<u>To...</u>				
<u>Set to...</u>	<u>To...</u>						

Setting	Description
	Selected (the default) Use the UTC time zone.
	Clear Use the locally set time zone.

5. Click **Submit**.

Adding Optimize Users to the OptimizeUsers Group

To allow users to invoke the services from the WmOptimize package using their credentials, you must set up the Central Users feature. If you do not configure the Central Users feature, you must ensure that each Optimize user defined in My webMethods has a corresponding user account defined in the Integration Server. After you enable and configure the Central Users feature, to complete the setup for Optimize, you must add Optimize to the group of allowed users.

When a user issues an Optimize request, an internal service in the WmOptimize package on the Integration Server is invoked to handle that request. The service is invoked using the credentials (that is, the user name and password) of the requesting user. As a result, the Integration Server must authenticate the user who issued the Optimize request. The user must be a member of the OptimizeUsers group. Note that the Optimize ACL and the OptimizeUsers group are automatically created when the WmOptimize package is loaded in the Integration Server.

To use the Central Users feature, you must enable and configure it from the Integration Server Administrator.

For instructions about how to enable and configure the Central Users Management feature, and how to manage users and groups, see *webMethods Integration Server Administrator's Guide*.

10 Integration with Software AG MashZone

- Overview 242
- Displaying Data from Optimize Analytic Engine into Software AG MashZone 242
- Constructing Queries to Display Data from Optimize Analytic Engine into Software AG MashZone 244

Overview

If you use Optimize for Process and Optimize for Infrastructure and your processes are enabled for analysis and execution, you can further process the obtained data (KPI instance readings, KPI definitions, problems and dimension instances) with Software AG MashZone.

Software AG MashZone is a browser-based application that enables you to analyze and visualize data from different distributed data sources. It provides a framework for creating user interface dashboards based on data obtained from various locations, including a URL, an XML file, an Excel spreadsheet, as well as other sources.

To ensure communication between the Analytic Engine and MashZone, a number of services exist in the WmOptimize package. For more information on available services, see "[Built-In Services](#)" on page 501.

You can extract the data in two possible ways:

- **Directly from Optimize Analytic Engine** - your queries invoke directly the Analytic Engine API to extract the data.
- **From webMethods Integration Server** - your queries use the `pub.optimize.mashzone.adapter.retrieveData` built-in service from the WmOptimize package deployed on to extract the data from the Analytic Engine.

Two ways exist for constructing queries for obtaining data from Optimize Analytic Engine and displaying it into Software AG MashZone. You can do this:

- by using the My webMethods Server user interface.
- by constructing your own queries using the `pub.optimize.mashzone.adapter.retrieveData` built-in service from the WmOptimize package.

Note: Before you begin this procedure, you must configure the default Analytic Engine host and port, as well as the default broker to which WmOptimize sends information, as described in "[Configuring the WmOptimize Package](#)" on page 236. You must also have a working understanding of MashZone and its operation, and be familiar with the MashZone documentation. For more information, see the MashZone online help or the documentation available within your MashZone installation.

Displaying Data from Optimize Analytic Engine into Software AG MashZone

To display data from Optimize Analytic Engine into MashZone

1. Build and upload a process for execution with Software AG Designer.
For more information, see *webMethods BPM Process Development Help*.

- Go to the KPI Detail page:

To go to...	Do this...
KPI Detail page	<ul style="list-style-type: none"> ■ Click  from the KPI Summary page. ■ Click  in the graph from the Rule Detail page. ■ Click  in the KPI Instance's DETAIL column from the Root Cause Analysis page.

- Click **Generate MashZone String**.

By default, the dialog box that appears contains a query string with data for the selected KPI instance obtained directly from the Analytic Engine API. You can modify the generated query, using the options in the dialog box.

Select...	To obtain a query string with data...
The Generate MashZone String to extract data from Integration Server option	For instances of the selected KPI by extracting the data from Integration Server using the <code>pub.optimize.mashzone.adapter.retrieveData</code> built-in service from the WmOptimize package. If you select this option, the data queries invoke the built-in service available in your Integration Server installation.
The Generate MashZone String for all KPI instances option	For all instances of the selected KPI.
The Generate MashZone String for LAST_DAY option	For instances of the selected KPI for the current day.
Both the Generate MashZone String for all KPI instances and the Generate MashZone String for LAST_DAY options	For all instances of the selected KPI for the current day.

- Copy the query string and paste it into MashZone to further process the information.

Note: The data will be aggregated according to the settings you defined and it will be displayed in GMT time format.

For more information about how to use MashZone, see the MashZone online help or the documentation available within your MashZone installation.

Constructing Queries to Display Data from Optimize Analytic Engine into Software AG MashZone

You can construct your own queries and extract data (such as KPI instance readings, dimension instances, problems, and KPI definitions) from Optimize Analytic Engine. The data will be obtained using the `pub.optimize.mashzone.adapter.retrieveData` built-in service and can be pasted and further processed in MashZone.

The sections below contain the supported query parameters and their values, grouped according to the data type.

General Purpose Parameters

dataType **String** Specifies the type of data you are looking for in Optimize.

<u>Set to...</u>	<u>To look for...</u>
KPI_INSTANCE_READING	KPI instance readings data in Optimize.
PROBLEM	Problems data in Optimize.
KPI_DEFINITION	KPI definitions data in Optimize.
DIMENSION_INSTANCE	Dimension instances data in Optimize.

keys **String List** Specifies a list of strings that act as the keys of the query parameters. The length of the *keys* string list must be equal to the length of the *values* string list, so the two can form valid pairs.

Note: The *keys* parameter is required only for KPI instance readings and dimension instances data. You can omit it when retrieving problems or KPI definitions data.

values **String List** Specifies a list of strings that act as the values of the query parameters. The length of the *keys* string list must be equal to the length of the *values* string list, so the two can form valid pairs.

Note: The *values* parameter is required only for KPI instance readings and dimension instances data. You can omit it when retrieving problems or KPI definitions data.

maxRows **String** Optional. Specifies the maximum number of data rows that you want the service to return. If you do not specify a value or specify a value less than 0, the service will use the default value of 1000. If you specify a value equal to 0, the service returns only the table header.

KPI Instance Readings Parameters

kpiName **String** Required when no *kpiInstanceId* is provided. Specifies the name of the KPI for which to return KPI instance readings.

kpiInstanceId **Integer** Required when no *kpiName* is provided. Specifies a unique identifier of the KPI instance for which you want readings.

dimensionValue **String** Optional. Specifies a dimension value for the KPI instance readings that you want returned. Optimize will search through all the dimensions included in the KPI dimension's hierarchy. The closest match to the provided value is returned. For example, a value of `abc` will match all strings that contain `abc`. For an exact match, use quotation marks ("`abc`").

aggregationInterval **String** Optional. Specifies how you want the service to aggregate the retrieved KPI instance readings. This is helpful when you expect a large amount of data returned. For example, if you set the time period to `LAST_YEAR`, the search might return more than 500,000 data readings. In this case, you might use the `ONE_WEEK` aggregation interval to have the service aggregate the data readings on a weekly basis, and, as a result, return no more than 52 entries.

Set to...

`ONE_MINUTE`

To aggregate the collected data...

For each minute in the collection period.

FIVE_MINUTE	For each five minute interval in the collection period.
TEN_MINUTE	For each ten minute interval in the collection period.
FIFTEEN_MINUTE	For each fifteen minute interval in the collection period.
THIRTY_MINUTE	For each thirty minute interval in the collection period.
ONE_HOUR	For each one hour in the collection period.
TWO_HOUR	For each two hour interval in the collection period.
FOUR_HOUR	For each four hour interval in the collection period.
EIGHT_HOUR	For each eight hour interval in the collection period.
TWELVE_HOUR	For each twelve hour interval in the collection period.
ONE_DAY	For each one day in the collection period.
ONE_WEEK	For each one week in the collection period.

timePeriod

String Required when no *startDate* and *endDate* are provided. Defines the time period for which to return KPI instance readings. For example, if the current date is January 10, 2009 and you specify `LAST_YEAR` as time period, the service will return data only from January 1 to January 10, 2009.

Set to...	To retrieve KPI instance readings from...
<code>LAST_HOUR</code>	The current hour.

	LAST_DAY	The current day.
	LAST_WEEK	The current week.
	LAST_MONTH	The current month.
	LAST_QUARTER	The current quarter.
	LAST_YEAR	The current year.
<i>startDate</i>	String	Required when no <i>timePeriod</i> is provided. Specifies the start of the time period for which to retrieve KPI instance readings. The time must be in the following format: yyyy-MM-dd'T'HH:mm:ssZ.
<i>endDate</i>	String	Required when no <i>timePeriod</i> is provided. Specifies the end of the time period for which to return KPI instance readings. The time must be in the following format: yyyy-MM-dd'T'HH:mm:ssZ.

Dimension Instances Parameters

<i>kpiName</i>	String	Specifies the name of the KPI associated with the dimension instances that you want returned. No data will be returned unless there is an exact match to the provided value.
<i>dimensionName</i>	String	Specifies the name of the dimension instances that you want returned. No data will be returned unless there is an exact match to the provided value.
<i>dimensionAttribute</i>	String	Specifies the attribute of the dimension instances that you want returned. No data will be returned unless there is an exact match to the provided value.
<i>dimensionValue</i>	String	Optional. Specifies the value pattern of the dimension instances that you want returned. The closest match to the provided value is returned. For example, a value of <code>abc</code> will match all strings that contain <code>abc</code> . For an exact match, use quotation marks (<code>"abc"</code>).

KPI Definitions Parameters

kpiName **String** Optional. Specifies the name of the KPI associated with the KPI definitions that you want returned. The closest match to the provided value is returned. For example, a value of *abc* will match all strings that contain *abc*. For an exact match, use quotation marks ("abc").

Problems Parameters

type **String** Optional. Specifies the type of problems you want returned. The default value is `RULE_VIOLATION`.

<u>Set to...</u>	<u>To look for...</u>
<code>RULE_VIOLATION</code>	Rule violation events.
<code>SLA_VIOLATION</code>	SLA violation events.
<code>ANY_TYPE</code>	Any type of problems.

description **String** Optional. Specifies the description of the problems that you want returned. The closest match to the provided value is returned. For example, a value of *abc* will match all strings that contain *abc*. For an exact match, use quotation marks ("abc").

timePeriod **String** Required when no *startDate* and *endDate* are provided. Defines the time period for which to return problems. For example, if the current date is January 10, 2009 and you specify `LAST_YEAR` as time period, the service will return data only from January 1 to January 10, 2009.

<u>Set to...</u>	<u>To retrieve problems from...</u>
<code>LAST_HOUR</code>	The current hour.
<code>LAST_DAY</code>	The current day.
<code>LAST_WEEK</code>	The current week.

	LAST_MONTH	The current month.
	LAST_QUARTER	The current quarter.
	LAST_YEAR	The current year.
<i>startDate</i>	String Optional. Specifies the start of the time period for which to return problems. The time must be in the following format: yyyy-MM-dd'THH:mm:ss.SSSZ.	
<i>endDate</i>	String Optional. Specifies the end of the time period for which to return problems. The time must be in the following format: yyyy-MM-dd'THH:mm:ss.SSSZ.	
<i>SLAOn</i>	String Optional. Specifies whether the rule is of SLA type or not.	
	Set to...	To specify that...
	true	The rule has an SLA attribute defined.
	false	The rule does not have an SLA attribute defined.
<i>SLACustomer</i>	String Optional. Specifies the customer name associated with the Service Level Agreement (SLA) of the rule that was violated to cause the problem event.	
<i>severity</i>	String Optional. Specifies the severity level assigned to the rule that was violated to cause the problem event.	
<i>duration</i>	String Optional. Specifies the duration of the problem event in milliseconds.	
<i>alertUser</i>	String Optional. Specifies the user(s) notified when the problem occurs.	
<i>ruleStatus</i>	String Optional. Specifies the rule status for the problem.	
	Set to...	To look for problems...
	inCompliance	In compliance with the rule.

	outOfCompliance	Out of compliance with the rule.
<i>relation</i>	String Optional. Specifies the name of the process model to which the problem is related.	
<i>category</i>	String Optional. Specifies the category of problems that you want returned. If the parameter is not set, the query retrieves only business data.	
	Set to...	To get...
	0	Problems data of business type.
	1	Problems data of system type.

II For Developers

■ Configuring and Using the Web Service Data Collector	253
■ Setting Up Web Service Actions	281
■ Optimize REST Services API	287

The chapters in this part of the guide contain information about the tasks primarily performed by Optimize developers.

11 Configuring and Using the Web Service Data Collector

■ Overview	254
■ Configuring the Web Service Data Collector	254
■ Sending Event and Dimensional Data	255
■ Sending Process Data	261

Overview

The Web Service Data Collector includes two interfaces that client applications can use to send data to Optimize:

- **The dimensionalData interface** (IMonitorDataCollector, or WSDimensionalDataCollector) is used for sending both business and event and dimensional data.
- **The processData interface** (IProcessDataCollector, or WSProcessDataCollector) is used for sending process data.

A Web Services Definition Language (WSDL) document for each interface is installed with Optimize. You can use these WSDL documents to build the client applications.

Configuring the Web Service Data Collector

To avoid losing messages because of a high volume of messages or because the JMS provider fails or is down for an extended period of time, configure the following queue sizes in webMethods Central Configuration:

- jmsResendQueueSize
- eventQueueSize
- dimensionQueueSize

To configure queue sizes in webMethods Central Configuration

1. In My webMethods: **Navigate > Applications > Administration > System-Wide > Environments > Define Environments.**
The Define Environments page displays.
2. Click the name of the configured environment for which you want to configure queue sizes.
The Edit Environment page displays.
3. Click the **Configure Servers** tab.
4. On the **CONFIGURATION** tree, find the data collector for which you want to configure queue sizes, and click **Data Collector Settings** under the name of the data collector.
The Data Collector Settings page displays.
5. Edit the settings for the three queue sizes. Default settings= 50000.
6. Click **Save** to save your changes, and click **Finish** to save your settings.
7. Put the changes into effect. See "[Changing Your Configuration Settings](#)" on page 57

Generating the Client

You can automatically generate a client from the Web Service Data Collector WSDL documents using Software AG Designer. In the Designer Service Development perspective, you generate the client as a Web service connector. Designer generates the Web service connector from a specified WSDL document when you generate a web service descriptor. A Web service connector is an Integration Server service that has an input and output signature that corresponds to the input and output messages from the WSDL document from which it was created. You provide to the connector the parameters that are required by the Web service, and the connector creates the required Service Oriented Architecture Protocol (SOAP) message and decodes the SOAP response automatically. For more information about Web service connectors, see the *Web Services Developer's Guide*.

Note: The Web Service Data Collector WSDL documents declare some common types. When you generate the Web service connector for a subsequent WSDL document, you will receive an error about duplicate definitions. You can ignore this error.

The WSDL documents are available through the following URLs:

- `http://<hostName>:<port>/services/WSPProcessDataCollector.wsdl`
- `http://<hostName>:<port>/services/WSDimensionalDataCollector.wsdl`

The `<hostName>` value defines the Web Service Data Collector host system. The default port for the Web Service Data Collector is 12603.

Sending Event and Dimensional Data

The recommended method for publishing monitor data into Optimize is to use the Web Service Data Collector's `dimensionalData` interface described in this section. The Web Service Data Collector contains web services for sending data as *events* and *dimensions*, where events capture metric data and dimensions capture metadata about the metric data. For more information, see "[Configuring KPIs](#)" on page 139

The `addEvent (event)` and `addEvents (events)` methods pass data as event types and a list of details about the event types, presented as a series of key/value data pairs. For example:

Data	Value
Event type	OrderProcess
Key/value pairs	Revenue = 500 Customer = 9 Product = 71 OrderID = 123456

The [addStronglyTypedEvent \(event\)](#) and [addStronglyTypedEvents \(events\)](#) methods are included only for backward compatibility.

The [addDimension \(dimension\)](#) and [addDimensions \(dimensions\)](#) methods pass metadata for the event data as a dimension type, or name, and a list of details about the dimension type presented as a series of key/value data pairs. For example:

Data	Value
Dimension 1	
Name	Customer
Key/value pairs	ID = 9 Name = ABC Corp Region = Central
Dimension 2	
Name	Product
Key/value pairs	ID = 71 Name = Widget

The [addStronglyTypedDimension \(dimension\)](#) and [addStronglyTypedDimensions \(dimensions\)](#) methods pass metadata for the event data as a dimension type, or name, and a list of details about the dimension type presented as a series of key/value data pairs. For example:

Data	Value
Dimension 1	
Name	Customer
Pairs	ID = 9 Name = ABC Corp Region = Central
Dimension 2	
Name	Product
Pairs	ID = 71 Name = Widget

Note: The methods of this interface use the classes contained in the `bam-wsdc.jar` file in the `Optimize_directory/dataCollector/lib` directory.

addEvent (event) and addEvents (events)

The addEvent method sends a single event. The addEvents method sends multiple events. The addEvents method does not support Web service callbacks.

Input Parameters (all parameters are required unless specified as optional)

<i>event</i>	Document Container for the list of event attributes.						
<i>attributes</i>	Document List Container for event attribute information.						
	<table> <thead> <tr> <th><u>Parameter</u></th> <th><u>Value</u></th> </tr> </thead> <tbody> <tr> <td><i>name</i></td> <td>String Attribute name (such as Revenue).</td> </tr> <tr> <td><i>value</i></td> <td>String or key/value Attribute value (such as 500).</td> </tr> </tbody> </table>	<u>Parameter</u>	<u>Value</u>	<i>name</i>	String Attribute name (such as Revenue).	<i>value</i>	String or key/value Attribute value (such as 500).
<u>Parameter</u>	<u>Value</u>						
<i>name</i>	String Attribute name (such as Revenue).						
<i>value</i>	String or key/value Attribute value (such as 500).						
<i>eventType</i>	String Type of the event (such as OrderProcess).						
<i>time</i>	<p>String Time to associate with the <i>eventType</i>. The time zone parameter of the date must be set to "GMT" in Designer. For input from Integration Server or Designer, the time must be in the following format:</p> <pre>yyyy-MM-dd'T'HH:mm:ss.SSS'Z'</pre> <pre>yyyy-MM-dd'T'HH:mm:ss.SSSZ</pre> <pre>yyyy-MM-dd'T'HH:mm:ss.SSS</pre> <pre>yyyy-MM-dd HH:mm:ss.SSS</pre> <p>Note: If left blank, this field defaults to the current data and time.</p>						
<i>auth</i>	Not used.						
<i>_port</i>	Not used.						

addEventWithCallback

The `addEventWithCallback` method sends a single event and supports Web service callbacks.

Input Parameters (all parameters are required unless specified as optional)

<i>event</i>	Document Container for the list of event attributes.						
<i>attributes</i>	Document List Container for the event attribute information.						
	<table border="1"> <thead> <tr> <th>Parameter</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td><i>name</i></td> <td>String Attribute name (such as Revenue).</td> </tr> <tr> <td><i>value</i></td> <td>String or key/value Attribute value (such as 500).</td> </tr> </tbody> </table>	Parameter	Value	<i>name</i>	String Attribute name (such as Revenue).	<i>value</i>	String or key/value Attribute value (such as 500).
Parameter	Value						
<i>name</i>	String Attribute name (such as Revenue).						
<i>value</i>	String or key/value Attribute value (such as 500).						
<i>eventType</i>	String Type of the event (such as OrderProcess).						
<i>time</i>	<p>String Time to associate with the <i>eventType</i>. The time zone parameter of the date must be set to "GMT" in Designer. For input from Integration Server or Designer, the time must be in the following format:</p> <pre>yyyy-MM-dd' T' HH:mm:ss.SSS' Z'</pre> <pre>yyyy-MM-dd' T' HH:mm:ss.SSSZ</pre> <pre>yyyy-MM-dd' T' HH:mm:ss.SSS</pre> <pre>yyyy-MM-dd HH:mm:ss.SSS</pre> <p>Note: If left blank, this field defaults to the current data and time.</p>						
<i>callbackEndpoint</i>	<p>String (Optional) Location of the Web service published using the EventCallback.wsdlWSDL located in the <i>Optimize_directory</i> /dataCollector/conf/glue/wSDL directory. Use the following format:</p> <pre>http://{host};{port}/{path}/{serviceName}.wsdl</pre> <p>If the value is null, the event will be processed but the callback will not be issued.</p>						

<i>timeoutMS</i>	Int (Optional) Overrides the Analytic Engine's configured default event processing timeout value (currently set to 100000 milliseconds). Value must be greater than the Analytic Engine's configured default to override.
<i>correlationId</i>	String (Optional) Unique identifier generated by the client sent back to the Web service during the callback operation. If value is null, the event will be processed but callback will not be issued.
<i>notifyOnFailureOnly</i>	Boolean (Optional) If true, Web service calls are issued only when a failure occurs.
<i>auth</i>	Not used.
<i>_port</i>	Not used.

addStronglyTypedEvent (event) and addStronglyTypedEvents (events)

The `addStronglyTypedEvent` method sends a single strongly typed event. The `addStronglyTypedEvents` method sends multiple strongly typed events.

Note: This method is deprecated, and it is supported only for backward compatibility. In a future release, the `addStronglyTypedEvent` and `addStronglyTypedEvents` methods will be removed.

Input Parameters (all parameters are required unless specified as optional)

<i>event</i>	Document Container for event information.				
<i>eventType</i>	String Type of the event (such as <code>OrderProcess</code>).				
<i>pairs</i>	Document List Container for the list of key/value event data.				
	<table> <thead> <tr> <th><u>Parameter</u></th> <th><u>Value</u></th> </tr> </thead> <tbody> <tr> <td><i>attribute</i></td> <td>String Attribute name (such as <code>Revenue</code>).</td> </tr> </tbody> </table>	<u>Parameter</u>	<u>Value</u>	<i>attribute</i>	String Attribute name (such as <code>Revenue</code>).
<u>Parameter</u>	<u>Value</u>				
<i>attribute</i>	String Attribute name (such as <code>Revenue</code>).				

	<i>value</i>	String or key/value Attribute value (such as 500).
	<i>timestamp</i>	<p>String The time when the event occurred. The time zone parameter of the date must be set to "GMT" in Designer. For input from Integration Server or Designer, the time must be in the following format:</p> <pre> yyyy-MM-dd' T' HH:mm:ss.SSS' Z' yyyy-MM-dd' T' HH:mm:ss.SSSZ yyyy-MM-dd' T' HH:mm:ss.SSS yyyy-MM-dd HH:mm:ss.SSS </pre> <p>Note: If left blank, this field defaults to the current data and time.</p>
<i>auth</i>		Not used.
<i>_port</i>		Not used.

addDimension (dimension) and addDimensions (dimensions)

The addDimension method sends a single dimension. The addDimensions method sends multiple dimensions.

Input Parameters (all parameters are required unless specified as optional)

<i>dimension</i>	Document Container for the list of dimension attributes.												
	<table border="0"> <tr> <td><i>attributes</i></td> <td>Document Container for dimension attribute information.</td> </tr> <tr> <td></td> <td> <table border="0"> <tr> <th><u>Parameter</u></th> <th><u>Value</u></th> </tr> <tr> <td><i>key</i></td> <td>String Attribute name (such as Revenue).</td> </tr> <tr> <td><i>value</i></td> <td>String or key/value Attribute value (such as 500).</td> </tr> <tr> <td><i>name</i></td> <td>String Name of the dimension (such as Customer).</td> </tr> </table> </td> </tr> </table>	<i>attributes</i>	Document Container for dimension attribute information.		<table border="0"> <tr> <th><u>Parameter</u></th> <th><u>Value</u></th> </tr> <tr> <td><i>key</i></td> <td>String Attribute name (such as Revenue).</td> </tr> <tr> <td><i>value</i></td> <td>String or key/value Attribute value (such as 500).</td> </tr> <tr> <td><i>name</i></td> <td>String Name of the dimension (such as Customer).</td> </tr> </table>	<u>Parameter</u>	<u>Value</u>	<i>key</i>	String Attribute name (such as Revenue).	<i>value</i>	String or key/value Attribute value (such as 500).	<i>name</i>	String Name of the dimension (such as Customer).
<i>attributes</i>	Document Container for dimension attribute information.												
	<table border="0"> <tr> <th><u>Parameter</u></th> <th><u>Value</u></th> </tr> <tr> <td><i>key</i></td> <td>String Attribute name (such as Revenue).</td> </tr> <tr> <td><i>value</i></td> <td>String or key/value Attribute value (such as 500).</td> </tr> <tr> <td><i>name</i></td> <td>String Name of the dimension (such as Customer).</td> </tr> </table>	<u>Parameter</u>	<u>Value</u>	<i>key</i>	String Attribute name (such as Revenue).	<i>value</i>	String or key/value Attribute value (such as 500).	<i>name</i>	String Name of the dimension (such as Customer).				
<u>Parameter</u>	<u>Value</u>												
<i>key</i>	String Attribute name (such as Revenue).												
<i>value</i>	String or key/value Attribute value (such as 500).												
<i>name</i>	String Name of the dimension (such as Customer).												
<i>auth</i>	Not used.												

_port Not used.

addStronglyTypedDimension (dimension) and addStronglyTypedDimensions (dimensions)

The `addStronglyTypedDimension` method sends a single strongly typed dimension. The `addStronglyTypedDimensions` method sends multiple strongly typed dimensions.

Note: This method is deprecated, and it is supported only for backward compatibility. In a future release, the `addDimension` and `addDimensions` methods will be enhanced, and the `addStronglyTypedDimension` and `addStronglyTypedDimensions` methods will be removed.

Input Parameters (all parameters are required unless specified as optional)

<i>dimension</i>	Document Container for dimension information.						
<i>name</i>	String Name of the dimension (such as <code>Customer</code>).						
<i>pairs</i>	Document List Container for the list of key/value dimension data.						
	<table> <thead> <tr> <th><u>Parameter</u></th> <th><u>Value</u></th> </tr> </thead> <tbody> <tr> <td><i>attribute</i></td> <td>String Attribute name (such as <code>ID</code>).</td> </tr> <tr> <td><i>value</i></td> <td>String or key/value Attribute value (such as <code>9</code>).</td> </tr> </tbody> </table>	<u>Parameter</u>	<u>Value</u>	<i>attribute</i>	String Attribute name (such as <code>ID</code>).	<i>value</i>	String or key/value Attribute value (such as <code>9</code>).
<u>Parameter</u>	<u>Value</u>						
<i>attribute</i>	String Attribute name (such as <code>ID</code>).						
<i>value</i>	String or key/value Attribute value (such as <code>9</code>).						
<i>auth</i>	Not used.						
<i>_port</i>	Not used.						

Sending Process Data

To send process data, the client uses the `processData` interface. The `processData` interface consists of the following methods:

- The [pushProcessControlOperation \(ProcessControlOperation\)](#) is an optional method that sends process state change events to the Optimize Process Tracker. The Process Tracker implicitly determines the start and stop events of a process if not they are explicitly specified. Use this method if you want to be able to pause and resume

a process or if the process name is not known when a step executes. The Process Tracker chains the operations together using the process instance ID if process control operations are used.

- The `pushStepControlOperation (StepControlOperation)` is a method that sends process step change events to the Optimize Process Tracker.
- The `pushExecutionError (ExecutionError)` method associates an error in the process with an Optimize error.

Note: These methods use the classes contained in the `bam-wsdc.jar` file in the `Optimize_directory \dataCollector\libdirectory`.

pushProcessControlOperation (ProcessControlOperation)

Pushes process state change events to the Process Tracker.

The input parameter is an instance of `ProcessControlOperation`.

Input Parameters (all parameters are required unless specified as optional)

ProcessControlOperation

Document Container for `ProcessControlOperation` instance.

businessData

Document List (Optional) Business data about the process instance.

Parameter	Value
<i>attributes</i>	Set <i>name</i> to the label for the data (such as <code>customer number</code>) and set <i>value</i> to the value of the data (such as <code>138</code>).
<i>event type</i>	String Unique identifier for event datatype in Optimize.
<i>time</i>	String Timestamp passed with event into Optimize for analysis. The time zone parameter of the date must be set to "GMT" in Designer. For input from Integration Server or Designer, the time must be in the following format: <pre>yyyy-MM-dd' T' HH:mm:ss.SSS' Z'</pre> <pre>yyyy-MM-dd' T' HH:mm:ss.SSSZ</pre> <pre>yyyy-MM-dd' T' HH:mm:ss.SSS</pre>

yyyy-MM-dd HH:mm:ss.SSS

ParentProcess InstanceId **Document List** (Optional) A unique, two-part identifier for the parent process instance to which the executing process instance belongs.

<u>Parameter</u>	<u>Value</u>
------------------	--------------

<i>keyValue</i>	Document List Set <i>name</i> to the label for the data (such as <code>customer number</code>) and set <i>value</i> to the value of the data (such as <code>138</code>).
-----------------	---

ParentProcess Name **String** (Optional) (Required when *ParentProcessInstanceId* is specified.) The name of the parent process to which the executing process belongs.

processExternalId **String** (Optional) Optimize identifier of the business process.

processInstanceDisplayName

String (Optional) A custom name for the process instance.

processInstanceId **Document List** Unique, two-part identifier for the process instance.

<u>Parameter</u>	<u>Value</u>
------------------	--------------

<i>keyValue</i>	Document List Set <i>name</i> to the first part of the identifier (such as <code>ordernumber</code>) and set <i>value</i> to the second part of the identifier (such as <code>5</code>).
-----------------	---

Optimize displays the identifier on the Process Instances page in two separate columns so you can sort by name or by value. The complete set of keys you supply constitutes the unique name Optimize uses internally for the process.

Note: ProcessInstanceID and event data share the same key value pairs.

<i>processIteration</i>	Int (Optional) Current iteration of the business process execution.
<i>processName</i>	String Name of the process. The process name must be unique if it is used to push operations to the Process Tracker.
<i>stronglyTypedBusinessData</i>	Document List (Optional) Business data about the process instance. (Deprecated. Provided only for backward compatibility. Use <i>businessData</i> instead.)

Parameter	Value
-----------	-------

<i>eventType</i>	String Unique identifier for event datatype in Optimize
<i>pairs</i>	Document List Set <i>attribute</i> to the label for the data (such as <code>customer number</code>) and set <i>value</i> to the value of the data (such as <code>138</code>).
<i>time</i>	<p>String (Optional) Timestamp passed with event into Optimize for analysis. The time zone parameter of the date must be set to "GMT" in Designer. For input from Integration Server or Designer, the time must be in the following format:</p> <pre> yyyy-MM-dd' T' HH:mm:ss.SSS' Z' yyyy-MM-dd' T' HH:mm:ss.SSSZ yyyy-MM-dd' T' HH:mm:ss.SSS yyyy-MM-dd HH:mm:ss.SSS </pre>

<i>timestamp</i>	String The time when the event occurred. The time zone parameter of the date must be set to "GMT" in Designer. The time must be in the following format:
------------------	---

```
yyyy-MM-ddTHH:mm:ss.SSSZ
```

<i>operation</i>	String Type of operation.
------------------	----------------------------------

Note: The following parameters may not appear as options in Designer. They must be entered manually.

<u>Set to...</u>	<u>To indicate that a business process has...</u>
start	Starts step. Business data can be passed with this operation.
stop	Stops step. Business data can be passed with this operation.
suspend	Paused. It is assumed that the process will resume execution at a later time. Time elapsed between suspend and resume operations will not contribute to the overall cycle time for the process. The elapsed time will not contribute to a process timeout action.
resume	Resumed execution. The timeout clock will reset.
cancel	Terminated abnormally.

Note: Process instances in the cancel or suspend state do not contribute to cycle time metrics. If a suspended process resumes and completes successfully, it will not contribute the time spent suspended to the cycle time.

pushProcessControlOperationWithCallback

Pushes process state change events to the Process Tracker.

The input parameter is an instance of ProcessControlOperation and is composed of the timeoutMS, callbackEndpoint, correlationId, and notifyOnFailureOnly parameters, in addition to the ProcessControlOperation parameters.

Input Parameters (all parameters are required unless specified as optional)

ProcessControlOperation

Document Container for ProcessControlOperation instance.

businessData

Document List (Optional) Business data about the process instance.

	<u>Parameter</u>	<u>Value</u>
	<i>attributes</i>	Set <i>name</i> to the label for the data (such as <code>customer number</code>) and set <i>value</i> to the value of the data (such as <code>138</code>).
	<i>event type</i>	String Unique identifier for event datatype in Optimize.
	<i>time</i>	String Timestamp passed with event into Optimize for analysis. The time zone parameter of the date must be set to "GMT" in Designer. For input from Integration Server or Designer, the time must be in the following format: <pre> yyyy-MM-dd'T'HH:mm:ss.SSS'Z' yyyy-MM-dd'T'HH:mm:ss.SSSZ yyyy-MM-dd'T'HH:mm:ss.SSS yyyy-MM-dd HH:mm:ss.SSS </pre>
<i>ParentProcess InstanceId</i>	Document List (Optional)	A unique, two-part identifier for the parent process instance to which the executing process instance belongs.
	<u>Parameter</u>	<u>Value</u>
	<i>keyValue</i>	Document List Set <i>name</i> to the label for the data (such as <code>customer number</code>) and set <i>value</i> to the value of the data (such as <code>138</code>).
<i>ParentProcess Name</i>	String (Optional)	(Required when <i>ParentProcessInstanceId</i> is specified.) The name of the parent process to which the executing process belongs.
<i>processInstanceDisplayName</i>	String (Optional)	A custom name for the process instance.
<i>processExternalId</i>	String (Optional)	Optimize identifier of the business process.

<i>processInstanceId</i>	Document List Unique, two-part identifier for the process instance.						
	<table> <thead> <tr> <th><u>Parameter</u></th> <th><u>Value</u></th> </tr> </thead> <tbody> <tr> <td><i>keyValue</i></td> <td> <p>Document List Set <i>name</i> to the first part of the identifier (such as <code>ordernumber</code>) and set <i>value</i> to the second part of the identifier (such as 5).</p> <p>Optimize displays the identifier on the Process Instances page in two separate columns so you can sort by name or by value. The complete set of keys you supply constitutes the unique name Optimize uses internally for the process.</p> </td> </tr> </tbody> </table>	<u>Parameter</u>	<u>Value</u>	<i>keyValue</i>	<p>Document List Set <i>name</i> to the first part of the identifier (such as <code>ordernumber</code>) and set <i>value</i> to the second part of the identifier (such as 5).</p> <p>Optimize displays the identifier on the Process Instances page in two separate columns so you can sort by name or by value. The complete set of keys you supply constitutes the unique name Optimize uses internally for the process.</p>		
<u>Parameter</u>	<u>Value</u>						
<i>keyValue</i>	<p>Document List Set <i>name</i> to the first part of the identifier (such as <code>ordernumber</code>) and set <i>value</i> to the second part of the identifier (such as 5).</p> <p>Optimize displays the identifier on the Process Instances page in two separate columns so you can sort by name or by value. The complete set of keys you supply constitutes the unique name Optimize uses internally for the process.</p>						
	<p>Note: ProcessInstanceID and event data share the same key value pairs.</p>						
<i>processIteration</i>	Int (Optional) Current iteration of the business process execution.						
<i>processName</i>	String Name of the process. The process name must be unique if it is used to push operations to the Process Tracker.						
<i>stronglyTypedBusinessData</i>	Document List (Optional) Business data about the process instance. (Deprecated. Provided only for backward compatibility. Use <i>businessData</i> instead.)						
	<table> <thead> <tr> <th><u>Parameter</u></th> <th><u>Value</u></th> </tr> </thead> <tbody> <tr> <td><i>eventType</i></td> <td>String Unique identifier for event datatype in Optimize.</td> </tr> <tr> <td><i>pairs</i></td> <td>Document List Set <i>attribute</i> to the label for the data (such as <code>customer number</code>) and set <i>value</i> to the value of the data (such as 138).</td> </tr> </tbody> </table>	<u>Parameter</u>	<u>Value</u>	<i>eventType</i>	String Unique identifier for event datatype in Optimize.	<i>pairs</i>	Document List Set <i>attribute</i> to the label for the data (such as <code>customer number</code>) and set <i>value</i> to the value of the data (such as 138).
<u>Parameter</u>	<u>Value</u>						
<i>eventType</i>	String Unique identifier for event datatype in Optimize.						
<i>pairs</i>	Document List Set <i>attribute</i> to the label for the data (such as <code>customer number</code>) and set <i>value</i> to the value of the data (such as 138).						

time **String** (Optional) Timestamp passed with event into Optimize for analysis. The time zone parameter of the date must be set to "GMT" in Designer. For input from Integration Server or Designer, the time must be in the following format:

```
yyyy-MM-dd'T'HH:mm:ss.SSS'Z'
yyyy-MM-dd'T'HH:mm:ss.SSSZ
yyyy-MM-dd'T'HH:mm:ss.SSS
yyyy-MM-dd HH:mm:ss.SSS
```

timestamp **String** The time when the event occurred. The time zone parameter of the date must be set to "GMT" in Designer. For input from Integration Server or Designer, the time must be in the following format:

```
yyyy-MM-ddTHH:mm:ss.SSSZ
```

operation **String** Type of operation.

Note: The following parameters do not appear as options in Designer. They must be entered manually.

<u>Set to...</u>	<u>To indicate that a business process has...</u>
start	Starts step. Business data can be passed with this operation.
stop	Stops step. Business data can be passed with this operation.
suspend	Paused. It is assumed that the process will resume execution at a later time. Time elapsed between suspend and resume operations will not contribute to the overall cycle time for the process. The elapsed time will not contribute to a process timeout action.
resume	Resumed execution. The timeout clock will reset.

cancel Terminated abnormally.

Note: Process instances in the cancel or suspend state do not contribute to cycle time metrics. If a suspended process resumes and completes successfully, it will not contribute the time spent suspended to the cycle time.

callbackEndpoint

String (Optional) Location of the Web service published using the EventCallback.wsdlWSDL located in the *Optimize_directory*/dataCollector/conf/glue/wsdl directory. Use the following format:

http://{host}:{port}/{path}/{serviceName}.wsdl

If the value is null, the event will be processed but the callback will not be issued.

timeoutMS

Int (Optional) Overrides the Analytic Engine's configured default event processing timeout value (currently set to 100000 milliseconds). Value must be greater than the Analytic Engine's configured default to override.

correlationId

String (Optional) Unique identifier generated by the client sent back to the Web service during the callback operation. If value is null, event will be processed but callback will not be issued.

notifyOnFailureOnly

Boolean (Optional) If true, Web service calls are issued only when a failure occurs.

pushStepControlOperation (StepControlOperation)

Pushes step state change events to the Process Tracker.

The input parameter is an instance of StepControlOperation.

Input Parameters (all parameters are required unless specified as optional)

StepControlOperation

Document Container for StepControlOperation instance.

<i>businessData</i>	<p>Document List (Optional) Business data about the process step.</p> <table border="1"> <thead> <tr> <th style="text-align: left;"><u>Parameter</u></th> <th style="text-align: left;"><u>Value</u></th> </tr> </thead> <tbody> <tr> <td><i>attributes</i></td> <td>Document List Set <i>name</i> to the label for the data (such as <code>customer number</code>) and set <i>value</i> to the value of the data (such as <code>138</code>).</td> </tr> <tr> <td><i>eventType</i></td> <td>String Unique identifier for event datatype in Optimize.</td> </tr> <tr> <td><i>time</i></td> <td> <p>String Timestamp passed with event into Optimize for analysis. The time zone parameter of the date must be set to "GMT" in Designer. For input from Integration Server or Designer, the time must be in the following format:</p> <pre>yyyy-MM-dd' T' HH:mm:ss.SSS' Z'</pre> <pre>yyyy-MM-dd' T' HH:mm:ss.SSSZ</pre> <pre>yyyy-MM-dd' T' HH:mm:ss.SSS</pre> <pre>yyyy-MM-dd HH:mm:ss.SSS</pre> </td> </tr> </tbody> </table>	<u>Parameter</u>	<u>Value</u>	<i>attributes</i>	Document List Set <i>name</i> to the label for the data (such as <code>customer number</code>) and set <i>value</i> to the value of the data (such as <code>138</code>).	<i>eventType</i>	String Unique identifier for event datatype in Optimize.	<i>time</i>	<p>String Timestamp passed with event into Optimize for analysis. The time zone parameter of the date must be set to "GMT" in Designer. For input from Integration Server or Designer, the time must be in the following format:</p> <pre>yyyy-MM-dd' T' HH:mm:ss.SSS' Z'</pre> <pre>yyyy-MM-dd' T' HH:mm:ss.SSSZ</pre> <pre>yyyy-MM-dd' T' HH:mm:ss.SSS</pre> <pre>yyyy-MM-dd HH:mm:ss.SSS</pre>
<u>Parameter</u>	<u>Value</u>								
<i>attributes</i>	Document List Set <i>name</i> to the label for the data (such as <code>customer number</code>) and set <i>value</i> to the value of the data (such as <code>138</code>).								
<i>eventType</i>	String Unique identifier for event datatype in Optimize.								
<i>time</i>	<p>String Timestamp passed with event into Optimize for analysis. The time zone parameter of the date must be set to "GMT" in Designer. For input from Integration Server or Designer, the time must be in the following format:</p> <pre>yyyy-MM-dd' T' HH:mm:ss.SSS' Z'</pre> <pre>yyyy-MM-dd' T' HH:mm:ss.SSSZ</pre> <pre>yyyy-MM-dd' T' HH:mm:ss.SSS</pre> <pre>yyyy-MM-dd HH:mm:ss.SSS</pre>								
<i>ParentProcessInstanceId</i>	<p>Document List (Optional) A unique, two-part identifier for the parent process instance to which the executing process instance belongs.</p> <table border="1"> <thead> <tr> <th style="text-align: left;"><u>Parameter</u></th> <th style="text-align: left;"><u>Value</u></th> </tr> </thead> <tbody> <tr> <td><i>keyValue</i></td> <td>Record List Set <i>name</i> to the label for the data (such as <code>customer number</code>) and set <i>value</i> to the value of the data (such as <code>138</code>).</td> </tr> </tbody> </table>	<u>Parameter</u>	<u>Value</u>	<i>keyValue</i>	Record List Set <i>name</i> to the label for the data (such as <code>customer number</code>) and set <i>value</i> to the value of the data (such as <code>138</code>).				
<u>Parameter</u>	<u>Value</u>								
<i>keyValue</i>	Record List Set <i>name</i> to the label for the data (such as <code>customer number</code>) and set <i>value</i> to the value of the data (such as <code>138</code>).								
<i>ParentProcessName</i>	String (Optional) (Required when <i>parentProcessInstanceId</i> is enabled.) The name of the parent process to which the executing process belongs.								
<i>processExternalId</i>	String (Optional) Optimize identifier of the business process.								

*processInstanceDisplayName***String** (Optional) A custom name for the process instance.*processInstanceId***Document List** Unique, two-part identifier for the process instance.

<u>Parameter</u>	<u>Value</u>
<i>keyValue</i>	<p>Document List Set <i>name</i> to the first part of the identifier (such as <i>ordernumber</i>) and set <i>value</i> to the second part of the identifier (such as 5).</p> <p>Optimize displays the identifier on the Process Instances page in two separate columns so you can sort by name or by value. The complete set of keys you supply constitutes the unique name Optimize uses internally for the process.</p>

Note: ProcessInstanceID and event data share the same key value pairs.

*processIteration***Int** (Optional) Current iteration of the business process execution.*processName***String** Name of the process. The process name must be unique if it is to be used to push operations to the Process Tracker.*stronglyTypedBusinessData***Document List** (Optional) Business data about the process step. (Deprecated. Provided only for backward compatibility. Use *businessData* instead.)

<u>Parameter</u>	<u>Value</u>
<i>eventType</i>	String Unique identifier for event datatype in Optimize.

<i>pairs</i>	Document List Set <i>attribute</i> to the label for the data (such as <code>customer number</code>) and set <i>value</i> to the value of the data (such as 138).								
<i>timeStamp</i>	String (Optional) Timestamp passed with event into Optimize for analysis. Default value is the Optimize clock time when the event was received. Must match <i>timestamp</i> below.								
<i>timestamp</i>	String The time when the event occurred. The time zone parameter of the date must be set to "GMT" in Designer. For input from Integration Server or Designer, the time must be in the following format: <code>yyyy-MM-ddTHH:mm:ss.SSSZ</code>								
<i>operation</i>	String Type of operation.								
	<table border="0" style="width: 100%;"> <thead> <tr> <th style="text-align: left;"><u>Set to...</u></th> <th style="text-align: left;"><u>To indicate that a business process has...</u></th> </tr> </thead> <tbody> <tr> <td style="vertical-align: top;"><code>start</code></td> <td>Starts step. Business data can be passed with this operation.</td> </tr> <tr> <td style="vertical-align: top;"><code>stop</code></td> <td>Stops step. Business data can be passed with this operation.</td> </tr> <tr> <td style="vertical-align: top;"><code>log_ business_data</code></td> <td>Sends business data. Does not affect execution or state of the step. This operation needs to have a timestamp that is greater than the start operation timestamp and less than the stop operation timestamp in order for dimensional intrinsic metrics to be captured properly. Business data still will be captured, just without the additional intrinsic metrics (if configured).</td> </tr> </tbody> </table>	<u>Set to...</u>	<u>To indicate that a business process has...</u>	<code>start</code>	Starts step. Business data can be passed with this operation.	<code>stop</code>	Stops step. Business data can be passed with this operation.	<code>log_ business_data</code>	Sends business data. Does not affect execution or state of the step. This operation needs to have a timestamp that is greater than the start operation timestamp and less than the stop operation timestamp in order for dimensional intrinsic metrics to be captured properly. Business data still will be captured, just without the additional intrinsic metrics (if configured).
<u>Set to...</u>	<u>To indicate that a business process has...</u>								
<code>start</code>	Starts step. Business data can be passed with this operation.								
<code>stop</code>	Stops step. Business data can be passed with this operation.								
<code>log_ business_data</code>	Sends business data. Does not affect execution or state of the step. This operation needs to have a timestamp that is greater than the start operation timestamp and less than the stop operation timestamp in order for dimensional intrinsic metrics to be captured properly. Business data still will be captured, just without the additional intrinsic metrics (if configured).								

<i>stepExternalId</i>	String (Optional) Optimize identifier of the process step.
<i>stepIteration</i>	Int (Optional) Current iteration of the business process execution.
<i>stepName</i>	String Name of the step. The name of the step must match the step name specified in the process model. If two processes share the same name, neither can be tracked.
<i>isGlobalBusinessData</i>	String (True or False) Indicates whether the data being passed in by the server refers to all future steps or only to specified steps in the process.

pushStepControlOperationWithCallback

Pushes step state change events to the Process Tracker.

The input parameter is an instance of StepControlOperation and is composed of the timeoutMS, callbackEndpoint, correlationId, and notifyOnFailureOnly parameters, in addition to the StepControlOperation parameters.

Input Parameters (all parameters are required unless specified as optional)

StepControlOperation

Document Container for ProcessControlOperation instance. **String**

businessData **Document List** (Optional) Business data about the process step.

<u>Parameter</u>	<u>Value</u>
<i>attributes</i>	Document List Set <i>name</i> to the label for the data (such as <code>customer number</code>) and set <i>value</i> to the value of the data (such as <code>138</code>).
<i>eventType</i>	String Unique identifier for event data type in Optimize.

<i>time</i>	<p>String Timestamp passed with event into Optimize for analysis. The time zone parameter of the date must be set to "GMT" in Designer. For input from Integration Server or Designer, the time must be in the following format:</p> <pre> yyyy-MM-dd' T' HH:mm:ss.SSS' Z' yyyy-MM-dd' T' HH:mm:ss.SSSZ yyyy-MM-dd' T' HH:mm:ss.SSS yyyy-MM-dd HH:mm:ss.SSS </pre>				
<i>ParentProcessInstanceId</i>	<p>Document List (Optional) A unique, two-part identifier for the parent process instance to which the executing process instance belongs.</p>				
	<table border="0"> <thead> <tr> <th style="text-align: left;"><u>Parameter</u></th> <th style="text-align: left;"><u>Value</u></th> </tr> </thead> <tbody> <tr> <td style="vertical-align: top;"><i>keyValue</i></td> <td> <p>Record List Set <i>name</i> to the label for the data (such as customer number) and set <i>value</i> to the value of the data (such as 138).</p> </td> </tr> </tbody> </table>	<u>Parameter</u>	<u>Value</u>	<i>keyValue</i>	<p>Record List Set <i>name</i> to the label for the data (such as customer number) and set <i>value</i> to the value of the data (such as 138).</p>
<u>Parameter</u>	<u>Value</u>				
<i>keyValue</i>	<p>Record List Set <i>name</i> to the label for the data (such as customer number) and set <i>value</i> to the value of the data (such as 138).</p>				
<i>ParentProcessName</i>	<p>String (Optional) (Required when <i>parentProcessInstanceId</i> is enabled.) The name of the parent process to which the executing process belongs.</p>				
<i>processExternalId</i>	<p>String (Optional) Optimize identifier of the business process.</p>				
<i>processInstanceDisplayName</i>	<p>String (Optional) A custom name for the process instance.</p>				
<i>processInstanceId</i>	<p>Document List Unique, two-part identifier for the process instance.</p>				
	<table border="0"> <thead> <tr> <th style="text-align: left;"><u>Parameter</u></th> <th style="text-align: left;"><u>Value</u></th> </tr> </thead> <tbody> <tr> <td style="vertical-align: top;"><i>keyValue</i></td> <td> <p>Document List Set <i>name</i> to the first part of the identifier</p> </td> </tr> </tbody> </table>	<u>Parameter</u>	<u>Value</u>	<i>keyValue</i>	<p>Document List Set <i>name</i> to the first part of the identifier</p>
<u>Parameter</u>	<u>Value</u>				
<i>keyValue</i>	<p>Document List Set <i>name</i> to the first part of the identifier</p>				

(such as `ordernumber`) and set *value* to the second part of the identifier (such as 5).

Optimize displays the identifier on the Process Instances page in two separate columns so you can sort by name or by value. The complete set of keys you supply constitutes the unique name Optimize uses internally for the process.

ProcessInstanceID and event data share the same key value pairs.

processIteration

Int (Optional) Current iteration of the business process execution.

processName

String Name of the process. The process name must be unique if it is to be used to push operations to the Process Tracker.

stronglyTypedBusinessData

Document List (Optional) Business data about the process step. (Deprecated. Provided only for backward compatibility. Use *businessData* instead.)

<u>Parameter</u>	<u>Value</u>
<i>eventType</i>	String Unique identifier for event data type in Optimize.
<i>pairs</i>	Document List Set <i>attribute</i> to the label for the data (such as <code>customer number</code>) and set <i>value</i> to the value of the data (such as 138).
<i>timeStamp</i>	String (Optional) Timestamp passed with event into Optimize for analysis. Default value is the Optimize clock time when the event

was received. Must match *timestamp* below.

timestamp

String The time when the event occurred. The time zone parameter of the date must be set to "GMT" in Designer. For input from Integration Server or Designer, the time must be in the following format:

yyyy-MM-ddTHH:mm:ss.SSSZ

operation

String Type of operation.

Set to...

To indicate that a business process has...

start

Starts step. Business data can be passed with this operation.

stop

Stops step. Business data can be passed with this operation.

log_business_data Sends business data. Does not affect execution or state of the step. This operation needs to have a timestamp that is greater than the start operation timestamp and less than the stop operation timestamp in order for dimensional intrinsic metrics to be captured properly. Business data still will be captured, just without the additional intrinsic metrics (if configured).

stepExternalId

String (Optional) Optimize identifier of the process step.

stepIteration

Int (Optional) Current iteration of the business process execution.

stepName

String Name of the step. The name of the step must match the step name specified in the process

	model. If two processes share the same name, neither can be tracked.
<i>callbackEndpoint</i>	<p>String (Optional) Location of the Web service published using the EventCallback.wsdlWSDL located in the <i>Optimize_directory</i> /dataCollector/conf/glue/wsdl directory. Use the following format:</p> <p>http://{host}:{port}/{path}/{serviceName}.wsdl</p> <p>If the value is null, the event will be processed but the callback will not be issued.</p>
<i>timeoutMS</i>	<p>Int (Optional) Overrides the Analytic Engine's configured default event processing timeout value (currently set to 100000 milliseconds). Value must be greater than the Analytic Engine's configured default to override.</p>
<i>correlationId</i>	<p>String (Optional) Unique identifier generated by the client sent back to the Web service during the callback operation. If value is null, event will be processed but callback will not be issued.</p>
<i>notifyOnFailureOnly</i>	<p>Boolean (Optional) If true, Web service calls are issued only when a failure occurs.</p>

pushExecutionError (ExecutionError)

Pushes data about a process-related error.

The input parameter is an instance of ExecutionError.

Input Parameters (all parameters are required unless specified as optional)

ExecutionError

Document Container for ExecutionError instance.

message **String** A short description of the error.

messageDetail **String** (Optional) A long description of the error. If a stack trace is available, store it in this field.

processInstanceId **Document** Unique, two-part identifier for the process step.

Parameter	Value
<i>keyValue</i>	Document List Set <i>name</i> to the first part of the identifier (such as <code>ordernumber</code>) and set <i>value</i> to the second part of the identifier (such as <code>5</code>). Optimize displays the identifier on the Process Instances page in two separate columns so you can sort by name or by value. The complete set of keys you supply constitutes the unique name Optimize uses internally for the process.

Note: ProcessInstanceID and event data share the same key value pairs.

processIteration **Int** (Optional) Current iteration of the business process execution.

processName **String** Name of the process. The process name must be unique if it is to be used to push operations to the Process Tracker.

stepName **String** (Optional) Name of the process step. The process step name must match the process step name specified in the model definition for this process. For information about configuring process models, see "[Configuring Business Processes](#)" on page 99

timestamp **String** The time when the event occurred. The time zone parameter of the date must be set to "GMT" in Designer. For input from Integration Server or Designer, the time must be in the following format:

yyyy-MM-ddTHH:mm:ss.SSSZ

type **String** The type of error that occurred.

pushExecutionErrorWithCallback

Pushes data about a process-related error.

The input parameter is an instance of ExecutionError.

Input Parameters (all parameters are required unless specified as optional)*ExecutionError*

Document Container for ExecutionError instance.

message **String** A short description of the error.

messageDetail **String (Optional)** A long description of the error. If a stack trace is available, store it in this field.

processInstanceId **Document** Unique, two-part identifier for the process step.

Parameter	Value
<i>keyValue</i>	Document List Set <i>name</i> to the first part of the identifier (such as <code>ordernumber</code>) and set <i>value</i> to the second part of the identifier (such as <code>5</code>). Optimize displays the identifier on the Process Instances page in two separate columns so you can sort by name or by value. The complete set of keys you supply constitutes the unique name Optimize uses internally for the process.

Note: ProcessInstanceID and event data share the same key value pairs.

processIteration **Int (Optional)** Current iteration of the business process execution.

processName **String** Name of the process. The process name must be unique if it is to be used to push operations to the Process Tracker.

stepName **String (Optional)** Name of the process step. The process step name must match the process step name specified in the model definition for this process. For information about configuring process models, see "[Configuring Business Processes](#)" on page 99

timestamp **String** The time when the event occurred. The time zone parameter of the date must be set to "GMT" in Designer.

For input from Integration Server or Designer, the time must be in the following format:

yyyy-MM-ddTHH:mm:ss.SSSZ

<i>type</i>	String The type of error that occurred.
<i>timeoutMS</i>	Int (Optional) Overrides the Analytic Engine's configured default event processing timeout value (currently set to 100000 milliseconds). Value must be greater than the Analytic Engine's configured default to override.
<i>callbackEndpoint</i>	String (Optional) Location of the Web service published using the EventCallback.wsdlWSDL located in the <i>Optimize_directory</i> /dataCollector/conf/glue/wsdl directory. Use the following format: http://{host}:{port}/{path}/{serviceName}.wsdl If the value is null, the event will be processed but the callback will not be issued.
<i>correlationId</i>	String (Optional) Unique identifier generated by the client sent back to the Web service during the callback operation. If value is null, event will be processed but callback will not be issued.
<i>notifyOnFailureOnly</i>	Boolean (Optional) If true, Web service calls are issued only when a failure occurs.

12 Setting Up Web Service Actions

■ Overview	282
■ Configuring Web Service Actions	282

Overview

When a rule is violated, you can have Optimize trigger a Web service action. For example, suppose that you defined a rule to determine when an adapter goes offline. Also suppose that the Integration Server on which the adapter is running contains a service that attempts to restart adapters. You can define an action that, when the adapter rule is violated, invokes a Web service that executes the Integration Server service to restart the adapter.

You can also provide optional authentication parameters for the web service action by specifying a user login and encrypted password.

Note: The web service actions described herein are supported only on RPC style web services.

Configuring Web Service Actions

To configure Web service actions

1. In My webMethods: **Navigate > Applications > Administration > System-Wide > Environments > Define Environments.**

The Define Environments page is displayed.

2. Click the name of the configured environment for which you want to configure Web service actions.

The Edit Environment page is displayed.

3. Click the **Configure Servers** tab.

4. On the **CONFIGURATION** tree, find the Analytic Engine for which you want to configure Web service actions, and click **WS Action Settings** under the name of the Analytic Engine.

The WS Action Settings page is displayed.

5. Identify the Web service to execute, the location of its WSDL document, and the parameters the service will pass, by uncommenting and editing this XML displayed in the text box:

```
<properties>
<!-- (Remove this line to uncomment the file.)
<property name="action">
  <string meta="name">{Test Action}</string>
  <string meta="url">{host:port/path/service.wsdl}</string>
  <string meta="method">{operation}</string>
  <list>
    [Place parameter name and types here]
    <element><string>{paramName1}</string></element>
    <element><string>{paramName2}</string></element>
    <element><string>{paramName...}</string></element>
  </list>
</property name="login">
```

```

    <string meta="user">{username}</string>
    [Must use a unique handle for each different password
    which is used for encryption]
    <string meta="handle">{passwordHandle}</string>
    <string meta="password">{password}</string>
  </property>
</property>
--> (Remove this line to uncomment the file.)
</properties>

```

Table A: Action Variables

In section A of the XML sample, replace the text in braces{} with the appropriate text as described.

Variable	Description
<i>Test Action</i>	Name: A unique name to identify the action.
<i>host:port/path/ service.wsdl</i>	URL + wsdl: The hostname or IP address and port number of the system on which the Web service will be executed, and the location and name of the WSDL document.
<i>operation</i>	Method: The name of the method to invoke. This name should match the method name provided in the WSDL document.

Table B: Action Parameter Attributes

The web service method called must have a signature matching the parameters listed below. Place these parameters in Section B of the XML code sample.

To specify parameters for displaying rule information and other attributes in your Web service actions, replace the <element><string> lines with one or more of the following:

```

<element><string>RuleName</string></element>
<element><string>RuleDefinition</string></element>
<element><string>RuleEvaluation</string></element>
<element><string>RuleSla</string></element>
<element><string>RuleCustomer</string></element>
<element><string>RuleSeverity</string></element>
<element><string>RuleStatus</string></element>
<element><string>Attributes</string></element>
<element><string>Time</string></element>

```

Literal Attribute	Description
-------------------	-------------

Enter the names of these attributes literally to include this information in the action.

RuleName	String containing of the name of the base rule, an instance of which has fired: Use this text literally to represent the string containing the name of the base rule, an instance of which was violated.
RuleInstanceName	String containing the name of the rule instance that fired: Use this text literally to represent the string containing the name of the rule instance that was violated.
RuleDefinition	String containing the definition of the rule: Use this text literally to represent the string containing the definition of the rule.
RuleEvaluation	String containing the evaluation of the rule: Use this text literally to represent the string containing the evaluation of the rule.
RuleSla	String containing the Service Level Agreement (SLA) associated with the rule: Use this text literally to represent the string containing the SLA status (Yes or No) of the rule.
RuleCustomer	String containing the name of a customer associated with the rule: Use this text literally to represent the string containing the name of a customer associated with the rule.
RuleSeverity	String containing the severity of the rule: Use this text literally to represent the string containing any severity label assigned to this rule.
RuleStatus	String containing the compliance status of the rule. Use this text literally to represent the compliance status (Rule Violation or In Compliance) of the rule.
Attributes	Array of strings containing key = value pairs of all attributes in the rule diagnosis: Use this text literally to represent an array of strings containing key/value pairs of all attributes in the rule diagnosis.
Time	String form of the time that the rule went out of compliance: Use this text literally to represent the time that the rule was violated, in string format.
ProcessInfo	Array of strings containing information about a process. Use this text literally to represent an array of strings

containing information about a process, such as process name, step names, and instance IDs.

Additional Attributes

Enter additional attributes by specifying monitor data field names.

paramName Allows you to specify a particular field from the underlying data of the monitor. For example, if the rule is on business data containing a Process InstanceId field, you would specify **ProcessInstanceId**.

Table C: Optional Authentication Parameters

Use these parameters in Part C of the XML code sample when calling a secure web service. Supply the following parameters under the “login” node:

Variable	Description
<i>username</i>	Specify a user name. Must be accompanied by a password.
<i>passwordHandle</i>	Handle that identifies the password specified in the password string. Each instance of a web service action must use a unique password handle. By default, WSAActionHandle automatically generates unique password handles for each web service action instance. Though you can manually define password handles, they will be overwritten when the configuration is saved.
<i>password</i>	Specify a password. Must be accompanied by a user name. Once this settings file is saved and processed, the password is encrypted, and is displayed only as asterisks.

Optimize displays the Web service action you identified in *actionName* on the Edit Rule and Rule Details pages. For more information about selecting a Web service action on this page, see the *webMethods Optimize User's Guide*.

6. If you want to identify more than one Web Service action, copy and edit the XML shown above for each action.
7. Click **Save**.
8. Apply the changes by deploying your updated environment (see "[Changing Your Configuration Settings](#)" on page 57 and restarting the Analytic Engine as described in "[Starting Optimize Manually](#)" on page 40).
9. Make sure the new settings work, as follows:

- a. In My webMethods **Navigate > Applications > Administration > Analytics > Rules > Rule List**.
- b. On the Rule List page, click **Create Rule**.
- c. In the **Actions** area on the Edit Rule page, click **Add Action**.
- d. Ensure that the **Action Name** list contains the Web Service actions you added to WSAActionConfiguration.properties in this procedure.

If the **Action Name** list does not contain the Web Service actions you identified, check the syntax of the XML in the WS Action Settings page.

13 Optimize REST Services API

■ Overview	288
■ Running the Restful Services using localhost	290
■ The Optimize Restful Services	290
■ Process Model Rest API Services	290
■ Process Analytics REST API	295

Overview

The Optimize Restful Services API contains a variety of services that enable users to query process model and process analytics data. With these services, users can extract Optimize data for use with reports and other applications. Users should be aware that use of these services can affect Optimize performance, as many of these services perform database queries. It is recommended that all implementations be tested in a production-like environment prior to release in order to understand and evaluate potential performance impacts.

Note: Data Level Security (DLS) is not currently supported for this API.

There are two components to this API as described below.

- Process Model REST API provides all the processes configured for tracking
- Process Analytics REST API provides analytic data for processes.

The API returns either XML or JSON results based on request headers. See the following table.

type	HTTP Header "Content-Type"	HTTP Header "Accept"
json	application/json	application/json
xml	application/xml	application/xml

The order of elements in the response (XML or JSON) may or may not be predefined; however, the order is subject to change and should not be relied upon when consuming the response.

Time Parameters

The input time format should comply to either ISO 8601 or Java epoch (long). See the following examples:

- ISO 8601 option 1: yyyy-MM-dd'T'HH:mm:ssZ
- ISO 8601 option 2 (with timezone offset): yyyy-MM-dd'T'HH:mm:ss-06:00
- ISO 8601 option 3 (without timezone offset): yyyy-MM-dd'T'HH:mm:ss
- Java epoch (example): 1407373267000

Max Results Parameter

When providing the max results parameter value, 0 or a negative number returns unlimited results, which can have a significant performance impact. Users should

exercise caution when configuring implementations that allow unlimited results. If no max results parameter is specified, the parameter defaults to 5000.

HTTP Request Errors

Errors encountered in the request format or during its execution are returned as an HTTP response. Errors in the request format or in the content of the request parameters are caught during request validation and returned in a simple response. Unexpected errors encountered during execution are returned with more detail. Unexpected errors are also captured in the Analytic Engine log, while validation errors are not captured.

Error Response Structure

```
<error>
  <errorMsg>error message</errorMsg>
  <statusCode>HTTP status code</statusCode>
  <trace>
    stack trace if applicable
  </trace>
</error>
```

Error Response Example

```
<error>
  <errorMsg>missing parameter [startTime]</errorMsg>
  <statusCode>400</statusCode>
  <trace/>
</error>
```

Statistical Results

Statistical results, such as historical cycle time average, are calculated over time for a specific statistical interval. The interval is a period of time where by all historical data that falls with the time period is used in the calculation. For example, a 1-DAY interval may calculate all historical data for all Mondays, depending upon the user's configuration of the Optimize Analytic Engine. See the "Configuring Optimize" chapter of the *Administering webMethods Optimize* guide for additional information on statistical intervals.

The Optimize REST API service's statistical interval is determined by the service relative to the input time range parameters. Some services provide an optional input parameter for the statistical interval so the user can choose an interval that is more suitable to their use case. If the service does not provide a statistical interval input parameter, then the service will use a 1-DAY interval for the day relative to the given time range's end time. For example, if the input time range ends on Tuesday, the Tuesday statistical interval will be used. If the user does not include the optional parameter, then a suitable interval will be determined based on the size of the time range. The following chart identifies the intervals automatically chosen base on the size of the time range.

Time Range	Statistical Interval
<= 12 hours	1 hour

Time Range	Statistical Interval
<= 2 days	4 hours
<= 2 weeks	1 day
<= 4 weeks	1 week
> 4 weeks	4 weeks

Running the Restful Services using localhost

If you want to consume Restful API services using a localhost configuration, you will need to change the default binding behavior configuration for your system. Note that this configuration may have negative side effects on multi-homed systems.

Note: If you use the machine name rather than localhost in your configuration, you can ignore this procedure.

To configure your system for consuming Restful services using localhost

1. Open the wrapper.conf file for the Analytic Engine using a text editor. The file is located in the [installation directory]\optimize\analysis\confdirectory.
2. Locate the following line in the file: `wrapper.java.additional.19=-Dopt.wm.glue.ignoreLocalHostCheckOnBinding=false`
3. Change the value from “false” to “true”
4. Save the wrapper.conf file and close it.
5. Restart the Analytic Engine to implement your changes.

The Optimize Restful Services

The remainder of this chapter explains the details of the Process Model and Process Analytics Rest API services.

Process Model Rest API Services

This section lists and explains the services within the Process Model section of the Optimize REST Services API. This API provides definition information for processes, steps and stages.

Process Model List

This service returns a list of process models that are deployed to the Optimize Analytic Engine. Each process model includes process definition information.

The URL format is as follows:

```
http://${optimize host}:${port}/services/rest/api/ProcessModels/  
Processes
```

Input Parameters

None

Output Parameters

<i>processModelId</i>	Returns the identifier of the process model.
<i>modelName</i>	Returns the name given to the process model.
<i>processPath</i>	Returns the process model path.
<i>description</i>	Returns the description provided for the model.
<i>modelVersion</i>	Returns the version of the model. Values are "BAM" or "BPM".
<i>modelType</i>	Returns the process model type, either "BAM" or "BPM".
<i>createdBy</i>	Returns the name of the process model author.
<i>deploymentVersion</i>	Returns the descriptive version of the process model.
<i>deploymentTime</i>	Returns the deployment time as an ISO 8601 date.
<i>deploymentTimeAsJavaEpoch</i>	Returns the deployment time as Java epoch.
<i>executionEnabled</i>	Returns "TRUE" or "FALSE" to indicate whether the process model execution is enabled.

<i>instanceLogging</i>	Returns "TRUE" or "FALSE" to indicate whether instance logging is enabled.
<i>runningInstancesExist</i>	Returns "TRUE" or "FALSE" to indicate whether running process models exist.
<i>thresholdNumber</i>	Returns the number used to notify the user when data archiving should be performed.
<i>trackingEnabled</i>	Returns "TRUE" or "FALSE" to indicate whether process tracking is enabled.

Process Models Result Example

```
<processModels>
  <processModel>
    <processModelId>MyModels/Order Fulfillment</processModelId>
    <modelName>Order Fulfillment</modelName>
    <processPath>MyModels</processPath>
    <description>Process for order fulfillment</description>
    <modelVersion>Production</modelVersion>
    <modelType>BAM</modelType>
    <createdBy>john doe</createdBy>
    <deploymentVersion>1</deploymentVersion>
    <deploymentTime>2014-08-05T15:33:06:00</deploymentTime>
    <deploymentTimeAsJavaEpoch>1407274405067</deploymentTimeAsJavaEpoch>
    <executionEnabled>>false</executionEnabled>
    <instanceLogging>>false</instanceLogging>
    <runningInstancesExist>>false</runningInstancesExist>
    <thresholdNumber>1</thresholdNumber>
    <trackingEnabled>>true</trackingEnabled>
  </processModel>
</processModels>
```

Process Steps

This service returns a list of process steps for a specified process model. Each step includes process definition information.

The URL format is as follows:

```
http://${optimize host}:${port}/services/rest/api/ProcessModels/Steps?
processModelId=${processModelId}&version=${version}
```

Input Parameters

<i>processmodelId</i>	String. The identifier of the process model. This attribute is required.
-----------------------	---

version **Integer.** The deployment version of the process model. This attribute is required.

Output Parameters

stepId Returns the process model step identifier.

stepName Returns the process model step name.

Step Models Result Example

```
<stepModels>
  <stepModel>
    <stepId>S3</stepId>
    <stepName>Order Receipt</stepName>
  </stepModel>
  <stepModel>
    <stepId>S6</stepId>
    <stepName>Find Customer Info</stepName>
  </stepModel>
  <stepModel>
    <stepId>S14</stepId>
    <stepName>Validate Credit</stepName>
  </stepModel>
  <stepModel>
    <stepId>S20</stepId>
    <stepName>Book Shipment</stepName>
  </stepModel>
  <stepModel>
    <stepId>S17</stepId>
    <stepName>Accept Order</stepName>
  </StepModel>
  <stepModel>
    <stepId>S27</stepId>
    <stepName>Validate Inventory</stepName>
  </stepModel>
</stepModels>
```

Process Stages

This service returns a list of process stages for a specified process. Each stage includes definition information.

The URL format is as follows:

```
http://${optimize host}:${port}/services/rest/api/ProcessModels/Stages?
processModelId=${processModelId}&version=${version}
```

Input Parameters

<i>processmodelId</i>	String. The identifier of the process model. This attribute is required.
<i>version</i>	Integer. The deployment version of the process model. This attribute is required.

Output Parameters

<i>stageId</i>	Returns the process model stage identifier.
<i>stageName</i>	Returns the process model stage name.
<i>stageDescription</i>	Returns the process model stage description.
<i>cancelOnBreach</i>	Returns "TRUE" or "FALSE" to indicate whether a breach cancels process instance tracking.
<i>stageCondition</i>	Returns the stage condition. Values are "LESS_THAN" or "GREATER_THAN".
<i>conditionDuration</i>	Returns the stage condition duration.
<i>endMilestone</i>	Returns the end milestone name.
<i>endMilestoneStepName</i>	Returns the end milestone step name.
<i>startMilestone</i>	Returns the start milestone name.
<i>startMilestoneStepName</i>	Returns the start milestone step name.

Stage Models Result Example

```
<stageModels>
  <stageModel>
    <stageId>1fa2ac94-169e-40a8-967a-7c07e98eb8e5</stageId>
    <stageName>ValidateCreditComplete-EndOfTheProcess</stageName>
    <stageDescription>Validate Credit Complete</stageDescription>
    <cancelOnBreach>>false</cancelOnBreach>
    <stageCondition>LESS_THAN</stageCondition>
    <conditionDuration>5000</conditionDuration>
  </stageModel>
</stageModels>
```

```

    <endMilestone>PROCESS.0.COMPLETED</endMilestone>
    <endMilestoneStepName>End of process</endMilestoneStepName>
    <startMilestone>STEP.S14.COMPLETED</startMilestone>
    <startMilestoneStepName>Validate Credit (S14).Complete</startMilestoneStepName>
  </stageModel>
</stageModels>

```

Process Analytics REST API

This section lists and explains the services within the Process Analytics section of the Optimize REST Services API. These services enable users to access process related analytic data that is produced by the Optimize Analytic Engine.

Process Metric Data

This service returns various aggregated process metric data for a specified process and time range. The data is derived from the process instances that were active during the time range. Cycle time statistical calculations can be derived from either the optional statsInterval parameter or from the size of the given range as shown below.

Time Range	Statistical Interval
<= 12 hours	1 hour
<= 2 days	4 hours
<= 2 weeks	1 day
<= 4 weeks	1 week
> 4 weeks	4 weeks

The URL format is as follows:

```

http://${optimize host}:${port}/services/rest/api/ProcessAnalytics/
ProcessMetrics?processModelId=${processModelId}&startTime=
${startTime}&endTime=${endTime}&statsInterval=${statsInterval}

```

Input Parameters

processmodelId **String.** The identifier of the process model. This attribute is required.

<i>startTime</i>	Integer or String . The query start time (Java long or ISO 8601). This attribute is required.
<i>endTime</i>	Integer or String . The query end time (Java long or ISO 8601). This attribute is required.
<i>statsInterval</i>	String . The statistics interval size. Valid values are "1HOUR", "4HOUR", "1DAY", "1WEEK", "4WEEK". This attribute is not required (if not present, the default is calculated from the time range).

Output Parameters

<i>cycleTimeAverage</i>	Returns the cycle time average, in milliseconds, for all process instances that completed during the time range.
<i>cycleTimeHistoricalAverage</i>	Returns the cycle time historical average for the relative statistics interval in milliseconds. This is the statistical calculation for the stats interval in which the time range ends.
<i>cycleTimeMax</i>	Returns the maximum cycle time for all process instances that completed during the time range in milliseconds.
<i>cycleTimeMin</i>	Returns the minimum cycle time for all process instances that completed during the time range in milliseconds.
<i>cycleTimeStandardDeviation</i>	Returns the cycle time standard deviation for the relative stats interval in milliseconds. This is the statistical standard deviation for the stats interval in which the time range ends.
<i>endTime</i>	Returns the given range time as ISO 8601.
<i>endTimeAsJavaEpoch</i>	Returns the range end time as Java epoch.
<i>ruleViolationCount</i>	Returns the total rule violations count for the specified process model/time range.

<i>ruleViolationsOpen</i>	Returns the count of open rule violations for the specified process model/time range.
<i>startTime</i>	Returns the range start time as ISO 8601.
<i>startTimeAsJavaEpoch</i>	Returns the range start time as Java epoch.
<i>volumeCompleted</i>	Returns the count of processes completed during the time range.
<i>volumeInProgress</i>	Returns the count of processes in progress at the end of the time range.
<i>volumeStarted</i>	Returns the count of processes started during the time range.

Process Metrics Result Example

```
<processMetrics>
  <cycleTimeAverage>0.18</cycleTimeAverage>
  <cycleTimeHistoricalAverage>0.2</cycleTimeHistoricalAverage>
  <cycleTimeMax>1200000</cycleTimeMax>
  <cycleTimeMin>650000</cycleTimeMin>
  <cycleTimeStandardDeviation>0.2</cycleTimeStandardDeviation>
  <endTime>2014-08-01T09:46-06:00</endTime>
  <endTimeAsJavaEpoch>1406907960000</endTimeAsJavaEpoch>
  <ruleViolationCount>2</ruleViolationCount>
  <ruleViolationsOpen>1</ruleViolationsOpen>
  <startTime>2014-05-01T09:46-06:00</startTime>
  <startTimeAsJavaEpoch>1398959160000</startTimeAsJavaEpoch>
  <volumeCompleted>5</volumeCompleted>
  <volumeInProgress>1</volumeInProgress>
  <volumeStarted>1</volumeStarted>
</processMetrics>
```

Process Instances

This service returns a list of process instances for a specified process and time range. Optionally, a status filter can be specified as well as a maximum results limiter. If the results are truncated by the maxResults limiter, the most recent instances will returned.

The URL format is as follows:

```
http://${optimize host}:${port}/services/rest/api/ProcessAnalytics/
ProcessInstances?processModelId=${processModelId}&startTime=
${startTime}&endTime=${endTime}&status=${status}&maxResults=
${maxResults}
```

Example: `http://localhost:12503/services/rest/api/ProcessAnalytics/ProcessInstances?processModelId=OrderFulfillment&startTime=1407996000000&endTime=1408049935760&status=ST`

Input Parameters

<i>processmodelId</i>	String. The identifier of the process model. This attribute is required.
<i>startTime</i>	Integer or String. The query start time (Java long or ISO 8601). This attribute is required.
<i>endTime</i>	Integer or String. The query end time (Java long or ISO 8601). This attribute is required.
<i>status</i>	String. The process status by which to filter. Valid values are "ALL", "STARTED", "RUNNING", "COMPLETED". This attribute is not required (defaults to "ALL").
<i>maxResults</i>	String. The maximum number of results. If unlimited results are returned, use 0 or negative number. This parameter is not required (defaults to 5000).

Output Parameters

<i>processInstanceId</i>	Returns the identifier for the process instance.
<i>displayName</i>	Returns the custom identifier of the process instance, if such exists.
<i>processModelId</i>	Returns the identifier for the process model.
<i>deploymentVersion</i>	Returns the deployment version of the process.
<i>processModelType</i>	Returns the process model type, either "BPM" or "BAM". This indicates whether the process execution was orchestrated by Process Engine or not.
<i>cycleTime</i>	Returns the process instance cycle time in milliseconds or -1 if no cycle time is available.
<i>instanceIteration</i>	Returns the iteration of the process instance.

<i>trackingStatus</i>	Returns the tracking status of the process instance. Valid values are "STARTED", "RUNNING", "COMPLETED", "SUSPENDED", "RESUMED", "CANCELLED", "UNKNOWN".
<i>processInstanceSeverity</i>	Returns the process instance severity. Valid values are "NONE", "NO_DATA", "WARNING", "ERROR", "CRITICAL", "INFORMATIONAL".
<i>startTime</i>	Returns the process instance start time as ISO 8601.
<i>startTimeAsJavaEpoch</i>	Returns the process instance start time as Java epoch.
<i>endTime</i>	Returns the process instance end time as ISO 8601 or returns empty if no end time is available.
<i>endTimeAsJavaEpoch</i>	Returns the process instance end time as Java epoch or returns -1 if no end time is available.

Process Instances Result Example

```
<results>
  <processInstances>
    <processInstance xmlns:xsi="http://www.w3.org/2001/
      XMLSchema-instance" xsi:type="processInstance">
      <processInstanceId>393x3-389x-ey393-56ez</processInstanceId>
      <processModelId>Order Fulfillment</processModelId>
      <processModelType>BAM</processModelType>
      <deploymentVersion>1</deploymentVersion>
      <cycleTime>0.2</cycleTime>
      <instanceIteration>1</instanceIteration>
      <trackingStatus>RUNNING</trackingStatus>
      <processInstanceSeverity>NONE</processInstanceSeverity>
      <startTime>2014-05-01T09:46-06:00</startTime>
      <startTimeAsJavaEpoch>1398959160000</startTimeAsJavaEpoch>
      <endTime>16006368-12-25T08:50-0700</endTime>
      <endTimeAsJavaEpoch>5050500505050505050</endTimeAsJavaEpoch>
    </processInstance>
  </processInstances>
  <totalCount>1</totalCount>
</results>
```

Process Tracking Errors

Returns a count of process tracking errors that occurred for a specified process and time range. Optimize detects tracking errors during runtime analysis. Tracking errors can be built-in, such as `stepOutOfSequence` or user defined such as a process timeout.

The URL format is as follows:

```
http://${optimize host}:${port}/services/rest/api/ProcessAnalytics/
ProcessErrors?processModelId=${processModelId}&startTime=
${startTime}&endTime=${endTime}
```

Input Parameters

<i>processmodelId</i>	String. The external process model identifier. This attribute is required.
<i>startTime</i>	Integer or String. The query start time (Java long or ISO 8601). This attribute is required.
<i>endTime</i>	Integer or String. The query end time (Java long or ISO 8601). This attribute is required.

Output Parameters

<i>errorTypes</i>	Returns the process error type.
<i>errorCount</i>	Returns the number of error occurrences.

Process Tracking Errors Result Example

```
<processErrors>
  <processError>
    <errorType>Timeout</errorType>
    <errorCount>22</errorCount>
  </processError>
</processErrors>
```

Step Instances

This service returns step instance data for a specified process, step and time range. Results can be filtered by the status of the step instance, such as completed or running. The step instance includes execution data and tracking data as well as step definition data.

The URL format is as follows:

```
http://${optimize host}:${port}/services/rest/api/ProcessAnalytics/
StepInstances?processModelId=${processModelId}&stepId=
${stepId}&startTime=${startTime}&endTime=${endTime}&status=
${status}&maxResults=${maxResults}
```

Input Parameters

<i>processmodelId</i>	String. The identifier of the process model. This attribute is required.
<i>stepId</i>	String. The identifier of the process model step. This attribute is required.
<i>startTime</i>	Integer or String. The query start time (Java long or ISO 8601). This attribute is required.
<i>endTime</i>	Integer or String. The query end time (Java long or ISO 8601). This attribute is required.
<i>status</i>	String. The process status to be returned. Valid values are "ALL", "STARTED", "RUNNING", "COMPLETED". This attribute is not required (defaults to "ALL").
<i>maxResults</i>	String. The maximum number of results. If unlimited results are returned, use 0 or a negative number. This attribute is not required (defaults to 5000).

Output Parameters

<i>analysisEnabled</i>	Returns "TRUE" or "FALSE" to indicate whether analysis is enabled for the process.
<i>deploymentVersion</i>	Returns the deployment version of the process.
<i>startTime</i>	Returns the step start time as ISO 8601.
<i>startTimeAsJavaEpoch</i>	Returns the step start time as Java epoch.
<i>endTime</i>	Returns the step end time as ISO 8601 or returns empty if no end time is available.
<i>endTimeAsJavaEpoch</i>	Returns the step end time as Java epoch or returns -1 if no end time is available.
<i>processInstanceId</i>	Returns the identifier for the process instance.

<i>instanceIteration</i>	Returns the iteration of the process instance.
<i>loopIteration</i>	Returns the loop iteration for the process instance.
<i>parentinstanceIteration</i>	Returns the iteration value for the parent process.
<i>processInstanceStatus</i>	Returns the status of the process instance. Returned values are "STARTED", "RUNNING", or "COMPLETED".
<i>processModelId</i>	Returns the process model identifier.
<i>processModelType</i>	Returns the process model type, either "BPM" or "BAM". This indicates whether the process execution was orchestrated by Process Engine or not.
<i>stepInstanceStatus</i>	Returns the status of the step instance. Returned values are "RUNNING" or "COMPLETED".
<i>stepIteration</i>	Returns the current iteration value of the step.
<i>stepId</i>	Returns the identifier of the step.
<i>stepName</i>	Returns the name of the step.
<i>subProcessDeploymentVersion</i>	Returns the deployment version of the sub-process.

Step Instances Result Example

```
<results>
  <stepInstances>
    <stepInstance xmlns:xsi="http://www.w3.org/2001/
      XMLSchema-instance" xsi:type="stepInstance">
      <analysisEnabled>true</analysisEnabled>
      <deploymentVersion>1</deploymentVersion>
      <startTime>2014-07-30T10:25-06:00</startTime>
      <startTimeAsJavaEpoch>1398959160000</startTimeAsJavaEpoch>
      <endTime>2014-08-01T09:46-06:00</endTime>
      <endTimeAsJavaEpoch>1406907960000</endTimeAsJavaEpoch>
      <processInstanceId>393x3-389x-ey393-56ez</processInstanceId>
      <instanceIteration>1</instanceIteration>
      <loopIteration>0</loopIteration>
      <parentInstanceIteration>0</parentInstanceIteration>
      <processInstanceStatus>COMPLETED</processInstanceStatus>
      <processModelId>Order_Fulfillment</processModelId>
      <stepInstanceStatus>RUNNING</stepInstanceStatus>
      <stepIteration>1</stepIteration>
      <stepId>S3</stepId>
      <stepName>Order_Receipt</stepName>
      <subprocessDeploymentVersion>2</subprocessDeploymentVersion>
    </stepInstance>
  </stepInstances>
</results>
```

```

    </stepInstances>
    <totalCount>1</totalCount>
</results>

```

Stage Metric Data

This service returns various aggregated stage metrics for each stage related to the given process. The data is derived from the stage instances that were active during the given time range. Cycle time statistical calculations can be derived from either the optional `statsInterval` parameter or from the size of the given range as shown below.

Time Range	Statistical Interval
<= 12 hours	1 hour
<= 2 days	4 hours
<= 2 weeks	1 day
<= 4 weeks	1 week
> 4 weeks	4 weeks

The URL format is as follows:

```

http://${optimize host}:${port}/services/rest/api/ProcessAnalytics/
StageMetrics?processModelId=${processModelId}&startTime=
${startTime}&endTime=${endTime}&statsInterval=${statsInterval}

```

Input Parameters

<i>processmodelId</i>	String. The identifier of the process model. This attribute is required.
<i>startTime</i>	Integer or String. The query start time (Java long or ISO 8601). This attribute is required.
<i>endTime</i>	Integer or String. The query end time (Java long or ISO 8601). This attribute is required.
<i>statsInterval</i>	String. The statistics interval size. Valid values are "1HOUR", "4HOUR", "1DAY", "1WEEK", "4WEEK". This attribute is not

required (if not provided, the default is calculated from the time range).

Output Parameters

<i>cancel OnBreach</i>	Returns an indication of whether the stage will cancel on breach.
<i>conditionDuration</i>	Returns the condition duration of the stage in milliseconds.
<i>cycleTimeAverage</i>	Returns the cycle time average, in milliseconds, for all stages that completed during the time range.
<i>endMilestone</i>	Returns the end milestone name for the stage.
<i>stageDescription</i>	Returns the description for the stage.
<i>stageId</i>	Returns the identifier for the stage.
<i>stageName</i>	Returns the name of the stage.
<i>startMilestone</i>	Returns the start milestone for the stage.
<i>volumeBreached</i>	Returns the count of breaches for the stage during the time range.
<i>volumeCompleted</i>	Returns the count of completed stages during the time range.
<i>volumeIncomplete</i>	Returns the count of incomplete stages during the time range.
<i>volumeRunningCount</i>	Returns the count of stages running at the end of the time range.
<i>volumeStarted</i>	Returns the count of stages started during the time range.

Stage Metrics Result Example

```
<stageMetrics>
  <stageMetric>
    <cancelOnBreach>false</cancelOnBreach>
    <condition>GREATER_THAN</condition>
```

```

    <conditionDuration>30000</conditionDuration>
    <cycleTimeAverage>32862</cycleTimeAverage>
    <endMilestone>Good Credit (S12).Complete</endMilestone>
    <stageDescription>stage 1</stageDescription>
    <stageId>18227992-9deb-46b3-b0f7-618ffa2948a5</stageId>
    <stageName>stage1</stageName>
    <startMilestone>Validate Credit (S6).Start</startMilestone>
    <volumeBreached>160</volumeBreached>
    <volumeCompleted>1160</volumeCompleted>
    <volumeIncomplete>0</volumeIncomplete>
    <volumeRunning>0</volumeRunning>
    <volumeStarted>1160</volumeStarted>
  </stageMetric>
</stageMetrics>

```

Stage Volumes

This service returns stage execution metrics by stage for the specified process and time range. The metrics consist of counts for sages started, running and complete during the specified time range.

The URL format is as follows:

```

http://${optimize host}:${port}/services/rest/api/ProcessAnalytics/
StageVolumes?processModelId=${processModelId}&startTime=
${startTime}&endTime=${endTime}

```

Input Parameters

<i>processmodelId</i>	String. The process model identifier of the process. This attribute is required.
<i>startTime</i>	Integer or String. The query start time (Java long or ISO 8601). This attribute is required.
<i>endTime</i>	Integer or String. The query end time (Java long or ISO 8601). This attribute is required.

Output Parameters

<i>stageId</i>	Returns the identifier for the stage.
<i>stageName</i>	Returns the name of the stage.

<i>volumeCompleted</i>	Returns the count of completed stages for the time range.
<i>volumeIncomplete</i>	Returns the count of incomplete stages for the time range.
<i>volumeRunning</i>	Returns the count of running stages during the time range.

Stage Volume Result Example

```
<stageVolumes>
  <stageVolume>
    <stageId>9abc537f-7c19-4d8e-80f2-f2f778e319ad</stageId>
    <stageName>AcceptOrderStart-EndOfTheProcess</stageName>
    <volumeCompleted>2</volumeCompleted>
    <volumeIncomplete>1</volumeIncomplete>
    <volumeRunning>1</volumeRunning>
  </StageVolume>
  <StageVolume>
    <stageId>8d127d72-63a2-41b0-abce-e987e8405e42</stageId>
    <stageName>FindCustomerInfoComplete-ValidateInventoryComplete</stageName>
    <volumeCompleted>1</volumeCompleted>
    <volumeIncomplete>0</volumeIncomplete>
    <volumeRunning>0</volumeRunning>
  </StageVolume>
</stageVolumes>
```

Stage Tracking Errors

This service returns a count of stage tracking errors, also known as breaches, that occurred for a specified process and time range. Counts are grouped by stage.

The URL format is as follows:

```
http://${optimize host}:${port}/services/rest/api/ProcessAnalytics/
StageErrors?processModelId=${processModelId}&startTime=
${startTime}&endTime=${endTime}
```

Input Parameters

<i>processmodelId</i>	String. The process model identifier. This attribute is required.
<i>startTime</i>	Integer or String. The query start time (Java long or ISO 8601). This attribute is required.
<i>endTime</i>	Integer or String. The query end time (Java long or ISO 8601). This attribute is required.

Output Parameters

<i>errorCount</i>	Returns the error count for the stage.
<i>stageId</i>	Returns the identifier for the stage.
<i>stageName</i>	Returns the name of the stage.

Stage Errors Result Example

```
<stageErrors>
  <stageError>
    <errorCount>160</errorCount>
    <stageId>18227992-9deb-46b3-b0f7-618ffa2948a5</stageId>
    <stageName>stage1</stageName>
  </stageError>
</stageErrors>
```

Stage Cycle Times

This service returns aggregated stage cycle time data for each stage related to the specified process and time range. The data is derived from the stage' instances that were completed during the given time range. Cycle time statistical calculations can be derived from either the optional statsInterval parameter or from the size of the given range as shown below.

Time Range	Statistical Interval
<= 12 hours	1 hour
<= 2 days	4 hours
<= 2 weeks	1 day
<= 4 weeks	1 week
> 4 weeks	4 weeks

The URL format is as follows:

```
http://${optimize host}:${port}/services/rest/api/ProcessAnalytics/
StageCycleTimes?processModelId=${processModelId}&startTime=
${startTime}&endTime=${endTime}&statsInterval=${statsInterval}
```

Input Parameters

<i>processmodelId</i>	String. The process model identifier of the process. This attribute is required.
<i>startTime</i>	Integer or String. The query start time (Java long or ISO 8601). This attribute is required.
<i>endTime</i>	Integer or String. The query end time (Java long or ISO 8601). This attribute is required.
<i>statsInterval</i>	String. The statistics interval size. Valid values are "1HOUR", "4HOUR", "1DAY", "1WEEK", "4WEEK". This attribute is not required (if not provided, the default is calculated from the time range).

Output Parameters

<i>average</i>	Returns the cycle time average in milliseconds for all stages that completed during the time range.
<i>max</i>	Returns the cycle time maximum in milliseconds for all stages that completed during the time range.
<i>min</i>	Returns the cycle time minimum in milliseconds for all stages that completed during the time range.
<i>historicalAverage</i>	Returns the cycle time historical average in milliseconds for all stages.
<i>historicalStdDev</i>	Returns the cycle time historical standard deviation in milliseconds for all stages.
<i>stageId</i>	Returns the identifier for the stage.
<i>stageName</i>	Returns the name of the stage.

Stage Cycle Times Result Example

```
<stageCycleTimes>
  <stageCycleTime>
    <average>32862.06896551724</average>
```

```

    <max>37010</max>
    <min>7901</min>
    <historicalAverage>2001.9</historicalAverage>
    <historicalStdDev>109.5</historicalStdDev>
    <stageId>18227992-9deb-46b3-b0f7-618ffa2948a5</stageId>
    <stageName>stage1</stageName>
  </stageCycleTime>
</stageCycleTimes>

```

Stage Instances

This service returns stage instance data for a specified process and time range. A set of stage model IDs may also be specified. Results may also be filtered by multiple stage statuses: started, completed, incomplete or breached. An "incomplete" status is a defined as a stage that was started but not completed before the process ended. If no status is given, then all stages that were active during the specified time range are returned.

The URL format is as follows:

```

http://${optimize host}:${port}/services/rest/api/ProcessAnalytics/
StageInstances?processModelId=${processModelId}&stageIds=
${stageIds}&startTime=${startTime}&endTime=${endTime}&status=
${status}&maxResults=${maxResults}

```

Example: `http://localhost:12503/services/rest/api/ProcessAnalytics/StageInstances?processModelId=ProcessModels/otc_bus_stages&startTime=1407996000000&endTime=1408049935760&status=STARTED,COMPLETED&b0f7-618ffa2948a5,67c79edc-7498-41c9-a695-b646652572cb&maxResults=10`

Input Parameters

<i>processmodelId</i>	String. The external process model identifier of the process. This attribute is required.
<i>stageIds</i>	The stage IDs, as comma separated. strings. This attribute is required.
<i>startTime</i>	Integer or String. The query start time (Java long or ISO 8601). This attribute is required.
<i>endTime</i>	Integer or String. The query end time (Java long or ISO 8601). This attribute is required.
<i>status</i>	String. Specifies the stage instance status. Valid values are "STARTED", "COMPLETED", "INCOMPLETE", "BREACHED". This attribute is no required (defaults to "ALL").

maxResults **Integer.** Specifies the maximum number of results. If unlimited results are returned, use 0 or negative number. This parameter is not required (defaults to 5000).

Output Parameters

<i>breach</i>	Returns "TRUE" or "FALSE" to indicate whether the stage was breached.
<i>cycleTime</i>	Returns the stage instance cycle time in milliseconds or returns -1 if no cycle times are available.
<i>processInstanceId</i>	Returns the process instance identifier.
<i>stageId</i>	Returns the identifier for the stage.
<i>stageInstanceId</i>	Returns the identifier for the stage instance.
<i>stageName</i>	Returns the name of the stage.
<i>startTime</i>	Returns the stage instance start time as ISO 8601.
<i>startTimeAsJavaEpoch</i>	Returns the stage instance start instance time as Java epoch.
<i>endTime</i>	Returns the stage instance end time as ISO 8601 or returns empty if no end time is available.
<i>endTimeAsJavaEpoch</i>	Returns the stage instance end time as Java epoch or returns -1 if no end time is available.
<i>stageState</i>	Returns the state of the stage instance. Valid values are "STARTED", "COMPLETE", "BREACHED", "INCOMPLETE".

Stage Instances Result Example

```
<results>
  <stageInstances>
    <stageInstance>
      <stageInstance>
        <cycleTime>37000</cycleTime>
        <processInstanceId>Shift=0&type=stages&instance=
          2014-08-01T12:01:29.968Z-stagetest1&id=rangeID_00010
          IterationID_0000100</processInstanceId>
        <stageId>18227992-9deb-46b3-b0f7-618ffa2948a5</stageId>
        <stageInstanceId>1875</stageInstanceId>
```

```

    <stageName>stage1</stageName>
    <startTime>2014-05-01T09:46-06:00</startTime>
    <startTimeAsJavaEpoch>1398959160000</startTimeAsJavaEpoch>
    <endTime>2014-08-01T09:46-06:00</endTime>
    <endTimeAsJavaEpoch>1406907960000</endTimeAsJavaEpoch>
    <stageState>COMPLETED</stageState>
  </stageInstance>
</stageInstances>
</results>

```

Stage Instances for a Process Instance

This service returns stage instance data for a specified process instance.

The URL format is as follows:

```

http://${optimize host}:${port}/services/rest/api/ProcessAnalytics/
StageInstancesForProcessInstance?processInstanceId=${processInstanceId}

```

Input Parameters

processInstanceId **String**. The process instance identifier (external ID or internal database ID). This attribute is required.

Output Parameters

<i>breach</i>	Returns "TRUE" or "FALSE" to indicate whether the stage was breached.
<i>cycleTime</i>	Returns the average stage instance cycle time for the process instance in milliseconds or returns -1 if no cycle time is available.
<i>endTime</i>	Returns the stage instance end time as ISO 8601 or returns empty if no end time is available.
<i>endTimeAsJavaEpoch</i>	Returns the stage instance end time as Java epoch or returns -1 if no end time is available.
<i>processInstanceId</i>	Returns the process instance identifier.
<i>stageId</i>	Returns the identifier for the stage.
<i>stageInstanceId</i>	Returns the identifier for the stage instance.

<i>stageName</i>	Returns the name of the stage.
<i>startTime</i>	Returns the stage instance start time as ISO 8601.
<i>startTimeAsJavaEpoch</i>	Returns the stage instance start time as Java epoch.
<i>stageState</i>	Returns the state of the stage instance. Valid values are "STARTED", "COMPLETE", "BREACHED", "INCOMPLETE".

Stage Instances Result Example

```
<results>
  <stageInstances>
    <stageInstance xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:type="stageInstance">
      <breach>false</breach>
      <cycleTime>37000</cycleTime>
      <endTime>2014-08-01T12:02-06:00</endTime>
      <endTimeAsJavaEpoch>1406916129618</endTimeAsJavaEpoch>
      <processInstanceId>2x5rfd</processInstanceId>
      <stageId>18227992-9deb-46b3-b0f7-618ffa2948a5</stageId>
      <stageInstanceId>355</stageInstanceId>
      <stageName>stage1</stageName>
      <startTime>2014-08-01T12:01-06:00</startTime>
      <startTimeAsJavaEpoch>1406916092618</startTimeAsJavaEpoch>
      <stageState>COMPLETED</stageState>
    </stageInstance>
  </stageInstances>
</results>
```

Active Process Instance Count for a Point in Time

Returns a count of process instances that were active at the given point in time for a specified process.

The URL format is as follows:

```
http://${optimize host}:${port}/services/rest/api/ProcessAnalytics/
ActiveProcessCountForTime?processModelId=${process model Id}&time=
${time}
```

Input Parameters

<i>processModelId</i>	String. The process model identifier. This attribute is required.
<i>time</i>	Integer or String. A specified point in time (Java long or ISO 8601). This attribute is required.

Output Parameters

<i>count</i>	Returns the count of active processes.
<i>processModelId</i>	Returns the process model identifier.
<i>processName</i>	Returns the name of the process.

Process Count Result Example

```
<ProcessCount>
  <count>23</count>
  <processModelId>Order Fulfillment</processModelId>
  <processName>Order Fulfillment</processName>
</ProcessCount>
```

Active Process Instance Count for Time Range

This service returns a count of process instances that were active for a specified process and time range (inclusive). Active process instances are defined as a process instances that were active at any point in the time range:

- Process instances that were started before the time range and ended during the time range.
- Process instances that were started before the time range and ended after the time range.
- Process instances that were started before the time range and are still running.
- Process instances that were started during the time range and ended during the time range.
- Process instances that were started during the time range and ended after the time range.
- Process instances that were started during the time range and are still running.

The URL format is as follows:

```
http://${optimize host}:${port}/services/rest/api/ProcessAnalytics/
ActiveProcessCountForTimeRange?processModelId=${process model
Id}&startTime=${startTime}&endTime=${endTime}
```

Input Parameters

<i>processModelId</i>	String. The process model identifier. This attribute is required.
<i>startTime</i>	Integer or String. The query start time (Java long or ISO 8601). This attribute is required.
<i>endTime</i>	Integer or String. The query end time (Java long or ISO 8601). This attribute is required.

Output Parameters

<i>count</i>	Returns the count of active process instances.
<i>processModelId</i>	Returns the process model identifier.
<i>processName</i>	Returns the name of the process.

Process Count Result Example

```
<ProcessCount>
  <count>23</count>
  <processModelId>Order Fulfillment</processModelId>
  <processName>Order Fulfillment</processName>
</ProcessCount>
```

III For Database Administrators

■ Installing Optimize Database Component Sets	317
■ Configuring Optimize Database Pools	321
■ Purging Data from Optimize Database Components	329

The chapters in this part of the guide contain information about the tasks primarily performed by Optimize database administrators.

14 Installing Optimize Database Component Sets

■ Overview	318
■ Database Component Sets	318
■ Order of Database Component Installation	318

Overview

You can install webMethods products including Optimize before you install Optimize database component sets. To install webMethods products, follow the instructions in *Installing webMethods and Intelligent Business Operations Products*.

When you are ready to install Optimize database component sets, read this chapter to understand database requirements and performance considerations for Optimize. After you have installed the required database component sets in the proper configuration, you can connect Optimize database activities to Optimize database component sets through JDBC pools.

Database Component Sets

Optimize requires four sets of database components:

- Analysis
- Process Tracker
- Process Audit Log
- My webMethods Server

All four Optimize database component sets can reside in the same schema on the same database. However, you can improve performance by distributing the database load over multiple database servers. To do so, you install the Optimize database component sets on different database servers and then configure the JDBC pools to point to the appropriate schemas on those database servers.

Note: If you are using an Oracle database for the Analysis component of an Optimize installation, set the Oracle `open_cursors` value to 1000 to minimize the chance of running out of cursors and generating an open cursors error. This setting may be especially critical for systems that are running under a heavy load.

Order of Database Component Installation

The following table lists the four Optimize database component sets, the database components in each set, and the order in which you must install the database components in each set.

For installation instructions, see *Installing webMethods and Intelligent Business Operations Products*. For a list of the required version of each database component, see the readme file for this release of Optimize.

Optimize Database Component Set	Database Components in Set, Required Version of Each Component, and Order of Installation
Analysis	<ol style="list-style-type: none"> 1. Operation Management 2. Distributed Locking 3. Analysis 4. If you want to install the data purge database component at the same time, include the following: Data Purge
Process Tracker	<ol style="list-style-type: none"> 1. Operation Management 2. Process Tracker 3. If you want to install the data purge database component at the same time, include the following: Data Purge
Process Audit Log	<ol style="list-style-type: none"> 1. Process Audit Log 2. In a webMethods Business Process Management environment, we recommend that you include the following database components in the Process Audit Log database schema: Process Engine IS Core Audit Log
My webMethods Server	<ol style="list-style-type: none"> 1. My webMethods Server

If the Analysis and Process Tracker database component sets are installed in the same database schema, you need to install the following database components only once:

- Operation Management
- Data Purge

To install the three Optimize database component sets in the same database schema, install the database components in the order shown in the following table:

Optimize Database Component Set	Database Components in Set, Required Version of Each Component, and Order of Installation
Analysis,	<ol style="list-style-type: none"> 1. Operation Management

Optimize Database Component Set	Database Components in Set, Required Version of Each Component, and Order of Installation
Process Tracker, and Process Audit Log (installed in one database schema)	<ol style="list-style-type: none"> 2. Distributed Locking 3. Analysis 4. Process Tracker 5. Process Audit Log 6. In a webMethods Business Process Management environment, we recommend that you include the following database components in the Process Audit Log database schema: <ul style="list-style-type: none"> Process Engine IS Core Audit Log 7. If you want to install the data purge database component at the same time, include the following: <ul style="list-style-type: none"> Data Purge
My webMethods Server	<ol style="list-style-type: none"> 1. My webMethods Server

15

Configuring Optimize Database Pools

■ Overview	322
■ Concepts	322
■ Installing Optimize Database Component Sets	324
■ Defining Database Connection Pools	325
■ Mapping Database Connection Pools to Optimize Activities	326

Overview

This chapter explains how to configure Java Database Connectivity (JDBC) connection pools to connect Optimize database activities to Optimize database component sets.

For more information about JDBC connection pools and database component sets, see:

- *Installing webMethods and Intelligent Business Operations Products*
- *Configuring BAM*

Concepts

A typical implementation of Optimize involves placing the four Optimize database component sets — Analysis, Process Tracker, Process Audit Log, and My webMethods Server — in four database schemas. For more information, see "[Installing Optimize Database Component Sets](#)" on page 317

Optimize has six components or *activities* that access the Optimize database component sets:

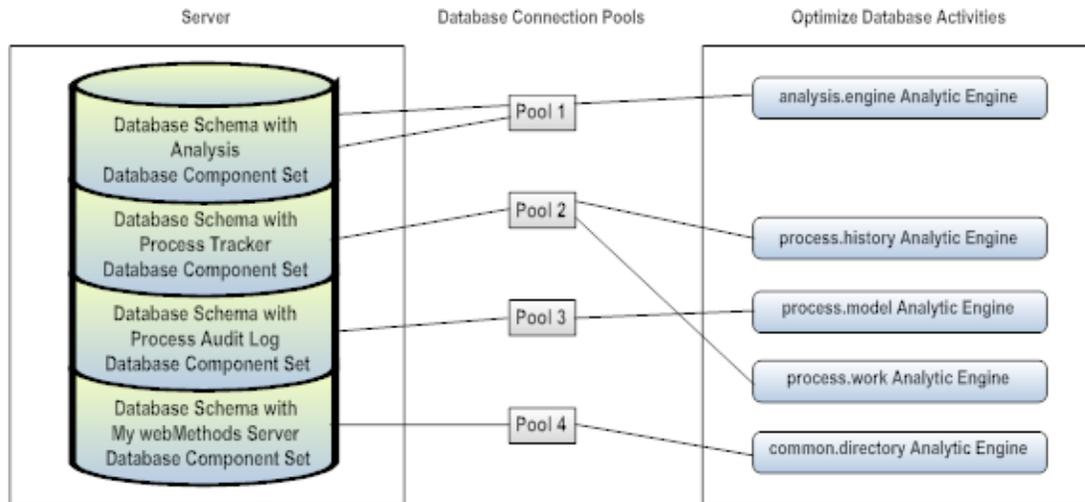
- analysis.engine Analytic Engine
- process.history Analytic Engine
- process.model Analytic Engine
- process.work Analytic Engine
- common.directory Analytic Engine

A typical Optimize configuration of database connection pools involve four connections:

- analysis.engine Analytic Engine through a pool to the Analysis database component set
- process.history Analytic Engine and process.work Analytic Engine through a single pool to the Process Tracker database component set
- process.model Analytic Engine through a pool to the Process Audit Log database component set
- common.directory Analytic Engine through a pool to the My webMethods Server database component set

See the following figure.

Connecting Optimize database activities to Optimize database component sets: typical configuration

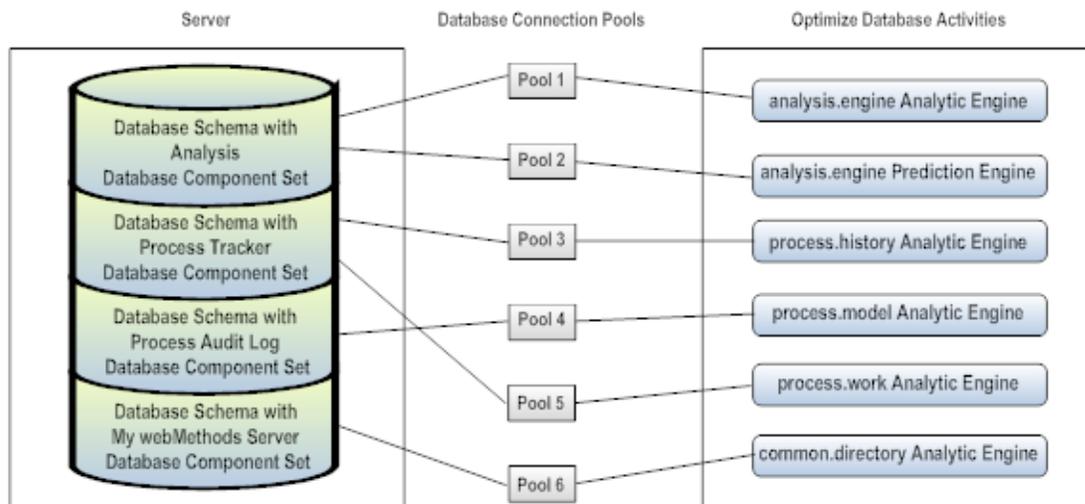


In high-volume situations, you may be able to improve performance by assigning separate database connection pools to:

- The process.history Analytic Engine and process.work Analytic Engine Optimize database activities; both connect to the Process Tracker database component set.
- The analysis.engine Analytic Engine Optimize database activities; connects to the Analysis database component set.

See the following figure.

Connecting Optimize database activities to Optimize database component sets: high-volume configuration



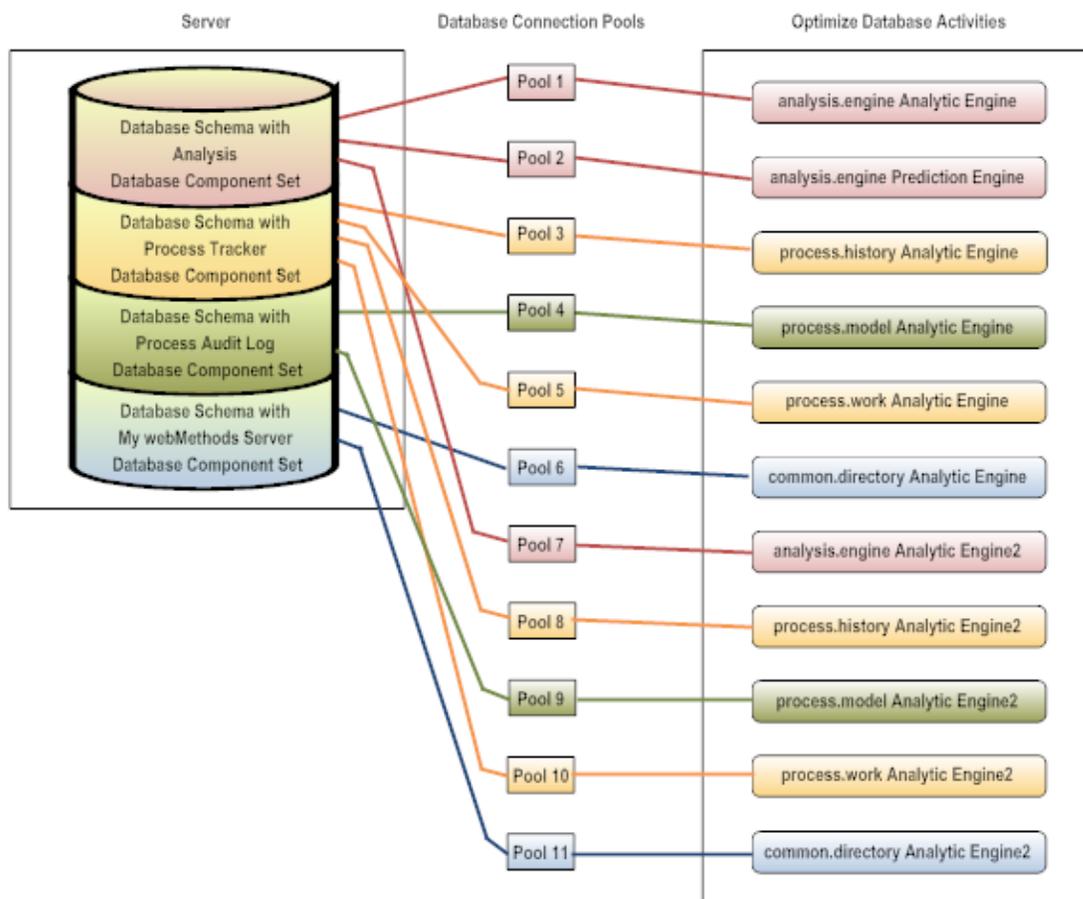
Note: You must define a minimum of one database connection pool per database schema. In an Optimize configuration with four database schemas as shown

in the figures above, you must define a minimum of four database connection pools.

Multiple Analytic Engines

In an environment with multiple Analytic Engines, each Analytic Engine has a separate set of four database activities (analysis.engine, process.history, process.model, and process.work) that connect to the same database schemas. You can reuse the first Analytic Engine's database connection pool definition for the second Analytic Engine, or you can give each Analytic Engine its own database connection pool definition as shown in the figure below.

Connecting Optimize database activities to Optimize database component sets: two Analytic Engines



Installing Optimize Database Component Sets

Before you configure Optimize database connection pools, the Optimize database component sets should be installed in one or more database schemas.

For instructions, see the following:

- *Installing webMethods and Intelligent Business Operations Products*
- ["Installing Optimize Database Component Sets" on page 317](#) in this guide

Defining Database Connection Pools

You can perform this step before or after you define the webMethods Optimize environment.

Note: You cannot complete the definition of an environment until you have defined database connection pools and mapped them to Optimize database activities.

To define database connection pools

1. In My webMethods: **Navigate > Applications > Administration > System-Wide > Environments > Database Pool Configuration.**

2. Click **Add Pool.**

The Database Pool Configuration page displays fields for defining pool information, a database connection, and pool settings.

3. Type a name for the new database connection pool in the **Name** text box.

If you are defining four database pools, you might name them *Analysis*, *ProcessTracker*, *ProcessAuditLog*, and *MywebMethodsServer*.

4. Type a description of the database connection pool in the **Description** text box.

You might describe the purpose of the database connection pools and the location of the database schema and Optimize database activity or activities that the pool is connecting.

5. Select the RDBMS from the drop-down list.

6. Type the URL to the database schema containing one or more Optimize database component sets in the **URL** text box.

The RDBMS-specific format for the URL appears in the text box.

Important: For DB2, you must add the following additional options to the URL:

```
AlternateID=<schema_name>;showSelectableTables=false
```

Here is the sample format with the two additional options:

```
jdbc:wm:db2://<server-name-or-IP-address>:<port>;
```

```
(DatabaseName=<database-name>|LocationName=<location-name>)
```

```
[;AlternateID=<schema_name>;showSelectableTables=false] The
```

```
<schema_name> parameter must be capitalized.
```

For more information, see *Configuring BAM*.

7. Type the database user and database password in the corresponding fields.
8. Inspect the default pool settings and make changes as needed.

We recommend these settings for Optimize:

<u>Database pool setting</u>	<u>Recommended value</u>
Minimum Connections	4
Maximum Connections	18
Idle Connection Timeout	20 seconds
Ramp-up Delay	500 milliseconds
Connection Tries	8
Retries Backoff	500 milliseconds
Allow Statement Caching	no (unchecked)

For more information, see *Configuring BAM*.

9. Click **Save**.
10. Repeat the above steps for each new database connection pool.

Mapping Database Connection Pools to Optimize Activities

After you have defined the Optimize database connection pools, you can map them to Optimize database activities.

To map database connection pools to Optimize database activities

1. In My webMethods: **Navigate > Applications > Administration > System-Wide > Environments > Define Environments**.

The Define Environments page displays.

Note: If you have not already done so, define the Optimize environment. For more information, see *Configuring BAM*.

2. Click the name of the defined environment for which you want to map database connection pools to Optimize database activities.
3. Click the **Map DB Pools** tab.

The list of Optimize database activities displays for the environment. The name of the list is **Database Components for [environment name]**.

4. Assign each Optimize database activity a database connection pool using the **Pool** drop-down list beside the name of the activity.

For example, an administrator defines four database connection pools named `Analysis`, `ProcessTracker`, `ProcessAuditLog`, and `CommonDirectory`. The administrator would assign them as follows:

- `Analysis` to `analysis.engine` Analytic Engine
 - `ProcessTracker` to `process.history` Analytic Engine and `process.work` Analytic Engine
 - `ProcessAuditLog` to `process.model` Analytic Engine
 - `CommonDirectory` to `common.directory` Analytic Engine
5. When you have assigned a database connection pool to each of the Optimize database activities, click **Finish**.

16 Purging Data from Optimize Database Components

■ Overview	330
■ Data Purge	330
■ Live Purge	340

Overview

This chapter explains how to purge data from the Optimize Analysis and Process Tracker database components using the Data Purge and the Live Purge features. It covers installing, running, configuring, and viewing purge operation events.

Data Purge

Best practices recommend that you purge data from the Analysis and Process Tracker database components on a regular basis to limit the growth of the database components and to improve performance. Optimize purges the Analysis and Process Tracker database components using stored procedures. The logic of the purge operation is driven by records in the OPERATION_PARAMETER table, which resides in the Analysis and Process Tracker database components. For more information, see ["Configuring Data Purge" on page 333](#).

Installing Data Purge

For instructions about installing the Data Purge database component, see *Installing webMethods and Intelligent Business Operations Products*.

Required Permissions for Oracle Database Users

The Oracle Data Purge feature depends on advanced Oracle capabilities. For example, the dbms_job package is used as the scheduling mechanism for operations, and the dbms_alert package is used as the alerting mechanism for operations.

In addition to the standard permissions, database users must have the following permissions:

- EXECUTE on SYS.DBMS_ALERT
- SELECT on SYS.V_\$PARAMETER

The permissions below are normally granted to PUBLIC, but make sure the permissions have been granted.

- EXECUTE on SYS.DBMS_JOB
- EXECUTE on SYS.UTL_SMTP
- EXECUTE on SYS.DBMS_METADATA
- EXECUTE on SYS.DBMS_OUTPUT
- EXECUTE on SYS.DBMS_LOB
- EXECUTE on SYS.PLITBLM
- EXECUTE on SYS.DBMS_UTILITY

These SYSDBA privileges can be granted either by using the DCI to call the storage database component or manually. If using the DCI, append the Login Role of sysdba to the server_URL, like this:

```
jdbc:wm:oracle://DBserver:1521;servicename=orcl;sysLoginRole=sysdba
```

Running Data Purge

Before you run the purge operation, shut down all Optimize processes. The purge can take several hours to execute. You can track the progress of a running purge operation in the OPERATION_LOG table. See "[Viewing Purge Operation Events](#)" on page 338.

Oracle

Important: To run massive_data_purge in Oracle, you must be logged in as the schema owner, *not* as SYSTEM or SYS.

Optimize purges data from database components in Oracle by calling a stored procedure from an Oracle database client. The procedure purges data by re-creating the database component and re-inserting the data you want to keep. With this approach, the purge operation always takes about the same length of time, regardless of the length of time between purges, so you can purge data infrequently (for example, about every three months). However, while frequent database purging will not speed up the time duration of the purge, you can improve the performance of the data purge operation by keeping fewer days of data.

The stored procedure call is:

```
execute massive_data_purge;
```

Avoid aborting a data purge procedure because doing so can affect foreign key constraints. However, if the massive_data_purge procedure for Oracle fails or is aborted, call this stored procedure to restore all Foreign Key constraints:

```
declare l_execution_count integer;
begin
metadatapkg.execute_pending_sql (l_execution_count,
                                'MASSIVE_DATA_PURGE',
                                'Add Foreign Keys',
                                metadatapkg.cyes
                                );
end;
```

Important: If the Analysis and Process Tracker database components are in separate schemas, the Data Purge must be executed in each schema separately.

SQL Server

Optimize purges data from databases in SQL Server by calling a stored procedure from an MSSQL database client. This procedure purges data by issuing DELETE statements.

The stored procedure call is:

```
exec data_mgmt_purge_data
```

Avoid aborting a data purge procedure because doing so can affect foreign key constraints. Data purge in SQL Server disables constraints one at a time. If the `data_mgmt_purge_data` procedure for SQL Server fails or is aborted, call this stored procedure to restore all Foreign Key constraints:

```
EXEC db_mgmt_modify_dependant_fk_constraints_all 'CHECK'
```

Important: If the Analysis and Process Tracker database components are in separate schemas, the Data Purge must be executed in each schema separately.

DB2

To run the DB2 Data Purge, execute the `data_mgmt_purge_data` stored procedure. Enter the following from the command line where DB2 has been installed:

1. Windows Only: `db2cmd` (skip this step if running on UNIX)
2. `db2 connect to [dbname] user [username] using [password]`
where:
 - `[dbname]` is the DB2 database containing the Analysis database component and/or the Process Tracker database component.
 - `[username]` `[password]` are the credentials for accessing the DB2 database listed above and access to the schemas containing the Analysis and Process Tracker database components.
 - `[analysis or process tracker schema name]` is the schema containing the Analysis and/or Process Tracker tables.
3. `db2 set path [Analysis Schema or Process Tracker Schema], SYSTEM`
where `[analysis or process tracker schema name]` is the schema containing the Analysis and/or Process Tracker tables.
4. `db2 set current schema [Analysis Schema or Process Tracker Schema]`
where `[analysis or process tracker schema name]` is the schema containing the Analysis and/or Process Tracker tables.
5. Initiate the purge:
 - Windows: `db2 call data_mgmt_purge_data()`
 - UNIX: `db2 "call data_mgmt_purge_data()"`

Important: If the Analysis and Process Tracker database components are in separate schemas, the Data Purge must be executed in each schema separately.

Resetting Referential Constraints if Data Purge Fails

If the Data Purge fails or is cancelled by the user, there is a chance the referential constraints were not re-enabled. Use the following command sequence to enable them.

1. (Windows Only:) `db2cmd` (Skip this step if running on UNIX.)

2. db2 connect to [dbname] user [username] using [password]
3. db2 set path [Analysis Schema or Process Tracker Schema], SYSTEM
4. db2 set current schema [Analysis Schema or Process Tracker Schema]
5. Initiate the purge:
 - Windows:


```
db2 call db_mgmt_modify_dependant_fk_constraints_all ('ENFORCED')
```
 - UNIX:


```
db2 "call db_mgmt_modify_dependant_fk_constraints_all ('ENFORCED')"
```

Configuring Data Purge

Data Purge uses values in the OPERATION_PARAMETER table when it runs. The Optimize Analytic Engine dynamically modifies records in the OPERATION_PARAMETER table based on:

- Attributes you set on the **Configure Servers** tab on the Define Environments page. For instructions, see ["Setting Data Purge Attributes for the Analytic Engine" on page 333](#).
- Parameters you enter directly into the OPERATION_PARAMETER table. For instructions, see ["Setting Purge Operation Parameters" on page 335](#).

You can identify types of operations to log. For instructions, see ["Specifying Information to Log" on page 338](#).

If you are using Oracle, you can identify people to notify by e-mail message when a purge operation has completed or encountered an error. For instructions, see ["Specifying People to Notify about the Oracle Purge Operation" on page 338](#).

Setting Data Purge Attributes for the Analytic Engine

Follow these instructions to set Data Purge attributes that control the number of days to retain business event data, the number of days to retain process data, and the frequency with which Optimize recalculates the values in the OPERATION_PARAMETER table.

1. In My webMethods: **Navigate > Applications > Administration > System-Wide > Environments > Define Environments**.
2. On the Define Environments page, click the name of the configured environment for which you want to set data purge attributes.
3. On the Edit Environment page, click the **Configure Servers** tab.
4. On the **CONFIGURATION** tree, find the Analytic Engine for which you want to set Data Purge attributes, and click **Data Maintenance Settings** under the name of the Analytic Engine.
5. Set the data maintenance attributes as follows.

<u>Attribute</u>	<u>Definition</u>
Business Days To Retain	Number of days to retain business event data in the Analysis database component. Business data includes metrics about processes, such as cycle time, instance count, or error count, and data captured from within processes, such as order revenue, items ordered, or line count. After the specified number of days, the data is eligible to be purged by the purge operation. The default is 90 days.
Data Maintenance Interval	The number of hours that Optimize waits before recalculating the values in the OPERATION_PARAMETER table in the Analysis database component (see " Setting Purge Operation Parameters " on page 335). The default is 4 hours.
Aggregated Business Days to Retain	Number of days to retain aggregated business event data in the Analysis database component. Aggregated business data represents a consolidation of business event data that is used to improve the performance of the KPI Summary and KPI Instance Details graphs. Aggregated business data takes up much less space and therefore can be kept for a longer period of time without consuming excessive disk space or affecting system performance. The default is 365 days.

Note: By default, Business Days to retain is set to fewer days than Aggregated Business Days to Retain to conserve storage space. However, data retained as aggregated data may not be as complete as data retained as business data. If you find you need to retain more complete data for a longer period of time, you can increase the number of Business Days to Retain.

6. Click **Save**.
7. On the **CONFIGURATION** tree, click **Process Tracker Settings** under **Analytic Engine**.
8. Set the Process Tracker attribute as follows.

<u>Attribute</u>	<u>Definition</u>
Days To Retain Processes	Number of days to retain process data in the Process Tracker database component. Process data is data about process execution, such as an Order Process started at 10:52:31, step 1 succeeded, step 2 failed, and so on. After the specified number of days, the data is eligible to be purged by the purge operation. The default is 60 days.

Attribute**Definition**

Note: Optimize recalculates the values in the OPERATION_PARAMETER table in the Process Tracker database component every 24 hours.

9. Click **Save**.

Setting Purge Operation Parameters

You set these parameters directly in the OPERATION_PARAMETER table:

- Mail host for e-mail notifications (Oracle only).
- Whether to drop old tables after the purge operation.
- Time to wait for the purge operation to complete before the job times out.
- Whether stale purge jobs should be removed.
- Number of days to retain data in the OPERATION_LOG table.

Note: If you installed the Analysis and Process Tracker database component sets in different database schemas, you set these parameters by editing two OPERATION_PARAMETER tables, one in each database schema. For more information, see ["Installing Optimize Database Component Sets" on page 317](#)

The OPERATION_PARAMETER table in the Analysis and Process Tracker database components contains a row for every database table and parameter associated with the purge operation. The table looks similar to this in each database component:

PARAMETER_ID	OPERATION_CD	PARAMETER_GROUP_CD	PARAMETER_CD	PARAMETER_DSCR	PARAMETER_VALUE
1	OPERATION_MGMT		MAILHOST	Mail host (server) through which to send notifications (e-mail messages) that indicate purge success or failure	host.com
2	MASSIVE_DATA_PURGE		DROP_OLD_TABLES	Flag to determine whether the old table should be	YES

PARAMETER_ID	OPERATION_CD	PARAMETER_GROUP_CD	PARAMETER_CD	PARAMETER_DSCR	PARAMETER_VALUE
				dropped after a successful purge operation	
3	MASSIVE DATA PURGE		JOB TIMEOUT	Time in seconds to wait for the purge operation to complete; if time exceeded, operation will stop and send notification that job has timed out	10800
4	MASSIVE DATA PURGE		REMOVE STALE JOBS	Flag to identify whether stale purge jobs should be removed	YES
5	MASSIVE DATA PURGE	CONTROL DATE	OPERATION_ LOG	Number of days to retain data in OPERATION_ LOG table	14
6	MASSIVE DATA PURGE	WHERE CLAUSE	BAM_FACT_ DTHNDLR_V1	Where clause for retaining data in BAM_FACT_ DTHNDLR_V1 table	ENTRY_ TIME> 11332971000
7	MASSIVE DATA PURGE	WHERE CLAUSE	BAM_FACT_ FCT4VNT_V1	Where clause for retaining data in BAM_FACT_ FCT4VNT_V1 table	ENTRY_ TIME> 11247387600

PARAMETER_ID	OPERATION_CD	PARAMETER_GROUP_CD	PARAMETER_CD	PARAMETER_DSCR	PARAMETER_VALUE
8	MASSIVE DATA PURGE	WHERE CLAUSE	BAM_PT_PROC ESS_INST	Where clause for retaining data in BAM_PT_ PROCESS_INST table	PROCESS_ INST_ID> 63110
9	MASSIVE DATA PURGE	WHERE CLAUSE	BAM_PT_STEP _INST	Where clause for retaining data in BAM_PT_STEP_ INST table	PROCESS_ INST_ID> 63110
10	MASSIVE DATA PURGE	WHERE CLAUSE	BAM_PT_STEP _STAGE	Where clause for retaining data in BAM_PT_STEP_ STAGE table	PROCESSED_ FL ! = 1

The PARAMETER_DSCR column describes each parameter, while the PARAMETER_VALUE column contains the value for each parameter. The table below indicates how each value is set by Optimize or by you.

Row	Value
1	You set this value. <i>It functions only with Oracle.</i> In the OPERATION_MGR table, you can identify people to receive e-mail alerts when the Oracle purge operation completes or encounters an error. You identify the mail host to use in this row. For instructions, see "Specifying People to Notify about the Oracle Purge Operation" on page 338.
2-4	These rows have values you can change if necessary.
5	You can set this value. It controls the number of days retained in the OPERATION_LOG table. For information on the OPERATION_LOG table, see "Viewing Purge Operation Events" on page 338.
6-7	Calculated by Optimize based on the Business Days To Retain attribute. See "Setting Data Purge Attributes for the Analytic Engine" on page 333.

Row	Value
8-10	Calculated by Optimize based on the Days To Retain Processes attribute. See "Setting Data Purge Attributes for the Analytic Engine" on page 333 .

Specifying Information to Log

All operations log errors in the OPERATION_LOG table (see ["Viewing Purge Operation Events" on page 338](#)). If you want to log additional information about an operation, such as the time an operation step began or the current status of the operation, set the DEBUG_FL parameter in the OPERATION_MGR table to Y. The additional logged information is also stored in the OPERATION_LOG table.

The OPERATION_MGR table contains rows for major database operations, such as purge. For example:

OPERATION_CD	DEBUG_FL	EMAIL_RECIPIENTS
DATA PURGE	Y	<i>name@company.com</i>
MASSIVE DATA PURGE	Y	<i>name@company.com</i> (Oracle)
MONITOR_ARCHIVE	Y	<i>name@company.com</i>

Specifying People to Notify about the Oracle Purge Operation

You can identify one or more people to receive e-mail alerts when the Oracle purge operation completes or encounters an error:

- Supply the recipients' e-mail addresses in the EMAIL_RECIPIENTS column of the OPERATION_MGR table, separated by commas. See the example above.
- You must also specify a mail host in the PARAMETER_VALUE column of the OPERATION_PARAMETER table. See ["Setting Purge Operation Parameters" on page 335](#).

Viewing Purge Operation Events

The OPERATION_LOG table contains logging data on purge execution. You can use the information in this table to track the performance of and troubleshoot the purge operation.

Optimize purges this table based on the value you set in the OPERATION_LOG parameter of the OPERATION_PARAMETER table. See ["Setting Purge Operation Parameters" on page 335](#).

Here is an example of the OPERATION_LOG table:

LOG_ID	OPERATION_CD	PROCEDURE_NAME	RTN_STATUS_CD	EVENT_TEXT	CTRL_INSERT_DT	CTRL_UPDATE_DT
2427	MASSIVE DATA PURGE	MASSIVE_DA TA_PURGE	0	Waiting for jobs to complete	5/8/2005 9:28:56 PM	5/8/2005 9:28:56 PM
2428	MASSIVE DATA PURGE	MASSIVE_DA TA_PURGE	0	START MASSIVE_DATA _PURGE	5/8/2005 9:28:58 PM	5/8/2005 9:28:58 PM
2429	MASSIVE DATA PURGE	MASSIVE_DA TA_PURGE	0	Removed 0 stale jobs	5/8/2005 9:30:02 PM	5/8/2005 9:30:02 PM
2430	MASSIVE DATA PURGE	MASSIVE_DA TA_PURGE	0	Creating CD job for table OPERATION _LOG	5/8/2005 9:31:00 PM	5/8/2005 9:31:00 PM
2431	MASSIVE DATA PURGE	MASSIVE_DA TA_PURGE	0	Dropping foreign keys related to tables being purged	5/8/2005 9:33:00 PM	5/8/2005 9:33:00 PM
2432	MASSIVE DATA PURGE	MASSIVE_DA TA_PURGE	0	Sending jobs to queue for processing	5/8/2005 9:35:23 PM	5/8/2005 9:35:23 PM
2433	MASSIVE DATA PURGE	MASSIVE_DA TA_PURGE	0	Job FullInsertJo bMgmt.FullIn sertJob('BAM _FACT_STEP_M ETRICS_V1...	5/8/2005 9:36:43 PM	5/8/2005 9:36:43 PM

LOG_ID	OPERATION_CD	PROCEDURE_NAME	RTN_STATUS_CD	EVENT_TEXT	CTRL_INSERT_DT	CTRL_UPDATE_DT
2434	MASSIVE DATA PURGE	MASSIVE_DA TA_PURGE	0	Registering alert: BAM_FACT_MTR CS_V1/319832 8093	5/8/2005 9:40:06 PM	5/8/2005 9:40:06 PM
2435	MASSIVE DATA PURGE	MASSIVE_DA TA_PURGE	0	Waiting for jobs to complete	5/8/2005 9:41:26 PM	5/8/2005 9:41:26 PM
2436	MASSIVE DATA PURGE	MASSIVE_DA TA_PURGE	0	Executing DDL: CREATE INDEX "BMFCTSTPMTR CSV1DD_IX_ X"...	5/8/2005 9:53:16 PM	5/8/2005 9:53:16 PM
2437	MASSIVE DATA PURGE	MASSIVE_DA TA_PURGE	0	BAM_FACT_STE P_METRICS_V1 rebuilt	5/8/2005 9:57:12 PM	5/8/2005 9:57:12 PM
2438	MASSIVE DATA PURGE	MASSIVE_DA TA_PURGE	0	END - All jobs have completed * Start Time: 2005-08-05 09:28:58 PM...	5/8/2005 9:59:35 PM	5/8/2005 9:59:35 PM

Live Purge

This section explains how to purge data from the Optimize database components using Live Purge. It covers configuring Live Purge, running it, and viewing the results of the Live Purge operations.

Live Purge allows users to execute a purge operation while the application is running. The database remains in a consistent state during the purge activities.

The Live Purge component is a part of your Optimize installation. For instructions on how to install, see *Installing webMethods and Intelligent Business Operations Products*.

You must have administrator privileges on My webMethods Server to be able to configure the parameters for the Live Purge operation, run it, and then view the results using the My webMethods user interface.

You can also use the Live Purge Standalone Command Line tool to execute a purge operation. For more information about the Live Purge Standalone tool, see "[Using the Live Purge Standalone Tool](#)" on page 347.

Configuring Live Purge

Before running a Live Purge operation, Live Purge parameters must be configured via the My webMethods user interface.

To configure the Live Purge parameters

1. In My webMethods Server: **Navigate > Administration > Analytics > Data Management > Live Purge**

My webMethods displays the **Live Purge** page.

2. On the Live Purge page, in the **General Settings** panel, select one of the two available radio buttons.

Depending on your choice, additional fields appear allowing you to define the number of days to retain or the retention period start date for the purge operation.

- Select the **Number of Days to Retain (Ending with Today)** radio button to specify the number of days for which to retain data in the selected Optimize database components. Outside the specified number of days, the data is eligible to be purged by the Live Purge operation.

If you select this radio button, an additional field appears, allowing you to define the number of days for which data will be retained during the Live Purge operation. The value must be an integer between 0 and 730 inclusive. The default value is 90 days. For example, if you specify 0 (zero), the Live Purge operation will retain all of today's data up to the current time of day. If you specify 1 (one), then all of today's data plus the data from the previous calendar day will be retained. The value you provide for this setting is validated and an appropriate error message displays.

- Select the **Retention Period Start Date (Ending with Today)** radio button to specify the period start date for which to retain data in the selected Optimize database components. All data prior to the specified date will be eligible to be purged by the Live Purge operation.

If you select this radio button, two additional fields are enabled, allowing you to define the start date and time of the retention period. You can use the **Calendar** icon or directly type the desired date. The value you provide for this setting is validated and an appropriate error message displays if you select a future date.

Note: If you use DB2, the fields for start time for the **Retention Period Start Date (Ending with Today)** radio button are not enabled. In this case the data from the whole day will be purged.

3. On the Live Purge page, in the **General Settings** panel, click the **Advanced Settings** button. A pop-up dialog displays the advanced settings that you can configure for the Live Purge operation.

The settings you define here are validated and an appropriate error message displays when the values are not valid.

Setting	Description
Transaction Size (records)	<p>Specifies the maximum number of database records to be purged in a single transaction. The value you provide here must be an integer between 100 and 20000 inclusive. The default value is 3000 records.</p> <p>Note: It is recommended that you execute the purge operation by smaller sizes. If the value entered in this field is too big, this may impact significantly the performance of the Analytic Engine during the Live Purge operation.</p>
Sleep Period (milliseconds)	<p>Specifies the wait time between the transactions. The default value is 0. The value you provide here must be an integer between 0 and 30000 inclusive. The default value is 0 milliseconds.</p> <p>Note: Use this parameter only if the Analytic Engine is under heavy load while the Live Purge operation is running. The larger the value of this field is, the longer it will take for the Live Purge operation to complete, but the performance of the Analytic Engine during this time will be better.</p>
Time Box (days)	<p>Specifies the length of a timeboxed period in days. Live Purge will delete the data by the timeboxed periods in chronological order, starting from the oldest and moving toward the most recent records. For example, if you define this setting to 5, the data will be deleted by five-day time frames. The value you provide here must be an integer between 0 and 180 inclusive. The default value is 7 days.</p> <p>This parameter allows you to make sure that if the Live Purge operation is stopped by an administrator or terminated abnormally, all data not included in the defined timeboxed period will remain in a consistent state and will be visible to the Analytic Engine.</p>

Setting	Description
Process Instances to Delete	<p>Specifies the status of the process instances to be purged. Two values are possible:</p> <ul style="list-style-type: none"> ■ COMPLETED - the Live Purge operation will be executed only over completed process instances. ■ ALL (the default) - the Live Purge operation will be executed over all process instances, regardless of their status.

- a. Click **Save** to save your settings and apply them when you start the Live Purge operation.

You must provide values within the allowed range to be able to save them.

- b. Click **Cancel** to revert to the latest saved settings in the Advanced Settings dialog.

4. In the **Selection Criteria** panel, check the database components to be purged.

By default the two database components are selected. If none of the components are selected, the Live Purge operation will not be carried out.

Database Component	Description
Process Data	Purge process instances
Analysis Data	Purge KPI readings, problems, etc.

Note: The statistics data will not be affected by the purge operation, due to its specific nature and the fact that it is updated incrementally, that is, every time period depends on the previous ones.

5. In the **Specific Items to Purge** panel, clear the names of specific process models uploaded for analysis only whose data you wish to retain. By default, all process model names are selected.

The table below contains all process models currently uploaded for analysis only on your system. By default, 10 items are displayed. You can use the scroll bar to see the complete list of process models uploaded for analysis only.

When the table contains process names, you can purge data as follows:

- Leave all process names selected (by default) to purge all data, including intrinsic metrics and system data as specified with the other criteria.
- Select specific process names to purge only the data related to them.
- Clear all process names so no data is purged.

When there are no process names in the table, the purge operation will delete only intrinsic and system data.

Note: When you purge all data related to a specific process model uploaded for analysis only, you can later delete the process model(s) in question by going to **Navigate > Administration > Business > Business Processes**. Currently, process models uploaded for execution cannot be deleted once they have been used.

Running Live Purge

After configuring the parameters for Live Purge via the My webMethods user interface, you can start the purge operation.

To start the Live Purge operation

1. In My webMethods Server: **Navigate > Administration > Analytics > Data Management > Live Purge**
2. Verify the configuration of the parameters for the purge operation.
3. Click **Start** to start the Live Purge operation.
4. Click **Yes** in the confirmation dialog that appears.

The database is purged according to the defined settings and remains in a consistent state during the purge activities.

Stopping Live Purge

You can stop a running purge operation using the My webMethods user interface.

To stop the Live Purge operation

1. In My webMethods Server: **Navigate > Administration > Analytics > Data Management > Live Purge**
2. Click **Stop** to stop the Live Purge operation.

- a. To immediately stop the purge operation, click **Yes** in the confirmation dialog that appears.

As a result, some inconsistent database records may remain in the database. However, these records will not be visible to the Analytic Engine and will be purged next time you run a Live Purge operation.

- b. To stop the purge operation after the latest timeboxed period is purged, click **Stop Gracefully**.

All database records within the currently processed period defined by the **Time Box** parameter are deleted before the Live Purge operation is stopped.

The purge operation is stopped and the Results panel displays the amount of data that has been purged.

Viewing Live Purge Results

While the Live Purge operation is running, My webMethods displays information about the state and the result of the procedure in a panel on the **Navigate > Administration > Analytics > Data Management > Live Purge** page. The current state and the time of the last state update for the Live Purge operation are displayed on top of the panel. The possible values for the state are:

The state...	Shows that...
Unknown	Currently there is no connection to the Analytic Engine and the Live Purge operation cannot obtain a valid state.
N/A	No Live Purge operation is running at the moment. No statistics are displayed in the Purge Results panel.
Estimating	<p>The Live Purge operation is currently estimating the database records that will be deleted according to the defined settings.</p> <p>Note: The Process Instances Scheduled for Purge and the Total Amount of DB Records to Purge fields in the Purge Results panel show you the result of the estimation.</p>
Deleting	<p>The Live Purge operation is currently running and database records are being deleted according to the defined settings.</p> <p>Note: The Remaining Time and Progress fields in the Purge Results panel show you the actual progress of the operation.</p>
Stopping	<p>A Live Purge stop has been initiated using the Stop button. A confirmation dialog appears.</p> <ul style="list-style-type: none"> ■ If you use the Stop Gracefully option in the confirmation dialog, all currently processed database records within the period defined by the Time Box parameter are deleted before the operation is stopped. ■ If you use the Yes option in the confirmation dialog, all currently processed database records within the period defined by the Transaction Size parameter are deleted before the operation is stopped.
Completed	The Live Purge operation is completed. All database records that were eligible for purge according to the current settings have been successfully deleted.

The state...	Shows that...
	<p>Note: The Purged DB Records field in the Purge Results panel shows the amount of database records that have been deleted during the operation.</p>
Stopped Gracefully	<p>The Live Purge operation has been stopped using the Stop Gracefully option.</p> <p>Note: All database records within the currently processed period defined by the Time Box parameter are deleted before the operation is stopped.</p>
Stopped	<p>The Live Purge operation has been stopped using the Yes option.</p> <p>Note: All database records within the currently processed period defined by the Transaction Size parameter are deleted before the operation is stopped.</p>
Failed	<p>The Live Purge operation could not be completed successfully.</p> <p>Note: The Purged DB Records field in the Purge Results panel displays the amount of database records that have been deleted before the operation failed.</p>

The Purge Results panel contains the following information:

The field...	Displays statistics about...
Process Instances Scheduled for Purge	The estimated amount of process instances which will be purged.
Total Amount of DB Records to Purge	The estimated amount of database records that will be purged.
Purged DB Records	The actual number of database records which have been purged during the Live Purge operation.
Purge Time	The time during which the Live Purge operation has been running.
Remaining Time	The time that remains until the Live Purge operation is completed.

The field...	Displays statistics about...
Speed of Purge	The approximate speed at which the Live Purge operation runs. It is measured in deleted database records per second.
Purge Progress	The progress of the Live Purge operation in percent.

Using the Live Purge Standalone Tool

You can use the Live Purge Standalone Command Line tool to execute a purge operation on the Optimize database components while the application is running. The database remains in a consistent state during the purge activities.

The Live Purge standalone tool is available in your Optimize installation under *Optimize_directory*\analysis\bin.

The installation folder contains:

- **livepurge.bat or livepurge.sh** - an executable file that enables you to start the Live Purge tool in a Windows or a UNIX environment.
- **livepurge.jar** - an executable jar file.
- **LivePurge.properties** - a file containing the properties that must be set before executing a purge operation, as well as instructions for setting those properties.

Important: During the installation, the *analysis_configuration_base_config_path=* property in the LivePurge.properties file is set to point to the *Optimize_directory* installation folder, by default *C:/SoftwareAG/optimize/analysis/conf*. Do not modify this value.

Prerequisites

Before running the Live Purge Standalone Command Line tool, you must configure its properties using the LivePurge.properties file. Instructions for doing so are available in the LivePurgeStandalone.properties file.

You must set at least the following property:

retention_date=

Starting the Live Purge Standalone Tool

You can start the Live Purge Standalone utility using the command prompt from the *Optimize_directory*\analysis\bin directory in the following way:

To start the Live Purge Standalone Command Line tool

1. Execute the following command:
 - On Windows systems:

```
livepurge.bat
```

- On UNIX systems, after setting the executable attribute:

```
./livepurge.sh
```

Stopping the Live Purge Standalone Tool

You can stop the Live Purge Standalone utility using the command prompt in the following way:

To stop the Live Purge Standalone tool

- Type `CTRL+C` at the command prompt.

A Monitored KPI Definitions

■ Overview	350
■ Universal Messaging KPI Definitions	350
■ Broker KPI Definitions	361
■ Territory KPI Definitions	371
■ Gateway KPI Definitions	373
■ Integration Server KPI Definitions	375
■ webMethods Trading Networks KPI Definitions	400
■ My webMethods Server KPI Definitions	408
■ Infrastructure Data Collector KPI Definitions	426
■ Adapter KPI Definitions	426
■ Terracotta KPI Definitions	440
■ Presto KPI Definitions	454

Overview

This appendix describes the KPIs that Infrastructure Data Collector collects for its monitored components.

webMethods KPI definitions represent the monitoring capabilities of the various monitored resources associated with webMethods components. The following section represents the KPI definitions for each webMethods monitored resource. A monitored component may represent a physical entity, a network service, or an abstraction of a resource that exists independently of its use in management.

Note: Object status is monitored automatically when the component has been selected for monitoring.

Universal Messaging KPI Definitions

The following KPIs are available for monitoring various aspects of Universal Messaging performance.

KPI Definition

["com.softwareag.um.realm.node" on page 350](#)

["com.softwareag.um.realm.channel" on page 353](#)

["com.softwareag.um.realm.queue" on page 355](#)

["com.softwareag.um.realm.datagroup" on page 356](#)

["com.softwareag.um.realm.datagroup" on page 356](#)

["com.softwareag.um.realm.datagroup" on page 356](#)

com.softwareag.um.realm.node

The KPIs in this enable you to monitor aspects of a Universal Messaging realm. A Universal Messaging realm is the name given to a single Universal Messaging server. Universal Messaging realms can support multiple network interfaces, optionally supporting different Universal Messaging protocols. Each such interface is represented by a URL, known as an RNAME. Thus a single realm server can have more than one RNAME.

Universal Messaging realm KPIs can be found on the Analytics Overview page, listed under the realm host and port.

Discovery Mechanism

Universal Messaging servers are discovered via Asset Discovery in Optimize.

KPIs

Name	Description
int CurrentConnections	Total number of connections on this realm.
int BuffersCreated	Number of data buffers created within the realms and for use within event fanout.
int BuffersReused	Number of data buffers reused by the event fanout.
int FanoutBacklog	Returns the number of events waiting to be processed by the fanout engine.
int FreeMemory	Returns the amount of free memory the Realm Server has within the JVM. This indicates the difference between what the JVM has currently allocated and what the Realm Server has used. The JVM may increase its memory usage if the free memory falls below a set threshold depending on the JVM parameters passed during the realm server startup script.
int NoOfChannels	Total number of channels within the realm.
int NoOfQueues	Total number of queues within the realm.
int NoOfDataGroups	Total number of data groups within the realm.
int NoOfThreads	Number of threads the JVM has allocated for the Realm Server. This includes all the threads used by the JVM as well.

Name	Description
int QueuedThreads	Total number of threads currently queued ready to execute tasks.
int ReusedThreads	Total number of reused threads within the realm.
int InternalSchedulerSize	Total number of scheduled tasks currently running within the realm.
int TotalConnections	Total number of connections served by this realm from the time it started.
int TotalGCCount	Total time in milliseconds that the JVM has spent doing GCs.
int TotalMemory	Total number of bytes that the JVM has allocated from the underlying OS.
int TotalPublished	Total number of events published on this realm from the time it has started.
int TotalSuscribed	Total number of events that this realm has sent to clients from the time it started.
int VendedThreads	Total number of threads that have been allocated threads within the realm.
int ObjectStatus	Indicates Operational or Failed status of the component.
int DeltaBuffersCreated	The number of data buffers created within the realms and for use within event fanout.
int DeltaBuffersReused	The number of data buffers reused by the event fanout.
int DeltaReUsedThreads	Returns the number of reused threads within the realm.
long DeltaTotalConnections	The number of connections served by this realm from the time it started.

Name	Description
long DeltaTotalGCCount	Total time in milliseconds that the JVM has spent doing GCs.
long DeltaTotalMemory	The total number of bytes that the JVM has allocated from the underlying OS.
long DeltaTotalPublished	The total number of events published on this realm from the time it has started.
long DeltaTotalSubscribed	The total number of events that this realm has sent to clients from the time it started.
long DeltaVendedThreads	The total number of threads that have been allocated threads within the realm.

com.softwareag.um.realm.channel

Universal Messaging realm channels are the logical rendezvous point for data that is published and subscribed. If you are using a Universal Messaging Provider for JMS, then channels are the equivalent of JMS topics runtime environment for the Brokers.

Universal Messaging realm channel KPIs can be found on the Analytics Overview page, listed under the realm channel host and port.

Discovery Mechanism

Universal Messaging servers are discovered via Asset Discovery in Optimize.

KPIs

Name	Description
int ObjectStatus	Total number of connections on this realm.
float CacheHitRatio	Returns the channel/queues cache ratio. This ratio represents the (number of cache hits)/(Total Access). This cache is only active when the channel/queue is a persistent store and the server caches the event instead of re-reading it from the store again.

Name	Description
float ConnectionRate	The number of subscriptions made to this store per second.
float ConsumedRate	The number of events published per second.
long CurrentNumber OfConnections	Current number of connections to this channel.
long CurrentNumber ofEvents	Total number of events within the channel/queue.
long FanoutTime	Total number of milliseconds it takes to fanout each event to all consumers.
int OutboundJoinCount	Total count of outbound joins for this channel.
int PercentageFreeInStore	Amount of free space within a store as a percentage of the overall space used.
float PublishRate	Total number of events published per second.
long TotalConsumed	Total number of events that this queue has delivered to subscribers.
long TotalNoOfConnections	Total number of connections that this queue has had since the realm started.
long TotalPublished	Total number of events published/pushed to this channel/queue since the realm has started.
long UsedSpace	Number of bytes that this channel/queue consumes on the server.
long DeltaTotalConsumed	The total number of events that this channel has delivered to subscribers.
long DeltaTotalNoOfConnections	The total number of connections that this channel has had since the realm started.

Name	Description
long DeltaTotalPublished	The total number of events published/pushed to this channel/queue since the realm has started.

com.softwareag.um.realm.queue

A Universal Messaging realm queue is much like a channel; the primary difference is that only one consumer can read any individual event from a queue. Consumed events are immediately removed from the queue, and are no longer available for consumption by any other consumer. Thus, a queue guarantees that each event is delivered only once.

Universal Messaging realm queue KPIs can be found on the Analytics Overview page, listed under the realm queue host and port.

Discovery Mechanism

Universal Messaging servers are discovered via Asset Discovery in Optimize.

KPIs

Name	Description
int ObjectStatus	Total number of connections on this realm.
float CacheHitRatio	Retrieves the channel/queues cache ratio. This ratio represents the (number of cache hits)/(Total Access). This cache is only active when the channel/queue is a persistent store and the server caches the event instead of re-reading it from the store again.
float ConnectionRate	The number of subscriptions made to this store per second.
float ConsumedRate	The number of events published per second.
long CurrentNumber ofConnections	Returns the current number of connections to this channel.
long CurrentNumber ofEvents	Returns the number of events within the channel/queue.

Name	Description
longFanoutTime	The total number of milliseconds it takes to fanout each event to all consumers.
int OutboundJoinCount	Returns a count of outbound joins for this channel.
int PercentageFreeInStore	Returns the amount of free space within a store as a percentage of the overall space used.
float PublishRate	Total number of events published per second.
long TotalConsumed	Returns the total number of events that this queue has delivered to subscribers.
long TotalNoOfConnections	Returns the total number of connections that this queue has had since the realm started.
long TotalPublished	Returns the total number of events published/pushed to this channel/queue since the realm has started.
long UsedSpace	Returns the number of bytes that this channel/queue consumes on the server.
long DeltaTotalConsumed	Returns the total number of events that this channel has delivered to subscribers.
long DeltaTotalNoOfConnections	Returns the total number of connections that this channel has had since the realm started.
long DeltaTotalPublished	Returns the total number of events published/pushed to this channel/queue since the realm has started.

com.softwareag.um.realm.datagroup

Universal Messaging data groups provide a very lightweight grouping structure that allows developers to manage user subscriptions remotely and transparently. Data groups provide an alternative to channels/topics for publish/subscribe.

Universal Messaging realm data group KPIs can be found on the Analytics Overview page, listed under the realm data group host and port.

Discovery Mechanism

Universal Messaging servers are discovered via Asset Discovery in Optimize.

KPIs

Name	Description
int ObjectStatus	Status of the realm data group.
float ConsumedRate	The number of events published per second.
longFanoutTime	The total number of milliseconds it takes to fanout each event to all consumers.
float PublishRate	Total number of events published per second.
long StreamCount	Current number of streams that have been added to this data group.
long TotalConsumed	Total number of events published to this data group since the realm has started.
long TotalStreamCount	Total number of streams that have been added to this data group.
long TotalPublished	Total number of events published to this data group since the realm has started.
long DeltaTotalConsumed	Returns the total number of events published to this data group since the realm has started.
long DeltaTotalStreamCount	Returns the total number of streams that have been added to this data group.
long DeltaTotalPublished	Returns the total number of events published to this data group since the realm has started.

com.softwareag.um.realm.interface

Universal Messaging interfaces represent various communication protocols to Universal Messaging from external clients.

Universal Messaging realm interface KPIs can be found on the Analytics Overview page, listed under the realm interface host and port.

Discovery Mechanism

Universal Messaging servers are discovered via Asset Discovery in Optimize.

KPIs

Name	Description
string Adapter	The adapter that this interface uses.
string AdapterAlias	The aliases known to this interface.
boolean AllowNIO	Indicates whether NIO is enabled on this interface.
int AuthTimeout	The number of milliseconds that the remote client has to authenticate with the server before the server closes the communications with the client.
int Backlog	The number of socket connect requests that the operating system will queue before sending a reject to the remote client.
boolean ClientConnectionsAllowed	Indicates whether this interface can be used by clients to connect.
int CurrentIdleCount	Indicates the number of idle threads waiting for connections.
string ErrorMessage	Only valid if the Status is error=3.
boolean InterRealmAllow	Returns whether this interface is allowed to be used in inter realm / cluster communication.
string Name	The name of this interface.
boolean ObjectStatus	Status of this interface.

Name	Description
long PoolExhausted	Indicates the total number of times the thread pool reached 0 idle.
int Port	The Port this interface will bind to at startup.
string Protocol	The Protocol (nsp, nhp, nsps, nhps) used by this adapter.
int ReadBandWidth	Returns the read bandwidth in bytes.
long RecvBufferSize	The socket buffer size in bytes used by this interface when receiving data.
long RxBytes	Returns the number of bytes this interface has received.
int SelectThreadSize	The number of select threads used by NIO.
long SendBufferSize	The socket buffer size in bytes used by this interface when sending data.
int Status	Returns the current status of this interface. Can be one of stopped=0, started=1, paused=2 or error=3
int ThreadCount	The current Thread Pool size that handles client socket connections.
long TotalAuthenticationTime	Returns the total time spent authenticating connections.
long TotalConnections	Returns the total number of connections created by this interface.
long TotalFailedConnections	Returns the total number of connections which failed during the initial security handshake and validation.
long TotalReads	Returns the total number of reads.

Name	Description
long TotalWrites	Returns the total number of writes.
long TxBytes	Returns the number of bytes this interface has transmitted.
string URL	The URL that specifies this interface.
long WaitTime	Returns the total milliseconds that there was no accept thread processing new connections.
int WriteBandWidth	Returns the write bandwidth in bytes.
boolean useForProxyForward	Returns true if proxy forwarding is enabled.
long DeltaPoolExhausted	Indicates the total number of times the thread pool reached 0 idle.
long DeltaRxBytes	Returns the number of bytes this interface has received.
long DeltaTotalAuthenticationTime	Returns the total time spent authenticating connections.
long DeltaTotalConnections	Returns the total number of connections created by this interface.
long DeltaTotalFailedConnections	Returns the total number of connections which failed during the initial security handshake and validation.
long DeltaTotalReads	Returns the total number of reads.
long DeltaTotalWrites	Returns the total number of writes.
long DeltaTxBytes	Returns the number of bytes this interface has transmitted.

Name	Description
long DeltaWaitTime	Returns the total milliseconds that there was no accept thread processing new connections.

com.softwareag.um.realm.threadpool

Universal Messaging threadpools provide a way to manage threads within Universal Messaging.

Universal Messaging realm threadpool KPIs can be found on the Analytics Overview page, listed under the realm threadpool host and port.

Discovery Mechanism

Universal Messaging servers are discovered via Asset Discovery in Optimize.

KPIs

Name	Description
int Idle	The number of idle threads.
int Queue	The number of current tasks queued within the thread pool.
int Size	The total number of allocated threads in the pool.
long Total	The total number of tasks executed during the thread pool's lifetime.
long DeltaTotal	The total number of tasks executed during its lifetime.

Broker KPI Definitions

webMethods Broker KPI definitions are as follows:

KPI Definition

["wm.brokerdc.BrokerServer" on page 362](#)

["wm.brokerdc.Broker" on page 366](#)

["wm.brokerdc.DocumentType" on page 369](#)

wm.brokerdc.BrokerServer

The Broker Server is a process that hosts one or more webMethods Brokers. It provides storage and a runtime environment for the Brokers.

KPIs can be found on the Analytics Overview page, listed under the Broker Server host and port.

Discovery Mechanism

Broker Servers are discovered via Asset Discovery and Network Discovery in Optimize.

KPIs

Name and Description	Description
string LicenseKey	Broker Server software license key.
string DataDirectory	Directory where the Broker Server stores data.
string VersionString	Complete version string of the Broker Server software.
string SSLStatus.status	Current status of Secure Socket Layer (SSL) usage. Possible values are <code>ENABLED</code> , <code>DISABLED</code> , <code>ERROR</code> , <code>NOTSUPPORTED</code> .
string SSLStatus.encryption_level	Level of SSL encryption available. Possible values are <code>NONE</code> , <code>US Domestic</code> , <code>US Export</code> .
string SSLStatus.error_string	Description of the last error encountered starting SSL.

Name and Description	Description
date ServerStats.now	Current time on the Broker Server.
date ServerStats.createTime	Time when the Broker Server was first started.
date ServerStats.serverStartTime	Time when the Broker Server was most recently started.
int ServerStats.numConnections	Current number of non-SSL connections.
int ServerStats.numSSLConnections	Current number of SSL connections.
int ServerStats.highestNumConnections	Highest number of non-SSL connections since the last time the Broker Server was started.
int ServerStats.highestNumSSLConnections	Highest number of SSL connections since the last time the Broker Server was started.
date ServerStats.highestNumConnectionsTime	Time when HighestNumConnections last changed.
date ServerStats.highestNumSSLConnectionsTime	Time when highest NumSSLConnections last changed.
int ServerStats.licensedConnections	Licensed number of total SSL and non-SSL connections.-1 if not limited.
int ServerStats.licensedSSLConnections	Licensed number of SSL connections. -1 if not limited.
boolean ServerStats.isLicenseExpiringSoon	True if expiring within a week.
boolean ServerStats.isLicenseExpired	True if expired already.
date ServerStats.licenseExpirationTime	When the license expires.
int ServerStats.licenseRemaining	Number of seconds until license expires.

Name and Description	Description
boolean ServerStats.isDiskSpaceLow	True if disk free < 1MB.
boolean ServerStats.isDiskSpaceVeryLow	True if disk free < 100KB.
double SystemStats.userTimeUsed	Number of seconds (or fractions thereof) of CPU user time used by the Broker since it started.
double SystemStats.systemTimeUsed	Number of seconds (or fractions thereof) of CPU system time used by the Broker since it started.
long SystemStats.guaranteedSpaceUsed	Number of KB the guaranteed storage is using on the disk.
long SystemStats.guaranteedSpaceReserved	Number of KB the reserved for guaranteed storage on the disk. If zero, storage will be resized as needed. If non-zero, storage will be fixed at this size.
long SystemStats.guaranteedSpaceMax	Maximum number of KB the guaranteed storage can grow to due to system limits, if disk space or reservation limits are not hit first.
long SystemStats.guaranteedDiskSize	Size of the guaranteed store disk in KB.
long SystemStats.guaranteedSpaceAllocated	Disk space allocated for guaranteed storage.
long SystemStats.guaranteedDiskFree	Amount of free space on the disk in KB.
string SystemStats.guaranteedDiskName	Name of the guaranteed disk partition.
long SystemStats.persistentSpaceUsed	Number of KB the persistent storage is using on the disk.
date SystemStats.now	Current time on the Broker Server.
long SystemStats.persistentSpaceMax	Maximum number of KB the persistent storage can grow to due to system

Name and Description	Description
	limits, if disk space or reservation limits are not hit first.
long SystemStats.persistentDiskSize	Size of the persistent store disk in KB.
long SystemStats.persistentDiskFree	Amount of free space on the disk in KB.
string SystemStats.persistentDiskName	Name of the persistent disk partition.
long SystemStats.persistentSpaceAllocated	Disk space allocated for persistent storage.
long SystemStats.swapSpaceFree	Number of KB of free swap space.
long SystemStats.swapSpaceMax	Total number of KB of swap space.
long StorageStats.configCurrentKbytesInuse	Current storage size in use in the Config store (Kb, long).
long StorageStats.configCurrentKbytesReserved	Current storage size reserved in the Config store (Kb, long).
long StorageStats.configMaxKbytesAvailable	Maximum storage size available in the Config store (Kb, long).
long StorageStats.dataCurrentKbytesInuse	Current storage size in use in the Data store (Kb, long).
long StorageStats.dataCurrentKbytesReserved	Current storage size reserved in the Data store (Kb, long).
long StorageStats.dataMaxKbytesAvailable	Maximum storage size available in the Data store (Kb, long).
long StorageStats.sessions	The length of the structure array describing statistics of each of the storage files used.
int ObjectStatus	Indicates Operational or Failed status of the component.

Name and Description	Description
boolean ContainsResourceMessage	Flag indicating a resource message, such as a log message or exception message, has been received from the monitored resource.
int MessageSeverity	Severity level of the resource message: <ul style="list-style-type: none"> ■ Critical = 1 ■ Major = 2 ■ Minor = 3 ■ Warning = 4 ■ Info = 5 ■ Ok = 6 ■ Indeterminate = 7

wm.brokerdc.Broker

The webMethods Broker coordinates the exchange of documents between adapters and other client programs.

webMethods Broker components are contained in Broker Server components. A webMethods Broker can contain Custom Adapter and Document Type components.

Discovery Mechanism

webMethods Brokers are discovered automatically from a Broker Server.

KPIs

Name	Description
string BrokerName	webMethods Broker name.
date BrokerStats.now	Current time on the webMethods Broker.
date BrokerStats.createTime	Time when the webMethods Broker was created.

Name	Description
int BrokerStats.cumulativeNumEventsDelivered	Total number of events (documents) delivered by all clients.
int BrokerStats.cumulativeNumEventsPublished	Total number of events (documents) published by all clients.
int BrokerStats.cumulativeNumEventsQueued	Total number of events (documents) placed in client queues.
int BrokerStats.numEventsDelivered	Delta value of number of events (documents) delivered by all clients.
int BrokerStats.numEventsPublished	Delta value of number of events (documents) published by all clients.
int BrokerStats.numEventsQueued	Delta value of number of events (documents) placed in the queue.
int BrokerStats.numClients	Number of clients on the webMethods Broker, whether connected or not.
int BrokerStats.numEventTypes	Number of event types (document types) installed.
int BrokerStats.numTracesPublished	Number of trace events (documents) published by the webMethods Broker.
int BrokerStats.retryQueueMaxPublishes	Maximum simultaneous publish operations allowed by the retry queue.
int BrokerStats.retryQueueMaxEvents	Maximum events allowed in the retry queue.
int BrokerStats.retryQueueCurrentPublishes	Number of publish operations in the retry queue.
int BrokerStats.retryQueueCurrentEvents	Number of events in the retry queue.

Name	Description
int BrokerStats.retryQueueNextOpSequenceNum	Number of publish operations that required retries.
int BrokerStats.retryQueueNumberOfAttempts	Number of attempts to publish events in the retry queue.
int BrokerStats.retryQueueReservedTotalPublishes	Number of active publish operations.
int BrokerStats.retryQueueReservedGuaranteedPublishes	Number of active publish operations involving guaranteed events.
int BrokerStats.retryQueueReservedVolatilePublishes	Number of active publish operations involving only volatile events.
int BrokerStats.retryQueueReservedGuaranteedEvents	Number of guaranteed events in the active publish operations.
int BrokerStats.retryQueueReservedVolatileEvents	Number of volatile events in the active publish operations.
int BrokerStats.traceNumEventsQueued	Number of trace and activity events published.
date BrokerStats.traceLastEventEnqueueTime	Last time a trace or activity event was published.
long BrokerStats.traceQueueLength	Length of the trace event queue.
long BrokerStats.traceQueueByteSize	Size of the trace event queue.
long BrokerStats.traceQueueHighestLength	Peak length of the trace event queue.
date BrokerStats.traceQueueHighestLengthTime	Time the trace event queue reached its peak length.
BrokerStats.brokerThroughput	Throughput delta since the last poll.
BrokerStats.brokerCumulativeThroughput	Total throughput since the server started.

Name	Description
QueueStats.stalledQueues	Total number of stalled queues. A queue is stalled if the queue length is greater than 0 and no doc is retrieved for half an hour. A queue is also stalled if the queue length is greater than 0 and the last retrieved is newer.
QueueStats.noOfQueues	Count of all queues in the Broker.
int ObjectStatus	Indicates Operational or Failed status of the component.
boolean ContainsResourceMessage	Flag indicating a resource message, such as a log message or exception message, has been received from the monitored resource.
int MessageSeverity	Severity level of the resource message: <ul style="list-style-type: none"> ■ Critical = 1 ■ Major = 2 ■ Minor = 3 ■ Warning = 4 ■ Info = 5 ■ Ok = 6 ■ Indeterminate = 7

wm.brokerdc.DocumentType

A document type is a definition of messages that travel over a network from a publisher to a subscriber, through the webMethods Broker. Document type components are created for all document types not used by the system.

Document type components are contained in webMethods Broker components.

Discovery Mechanism

Document types are discovered automatically by webMethods Broker.

KPIs

Name	Description
date CreateTime	Time when the document type was created.
date LastDocumentDeliveryTime	Time when last document of this type was delivered.
date LastDocumentPublishTime	Time when last document of this type was published.
date LastForwardReceiveTime	Time when last document of this type was received via forwarding from another webMethods Broker.
int NumClientSubscriptions	Number of client subscriptions open which include this document type.
int NumEventsDelivered	Number of documents of this type delivered.
int NumEventsPublished	Number of documents of this type published.
int NumForwardsReceived	Number of documents of this type received via forwarding from another webMethods Broker.
date UpdateTime	Time when the document type was last changed.
int ObjectStatus	Indicates Operational or Failed status of the component.
boolean ContainsResourceMessage	Flag indicating a resource message, such as a log message or exception message, has been received from the monitored resource.
int MessageSeverity	Severity level of the resource message:

Name	Description
	<ul style="list-style-type: none"> ■ Critical = 1 ■ Major = 2 ■ Minor = 3 ■ Warning = 4 ■ Info = 5 ■ Ok = 6 ■ Indeterminate = 7

Territory KPI Definitions

A territory is a collection of webMethods Brokers that can forward documents to each other. webMethods Brokers in the same territory also share client group and document type information.

Territory components are logical constructs, so they will appear as Component Categories on the Analytics Overview page.

KPIs

Name	Description
string brokerName	webMethods Broker name.
string brokerHost	webMethods Broker host name.
string connectedFrom	IP address from which webMethods Broker is currently connected.
date lastConnectTime	Time when the webMethods Broker last connected.
int numEventsForwarded	Number of documents forwarded to the webMethods Broker.
int numEventsReceived	Number of documents received from the webMethods Broker.

Name	Description
int numEventsEnqueued	Number of documents enqueued for the webMethods Broker.
date lastEventForwardTime	Time when a document was last forwarded to the webMethods Broker.
date lastEventReceiveTime	Time when a document was last received from the webMethods Broker.
date lastEventEnqueueTime	Time when a document was last enqueued for the webMethods Broker.
long queueLength	Current size (in number of documents) of the webMethods Broker's queue.
long queueByteSize	Current size (in bytes) of the webMethods Broker's queue.
long queueHighestLength	Highest length (in number of documents) of the webMethods Broker's queue.
date queueHighestLengthTime	Time when queueHighestLength was last changed.
int ObjectStatus	Indicates Operational or Failed status of the component.
boolean ContainsResourceMessage	Flag indicating a resource message, such as a log message or exception message, has been received from the monitored resource.
int MessageSeverity	Severity level of the resource message: <ul style="list-style-type: none"> ■ Critical = 1 ■ Major = 2 ■ Minor = 3 ■ Warning = 4 ■ Info = 5 ■ Ok = 6

Name	Description
	■ Indeterminate = 7

Gateway KPI Definitions

A gateway is a connection between two webMethods Brokers in different territories, allowing the transfer of events between the territories. Each webMethods Broker belongs to its own territory but can share event types with its companion webMethods Broker across the gateway. A webMethods Broker can be a gateway to multiple territories.

Gateway components are logical constructs, so they will appear as Component Categories on the Analytics Overview page.

Discovery Mechanism

Gateway components are discovered automatically via webMethods Broker.

KPIs

Name	Description
string remoteTerritoryName	Name of the remote territory.
string gatewayBrokerName	Remote webMethods Broker name.
string gatewayHost	Remote webMethods Broker's host name.
date now	Current time on the webMethods Broker host.
string connectedFrom	IP address where webMethods Broker is currently connected from. The attribute will be empty if there is no connection.
date lastConnectTime	Time when the webMethods Broker last connected.
boolean gateway_pause	A flag indicating whether outbound traffic from this gateway webMethods Broker to territory_name is enabled or disabled

Name	Description
	(paused). Will be true if outbound traffic to territory_name has been paused.
string gateway_pause_client	The client id of the webMethods Broker Admin Client that paused the gateway. If outbound traffic is not currently paused for territory_name, this parameter will be empty.
int gateway_pause_session	The session id from which the gateway was paused. If outbound traffic is not currently paused for territory_name, this parameter will be 0.
date gateway_pause_time	The time when the gateway was paused. If outbound traffic is not currently paused for territory_name, this parameter will be 0.
int numEventsForwarded	Number of documents forwarded across the gateway.
int numEventsReceived	Number of documents received across the gateway.
int numEventsEnqueued	Number of documents enqueued for the gateway.
date lastEventForwardTime	Time when a document was last forwarded across the gateway.
date lastEventReceiveTime	Time when a document was last received across the gateway.
date lastEventEnqueueTime	Time when a document was last enqueued for the gateway.
string protocol_version	The version of the territory protocol.
long queueLength	Current size (in number of documents) of the gateway's queue.

Name	Description
long queueByteSize	Current size (in bytes) of the gateway's queue.
long queueHighestLength	Highest length (in number of documents) of the gateway's queue.
date queueHighestLengthTime	Time when queueHighestLength was last changed.
int ObjectStatus	Indicates Operational or Failed status of the component.
boolean ContainsResourceMessage	Flag indicating a resource message, such as a log message or exception message, has been received from the monitored resource.
int MessageSeverity	Severity level of the resource message: Critical = 1 Major = 2 Minor = 3 Warning = 4 Info = 5 Ok = 6 Indeterminate = 7

Integration Server KPI Definitions

Integration Server KPI definitions are as follows:

KPI Definition
"com.wm.isextdc.IntegrationServer" on page 376
"com.wm.isextdc.IsPackage" on page 379
"com.wm.isextdc.IsService" on page 380

KPI Definition

["com.wm.isextdc.IsFtpPort" on page 383](#)

["com.wm.isextdc.IsSocketPort" on page 384](#)

["com.wm.isextdc.IsHttpPort" on page 385](#)

["com.wm.isextdc.IsHttpsPort" on page 387](#)

["com.wm.isextdc.IsEmailPort" on page 388](#)

["com.wm.isextdc.RepoV2" on page 389](#)

["com.wm.isextdc.IsBrokerTransport" on page 390](#)

["com.wm.isextdc.IsScheduler" on page 392](#)

["com.wm.isextdc.IsSchedulerTask" on page 392](#)

["com.wm.isextdc.IsCluster" on page 394](#)

["com.wm.isextdc.IsClusterServer" on page 395](#)

["com.wm.isextdc.IsClusterServer" on page 395](#)

["com.wm.isextdc.IsClusterServer" on page 395](#)

["com.wm.isextdc.IsJDBCFunctionalAlias" on page 398](#)

com.wm.isextdc.IntegrationServer

Integration Server provides an environment for orderly, efficient, and secure execution of services.

Discovery Mechanism

Integration Servers are discovered via Asset Discovery and Network Discovery in Optimize.

KPIs

Name	Description
boolean SupportStrongSSLEncryption	True if strong (128-bit) encryption is available.
int MaxLicensedSessions	Maximum number of sessions licensed.
int MaxMemory	Maximum memory allocated to an IS process.
int MaxServiceThread	Maximum allocated service threads. Takes effect after restart.
int MinServiceThread	Minimum allocated service threads. Takes effect after restart.
int ServerLogLevel	Integration Server logging level. Possible values are from 0 to 9 with higher values producing more log messages.
int SessionTimeout	Session timeout value in minutes.
long PollingRate	Monitored object resource polling rate in seconds.
long TotalMemory	Total system memory available.
long UsedMemory	Total system memory in use.
long FreeMemory	Total system memory free.
int CurrentNumServiceThreads	Current number of service threads within last minute.
int PeakNumServiceThreads	Peak number of service threads used.
int CurrentNumVMThreads	Current number of Java VM threads within last minute.
int PeakNumVMThreads	Peak number of Java VM threads used.

Name	Description
int CurrentNumSessions	Current number of client sessions within last minute.
int PeakNumSessions	Peak number of client sessions.
int CurrentNumLicensedSession	Current number of licensed sessions within last minute.
int PeakNumLicensedSession	Peak number of licensed sessions.
long RequestsPerMinuteCurrent	Current number of service requests per minute within last minute.
long RequestsPerMinuteLifetime	Lifetime service requests per minute.
long CurrentAverageServiceRequestTime	Average request processing time within last minute.
long LifetimeAverageServiceRequestTime	Average request processing time for the lifetime of the server.
long NumServiceErrors	Number of service errors during the server uptime.
long deltaNumServiceErrors	Delta of the number of service errors during the server uptime.
int NumCompletedRequests	Number of service requests during server uptime.
int deltaNumCompletedRequests	Delta of the number of service requests during server uptime.
string HTTPProxyHostName	Host name of the outbound proxy server.
int HTTPProxyPort	Port used with the outbound proxy server.
int HTTPSProxyPort	Port used with the outbound HTTPS proxy server

Name	Description
int FTPProxyPort	Port used with the outbound FTP proxy server.
int ObjectStatus	Indicates Operational or Failed status of the component.
boolean ContainsResourceMessage	Flag indicating a resource message, such as a log message or exception message, has been received from the monitored resource.
int MessageSeverity	Severity level of the resource message: Critical = 1 Major = 2 Minor = 3 Warning = 4 Info = 5 Ok = 6 Indeterminate = 7

com.wm.isextdc.IsPackage

A package contains a set of services and related files, such as specifications, records, and output templates.

Discovery Mechanism

Automatically discovered via IS discovery

KPIs

Name	Description
int NumServicesLoaded	Total number of loaded services.
int NumLoadErrors	Number of load errors.
int NumLoadWarnings	Number of load warnings.

Name	Description
int ObjectStatus	Indicates Operational or Failed status of the component.
boolean ContainsResourceMessage	Flag indicating a resource message, such as a log message or exception message, has been received from the monitored resource.
int MessageSeverity	Severity level of the resource message: Critical = 1 Major = 2 Minor = 3 Warning = 4 Info = 5 Ok = 6 Indeterminate = 7

com.wm.isextdc.IsService

A service is a server-resident unit of functionality that clients can invoke. A service might be an entire application or used as part of a larger application. There are several types of services: flow, Java, C/C++, and WIDL.

Discovery Mechanism

Automatically discovered via IS discovery

KPIs

Name	Description
boolean Stateless	True for a stateless service.
boolean EnforceInternalACL	True if ACLs are enforced for nested service invocations.
boolean Caching	True if caching is enabled.

Name	Description
boolean CachingPrefetch	True if caching prefetch is enabled.
int CachingExpiration	Result cache expiration interval, in seconds.
int PrefetchActivate	Cache prefetch for the service result.
long CurrentlyRunning	Current number of requests being processed.
long CumulativeCount	Total number of service requests.
long CumulativeRunTime	Cumulative run time for this service, in milliseconds. The Cumulative Run Time by IS Service is an aggregated total of the run times over the Infrastructure Data Collector (IDC) polling interval, measured in milliseconds. For example, if the service runs three times in the polling interval for 10s each time, the KPI will show 30 000, calculated as 3 x 10 000 ms, added to the cumulative runtime value since start of Integration Server. Cumulative Runtime \ Cumulative Count will give the average runtime over the lifetime of the data.
long Count	Total number of service requests.
long Run Time	Delta value of run time for this service, in milliseconds. The Run Time by IS Service is an aggregated total of the run times over the Infrastructure Data Collector (IDC) polling interval, measured in milliseconds. For example, if the service runs three times in the polling interval for 10s each time, the KPI will show 30 000 calculated as 3 x 10 000 ms. The average runtime during the polling interval can be calculated by "Run Time by IS Service" \ "Count".

Name	Description
long RetryCount	Number of the times the service failed and was retried.
long ErrorCount	Number of times the service resulted in failure.
long deltaErrorCount	Delta of the number of times the service resulted in failure.
long EMAILProtocolNumInvokes	The number of times this service has been invoked by EMAIL protocol.
long deltaEMAILProtocolNumInvokes	Delta of the number of times this service has been invoked by EMAIL protocol.
long FTPProtocolNumInvokes	The number of times this service has been invoked by FTP protocol.
long deltaFTPProtocolNumInvokes	Delta of the number of times this service has been invoked by FTP protocol.
long HTTPProtocolNumInvokes	The number of times this service has been invoked by HTTP protocol.
long deltaHTTPProtocolNumInvokes	Delta of the number of times this service has been invoked by HTTP protocol.
long HTTPSProtocolNumInvokes	The number of times this service has been invoked by HTTPS protocol.
long deltaHTTPSProtocolNumInvokes	Delta of the number of times this service has been invoked by HTTPS protocol.
long SOAPProtocolNumInvokes	The number of times this service has been invoked by SOAP protocol.
long deltaSOAPProtocolNumInvokes	Delta of the number of times this service has been invoked by SOAP protocol.
long BROKEREXPORTProtocolNumInvokes	The number of times this service has been invoked by Broker Transport protocol.

Name	Description
long deltaBROKERXPORTProtocolNumInvokes	Delta of the number of times this service has been invoked by Broker Transport protocol.
int ObjectStatus	Indicates Operational or Failed status of the component.
boolean ContainsResourceMessage	Flag indicating a resource message, such as a log message or exception message, has been received from the monitored resource.
int MessageSeverity	Severity level of the resource message: Critical = 1 Major = 2 Minor = 3 Warning = 4 Info = 5 Ok = 6 Indeterminate = 7

com.wm.isextdc.IsFtpPort

An FTP port.

Discovery Mechanism

Automatically discovered via Integration Server discovery.

KPIs

Name	Description
int ListenerPort	Listener network port.
boolean DefaultIPAccess	True if access is open to all addresses <i>not</i> on the IPAccessList. False if access is open to only addresses on the IPAccessList.

Name	Description
boolean DefaultSvcAccess	True if all services are available except those in the SvcAccessList. False if only services in SvcAccessList are available.
boolean Primary	True if the port is the primary server port.
int ObjectStatus	Indicates Operational or Failed status of the component.
boolean ContainsResourceMessage	Flag indicating a resource message, such as a log message or exception message, has been received from the monitored resource.
int MessageSeverity	Severity level of the resource message: Critical = 1 Major = 2 Minor = 3 Warning = 4 Info = 5 Ok = 6 Indeterminate = 7

com.wm.isextdc.lsSocketPort

A native socket port, used when the Integration Server is configured as an Enterprise Gateway Server.

Discovery Mechanism

Automatically discovered via Integration Server discovery.

KPIs

Name	Description
int ListenerPort	Listener network port.

Name	Description
boolean DefaultIPAccess	True if access is open to all addresses <i>not</i> on the IPAccessList. False if access is open to only addresses on the IPAccessList.
boolean DefaultSvcAccess	True if all services are available except those in the SvcAccessList. False if only services in SvcAccessList are available.
boolean Primary	True if the port is the primary server port.
int ObjectStatus	Indicates Operational or Failed status of the component.
boolean ContainsResourceMessage	Flag indicating a resource message, such as a log message or exception message, has been received from the monitored resource.
int MessageSeverity	Severity level of the resource message: Critical = 1 Major = 2 Minor = 3 Warning = 4 Info = 5 Ok = 6 Indeterminate = 7

com.wm.isextdc.IsHttpPort

An HTTP port.

Discovery Mechanism

Automatically discovered via Integration Server discovery.

KPIs

Name	Description
string Type	Possible values are <code>load balancingport</code> and <code>Enterprise Gateway Server port</code> . A load balancing port routes a request to another server in the cluster for processing. An Enterprise Gateway Server port routes a request to an internal server behind the firewall for processing.
int ListenerPort	Listener network port.
boolean DefaultIPAccess	True if access is open to all addresses <i>not</i> on the <code>IPAccessList</code> . False if access is open to only addresses on the <code>IPAccessList</code> .
boolean DefaultSvcAccess	True if all services are available except those in the <code>SvcAccessList</code> . False if only services in <code>SvcAccessList</code> are available.
boolean Primary	True if the port is the primary server port.
int ObjectStatus	Indicates Operational or Failed status of the component.
boolean ContainsResourceMessage	Flag indicating a resource message, such as a log message or exception message, has been received from the monitored resource.
int MessageSeverity	Severity level of the resource message: Critical = 1 Major = 2 Minor = 3 Warning = 4 Info = 5 Ok = 6 Indeterminate = 7

com.wm.isextdc.IsHttpsPort

An HTTPS port.

Discovery Mechanism

Automatically discovered via Integration Server discovery.

KPIs

Name	Description
int PublicKeyLength	Public key length, either 40 bit or 128 bit.
boolean WeakEncryptionAllowed	True if weak encryption (less than 128 bits) is allowed.
int ListenerPort	Listener network port.
boolean DefaultIPAccess	True if access is open to all addresses <i>not</i> on the IPAccessList. False if access is open to only addresses on the IPAccessList.
boolean DefaultSvcAccess	True if all services are available except those in the SvcAccessList. False if only services in SvcAccessList are available.
boolean Primary	True if the port is the primary server port.
int ObjectStatus	Indicates Operational or Failed status of the component.
boolean ContainsResourceMessage	Flag indicating a resource message, such as a log message or exception message, has been received from the monitored resource.
int MessageSeverity	Severity level of the resource message: Critical = 1 Major = 2 Minor = 3

Name	Description
	Warning = 4
	Info = 5
	Ok = 6
	Indeterminate = 7

com.wm.isextdc.IsEmailPort

An e-mail port.

Discovery Mechanism

Automatically discovered via Integration Server discovery.

KPIs

Name	Description
int CheckInterval	Interval at which to check for mail.
boolean LogoutAfterEachCheck	True if the server will log off the IMAP server after checking for mail.
boolean RequiresAuth	True if clients will pass the user name and password via the user and pass parameters in the e-mail subject line.
int NumThreads	Number of threads for this listener (IMAP only).
boolean DeleteValidMsg	True if e-mail is deleted from the IMAP server after processing (IMAP only).
boolean DeleteInvalidMsg	True if invalid e-mail is deleted from the IMAP server (IMAP only).
int ListenerPort	Listener network port.

Name	Description
boolean DefaultIPAccess	True if access is open to all addresses <i>not</i> on the IPAccessList. False if access is open to only addresses on the IPAccessList.
boolean DefaultSvcAccess	True if all services are available except those in the SvcAccessList. False if only services in SvcAccessList are available.
boolean Primary	True if the port is the primary server port.
int ObjectStatus	Indicates Operational or Failed status of the component.
boolean ContainsResourceMessage	Flag indicating a resource message, such as a log message or exception message, has been received from the monitored resource.
int MessageSeverity	Severity level of the resource message: Critical = 1 Major = 2 Minor = 3 Warning = 4 Info = 5 Ok = 6 Indeterminate = 7

com.wm.isextdc.RepoV2

Repository Server Version 2 (applies only to discovery and monitoring of Integration Servers prior to version 7.1)

Discovery Mechanism

Auto-discover. Repository Server Version 2 is discovered by connecting to any Integration Server that uses it as Repository Server.

KPIs

Name	Description
boolean PrimaryServer	True if the contacted server is the primary server.
boolean RepoFileSystemBased	True if a file-based repository is used. False if a repository that supports JDBC is used.
int Port	The Repository Server Port
boolean Enabled	Indicates when the Repository Server is enabled. 1 = enabled, 0 = not enabled.
int ObjectStatus	Indicates Operational or Failed status of the component.
boolean ContainsResourceMessage	Flag indicating a resource message, such as a log message or exception message, has been received from the monitored resource.
int MessageSeverity	Severity level of the resource message: Critical = 1 Major = 2 Minor = 3 Warning = 4 Info = 5 Ok = 6 Indeterminate = 7

com.wm.isextdc.IsBrokerTransport

Reflects the Integration Server's use of a webMethods Broker. Returns the state of the Integration Server connection(s) to the webMethods Broker. Shows the webMethods Broker name and host. Exposes any Integration Server-side statistics on the use of the webMethods Broker.

Discovery Mechanism

Automatically discovered via Integration Server discovery.

KPIs

Name	Description
boolean Connected	Connected information for each webMethods Broker transport.
long NumDocsSent	Number of documents sent to the webMethods Broker from this Integration Server since the server has been running.
long deltaNumDocsSent	Delta of the number of documents sent to the webMethods Broker from this Integration Server since the server has been running.
long NumDocsRecv	Number of documents received from the webMethods Broker by this Integration Server since the server has been running.
long deltaNumDocsRecv	Delta of the number of documents received from the webMethods Broker by this Integration Server since the server has been running.
int ObjectStatus	Indicates Operational or Failed status of the component.
boolean ContainsResourceMessage	Flag indicating a resource message, such as a log message or exception message, has been received from the monitored resource.
int MessageSeverity	Severity level of the resource message: Critical = 1 Major = 2 Minor = 3

Name	Description
	Warning = 4
	Info = 5
	Ok = 6
	Indeterminate = 7

com.wm.isextdc.IsScheduler

The KPIs in this component provide information about the status of ISScheduler objects and the number of tasks currently running.

Note: IS Scheduler monitoring is supported only for Integration Server 8.2 and newer.

Discovery Mechanism

Automatically discovered via Integration Server discovery.

KPIs

Name	Description
ObjectStatus	<p>Represents the current status of the object.</p> <ul style="list-style-type: none"> ■ A value is "1" or "Online" indicates that the Scheduler is running ■ A value of "0" "Offline" indicates that the Scheduler is paused.
NumberOfTasks	Returns the number of user-defined tasks scheduled by the scheduler.

com.wm.isextdc.IsSchedulerTask

The KPIs in this component provide information about the execution state, time remaining for the next scheduled run, status, and targets of IS Scheduler Tasks that are created.

Note: IS Scheduler monitoring is supported only for Integration Server 8.2 and newer.

Discovery Mechanism

Automatically discovered via Integration Server discovery.

If auto-discovery is enabled and a Scheduler task is added or deleted using the Scheduler interface, the changes are not automatically reflected on the Monitored Components page. The user must re-discover the Integration Server to update the Monitored Components page display to reflect the Scheduler task additions or deletions.

KPIs

Name	Description
Execution State	<p>Represents the current execution state of the task:</p> <ul style="list-style-type: none"> ■ 0: The task is currently active. ■ 1: The task is currently running. ■ 2: The task is suspended.
NextRun	<p>Represents the number of seconds remaining for the scheduled task to run. The meaning of the values for active and suspended tasks varies for clustered or standalone systems.</p> <p>For a standalone server environment, the following values apply:</p> <ul style="list-style-type: none"> ■ For a task with an Active status, the value is the number of seconds remaining for the task to run. If the Scheduler is paused, it shows the number of seconds remaining to run. But when the value becomes < 0, it shows 0. ■ For a task with a Suspended status, the value is '-1'
	<p>For a Clustered environment the following values apply. Note that both parent and child tasks exists in this environment and the meaning of active and suspended statuses vary for these tasks:</p> <ul style="list-style-type: none"> ■ Parent Task:

Name	Description
	<ul style="list-style-type: none"> ■ For a task with an active status, no KPI value is sent. ■ For a task with a suspended status, no KPI value is sent. ■ Child Task: <ul style="list-style-type: none"> ■ For a task with an active status, the value is the number of seconds remaining for the task to run. If the Scheduler is paused, the value displays the number of seconds remaining to run. But, when the value becomes <0, it shows 0. ■ For a task, with a suspended status, the value is '-1'.
ObjectStatus	<p>Represents the current status of the object.</p> <ul style="list-style-type: none"> ■ A value is "1" or "Online" indicates that the execution state of the task is Active or Running. ■ A value of "0" "Offline" indicates that the execution state of the task is Suspended.

com.wm.isextdc.IsCluster

An Integration Server cluster is a collection of Integration Servers configured for load balancing and automatic failover. The KPIs in this component provide information about the number of servers and status of objects in a cluster.

- Note:** Monitoring of Coherence based IS Clusters is supported only for Integration Server versions 8.2.1 and 8.2.2. Monitoring of Terracotta based IS Clusters is supported only for Integration Server versions 9.0 and newer.
- Note:** While polling the cluster and its components for KPI values, if the Integration Server through which the connection is established is shut down, the KPI values continue to be collected only if another Integration Server that is part of the cluster is discovered (not necessarily monitored) by the same Infrastructure Data Collector.
- Note:** In some cases, KPI data collection for Integration Server Cluster (IS Cluster) components may be stopped. KPI data is not collected for IS Cluster

components if the Integration Server through which the connection is established is suspended or deleted. As a workaround, rediscover an Integration Server asset which is part of an IS Cluster configuration, if the connection has been suspended or deleted.

Discovery Mechanism

Automatically discovered via Integration Server discovery.

KPIs

Name	Description
NumberOfServers	Represents the number of Integration Server servers that are part of the cluster.
ObjectStatus	Represents the status of a component, whether it is Operational or Failed.

com.wm.isextdc.IsClusterServer

The KPIs in this component provide information about Integration Server cluster requests, service and system status, and component status.

Note: Monitoring of Coherence based IS Clusters is supported only for Integration Server versions 8.2.1 and 8.2.2. Monitoring of Terracotta based IS Clusters is supported only for Integration Server versions 9.0 and newer.

Discovery Mechanism

Automatically discovered via Integration Server discovery.

If auto-discovery is enabled and a Cluster server leaves or joins the cluster, the changes are not automatically reflected on the Monitored Components page. Users must re-discover the Integration Server to update the Monitored Components page display to reflect the cluster changes.

KPIs

Name	Description
AverageRequestDuration	Represents the average request duration in milliseconds.

Name	Description
NumberOfRequests	Represents the total number of requests on the Integration Server.
deltaNumberOfRequests	Delta of the total number of requests on the Integration Server.
CurrentRequestsPerMinute	Represents the current number of requests per minute on the Integration Server.
LifetimeNumberOfRequestsPerMinute	Represents the lifetime number of requests per minute on the Integration Server.
TotalServiceErrors	Represents the total number of service errors on the Integration Server.
deltaTotalServiceErrors	Delta of the total number of service errors on the Integration Server.
TotalSystemErrors	Represents the total number of system errors on the Integration Server.
deltaTotalSystemErrors	Delta of the total number of system errors on the Integration Server.
ObjectStatus	Represents the status of a component, whether it is Operational or Failed.

com.wm.isextdc.isJMSTrigger

The KPIs in this component provide information about JMS triggers on an Integration Server.

Note: Monitoring of IS JMS triggers is supported only for Integration Server versions 9.5.1.

Discovery Mechanism

Automatically discovered via Integration Server discovery. In order to monitor a newly added IS JMS trigger, you must re-discover the applicable Integration Server.

KPIs

Name	Description
ObjectStatus	Represents the status of the Integration Server component. 1 - operational 0 - failed.
JMSTriggerType	Represents the type of trigger that was invoked. 0 - standard 1 - soapJMS.
MaxConcurrent Threads	Represents the maximum number of messages that Integration Server can process concurrently on each connection for this trigger. The default is 1.
ConsumersCurrent ThreadCount	Represents the total thread number used by all consumers of this trigger.

com.wm.isextdc.isNativeTrigger

The KPIs in this component provide information about Broker local triggers on an Integration Server.

Note: Monitoring of IS Native Triggers is supported only for Integration Server versions 9.5.1.

Discovery Mechanism

Automatically discovered via Integration Server discovery. In order to monitor a newly added IS Native trigger, you must re-discover the applicable Integration Server.

KPIs

Name	Description
ObjectStatus	Represents the status of the Integration Server component. 1 - operational 0 - failed.

Name	Description
ProcessingState	Represents trigger processing status: 1 - active; 2 - active-temp; -1 - suspended; -2 -- suspended-temp.
RetrievalState	Represents trigger retrieval status: 1 - active; 2 - active-temp; -1 - suspended; -2 -- suspended-temp.
ActiveThreadCount	Represents the number of active threads.
MaxExecutionThreads	Represents the maximum number of documents that Integration Server can process concurrently.
QueueCapacity	Represents the maximum number of documents that Integration Server maintains in the queue for this trigger. The default is 10.
MaxExecutionThreadsThrottle	Represents the maximum number of the execution thread pool that can be used.
QueueCapacityThrottle	Represents the maximum size of the queue capacity that can be used.

com.wm.isextdc.IsJDBCFunctionalAlias

The KPIs in this component provide information about JDBC Pool functional aliases on an Integration Server.

Integration Server provides functions that correspond to most database components. For example, there is an IS Core Audit Log function that corresponds to the ISCoreAudit database component, and a Cross Reference function that corresponds to the Cross Reference database component. You direct each function to write to its database components by pointing the function at the appropriate JDBC connection pool. The JDBC database connection pools are defined via the Integration Server Administrator that Integration Server uses to connect to database components. Each pool specifies the connection to a database server that hosts database components.

Discovery Mechanism

Automatically discovered via Integration Server discovery. In order to monitor a newly added JDBC Pool functional aliases, you must re-discover the applicable Integration Server.

KPIs

Name	Description
Object Status	1 if the Functional Alias is connected to a pool and that pool can be pinged.
Assigned Connections	The number of connections currently being used. (TotalConnections - AvailableConnections)
Available Connections	Current number of available unused JDBC Connections
MinConnections	Minimum number of connections the pool must keep open at all times. If you use this pool alias for more than one function, each pool instance keeps the specified number of connections open. For example, if you specify keeping at least 3 connections open, and the IS Core Audit Log and the Document History database components both use this pool, the pool keeps a total of 6 connections open - 3 for the IS Core Audit Log pool instance and 3 for the Document History pool instance.
MaxConnections	Maximum number of connections the pools can have open at one time. Calculate this value as part of the total possible number of connections that could be opened simultaneously by all functions and applications that write to the database. Make sure the total number does not exceed the database's connection limit. If one of the applications opens more connections than the database allows, the database will reject subsequent requests for

Name	Description
	connections from any application. To continue the previous example, if Trading Networks also writes to the database and has a pool that could open up to 5 connections, you could specify only 17 as the maximum number of connections for the current pool. The IS Core Audit Log pool instance could use up to 17 connections, and the Document History pool instance could use the remaining 5 connections.
TotalConnections	The total number of currently active connections.
Connection Utilization	Percentage of Connections currently in use (AssignedConnections / TotalConnections).
AvailableConnectionsThreshold	When the available connections reach the specified percentage, a warning is issued in Integration Server.

webMethods Trading Networks KPI Definitions

Trading Networks KPI definitions are as follows:

KPI Definition

["com.wm.tnextdc.TNServer" on page 400](#)

["com.wm.tnextdc.TaskManager" on page 405](#)

["com.wm.tnextdc.TNDBConnectionPool" on page 406](#)

com.wm.tnextdc.TNServer

Reports product level information about Trading Networks Server. Contains all other manageable Trading Networks objects.

Discovery Mechanism

Automatically discovered via Integration Server discovery.

KPIs

Name	Description
string Build	Build number.
string ServerType	TNServer or TNPartner.
boolean OEM	True for OEM version, false is not.
int StatsPersistFrequency	Frequency, in seconds, with which Trading Networks writes statistics to the database.
struct DocStats	Statistics for documents that Trading Networks has processed.
long TotalCount	Total number of documents that Trading Networks has processed. If document processing causes the document to be transformed and resubmitted, this will count as more than one document.
long TotalSize	Total number of bytes of all documents that Trading Networks has processed.
long TotalSuccessCount	Total number of documents that Trading Networks has processed successfully.
long TotalErrorCount	Total number of documents that Trading Networks has not processed successfully.
long TotalProcessingDuration	Total amount of time (in milliseconds) for processing all documents. This value includes time between retries.
long StatsGatheringDuration	Total amount of time, in milliseconds, for gathering these document statistics.

Name	Description
struct TNSyncAsyncExecution	Statistics for documents that Trading Networks has processed by synchronous and asynchronous service execution.
long SuccessCount	Number of documents that Trading Networks has successfully processed by synchronous and asynchronous service execution.
long ErrorCount	Number of documents that Trading Networks has not successfully processed by synchronous and asynchronous service execution.
long TotalCount	Total number of documents that Trading Networks processed, successfully and unsuccessfully, by synchronous and asynchronous service execution.
long TotalSize	Total number of bytes for all documents in TotalCount. This value includes only documents that Trading Networks has saved to the database because Trading Networks only sets BizDocContentPart.length when saving a document.
long TotalProcessingDuration	Total amount of time, in milliseconds, that it took Trading Networks to process all documents in TotalCount.
struct TNImmediateReliableExecution	Statistics for service execution tasks.
long SuccessCount	Number of service execution tasks that the Trading Networks Task Manager has completed successfully.
long ErrorCount	Number of service execution tasks that the Trading Networks Task Manager has not completed successfully.
long TotalCount	Total number of service execution tasks that the Trading Networks Task

Name	Description
	Manager completed (successfully and unsuccessfully).
long TotalSize	Total number of bytes for all documents in TotalCount.
long TotalProcessingDuration	Total amount of time, in milliseconds, that it took the Trading Networks Task Manager to process all service execution tasks in TotalCount. This value includes processing time and time waiting for retries.
long RetryCount	Total number of retries for all service execution tasks in TotalCount.
struct TNImmediateReliableDelivery	Statistics for delivery tasks.
long SuccessCount	Number of documents the Trading Networks Task Manager has delivered successfully.
long ErrorCount	Number of documents the Trading Networks Task Manager has not delivered successfully.
long TotalCount	Total number of documents the Trading Networks Task Manager has delivered (successfully and unsuccessfully).
long TotalSize	Total number of bytes for all documents in TotalCount.
long TotalProcessingDuration	Total amount of time, in milliseconds, that it took the Task Manager to deliver all documents in TotalCount. This value includes processing time and time waiting for retries.
long RetryCount	Total number of retries for all documents in TotalCount.

Name	Description
struct TNScheduledDocProcessingStats	Statistics for documents delivered via batch.
long SuccessCount	Number of documents successfully delivered via batch.
long ErrorCount	Number of documents not successfully delivered via batch.
long TotalCount	Total number of documents delivered via batch (successfully and unsuccessfully).
long TotalSize	Total number of bytes for all documents in TotalCount.
long TotalProcessingDuration	Total amount of time, in milliseconds, spent delivering documents. This includes retry time, but not the time spent in queue.
long RetryCount	Total number of retries for all documents in TotalCount.
struct TNPollingDocProcessing	Statistics for document polling.
long SuccessCount	Number of documents successfully retrieved by partners via polling.
long ErrorCount	Number of documents not successfully retrieved by partners via polling.
long TotalCount	Total number of documents retrieved by partners via polling (successfully and unsuccessfully).
long TotalSize	Total number of bytes for all documents in TotalCount.
long TotalProcessingDuration	Total amount of time, in milliseconds, for all documents to be retrieved via polling.

Name	Description
int ObjectStatus	Indicates Operational or Failed status of the component.
boolean ContainsResourceMessage	Flag indicating a resource message, such as a log message or exception message, has been received from the monitored resource.
int MessageSeverity	Severity level of the resource message: Critical = 1 Major = 2 Minor = 3 Warning = 4 Info = 5 Ok = 6 Indeterminate = 7

com.wm.tnextdc.TaskManager

Reports information about the background thread that manages delivery jobs and routing jobs.

Discovery Mechanism

Automatically discovered via Integration Server discovery.

KPIs

Name	Description
int ThreadPoolSize	Maximum threads defined by Integration Server's thread pool size.
int MaxThreads	Maximum threads allowed for Trading Networks Task Manager.

Name	Description
int MaxThreadPoolPercentage	Maximum percentage of Integration Server thread pool allowed for use by the Trading Networks Task Manager.
int SweepTime	Task manager sweep time in seconds.
int UsedThreads	Number of processing tasks.
int AvailableThreads	The difference between MaxThreads and UsedThreads.
int ObjectStatus	Indicates Operational or Failed status of the component.
boolean ContainsResourceMessage	Flag indicating a resource message, such as a log message or exception message, has been received from the monitored resource.
int MessageSeverity	Severity level of the resource message: Critical = 1 Major = 2 Minor = 3 Warning = 4 Info = 5 Ok = 6 Indeterminate = 7

com.wm.tnextdc.TNDBCConnectionPool

Reports information about the database connection pool used by Trading Networks.

Discovery Mechanism

Automatically discovered via Integration Server discovery.

KPIs

Name	Description
int AvailableConnections	Number of unused connections in the pool.
int UsedConnections	Number of used connections in the pool.
int MaxConnections	Maximum number of active connections allowed for this connection pool.
long ConnectionTimeout	Number of milliseconds a connection is idle before being removed from the pool and closed.
string ConnectionPoolName	Connection pool name to uniquely identify Trading Networks database connections. For user to find information (for example, database URL or port).
int ObjectStatus	Indicates Operational or Failed status of the component.
boolean ContainsResourceMessage	Flag indicating a resource message, such as a log message or exception message, has been received from the monitored resource.
int MessageSeverity	Severity level of the resource message: Critical = 1 Major = 2 Minor = 3 Warning = 4 Info = 5 Ok = 6 Indeterminate = 7

My webMethods Server KPI Definitions

My webMethods Server KPI definitions are as follows:

KPI Definition
"com.softwareag.mws" on page 408
"com.softwareag.mws.directory" on page 412
"com.softwareag.mws.handler.sessionmanager" on page 413
"com.softwareag.mws.handler.authmanager" on page 414
"com.softwareag.mws.handler.presmanager" on page 415
"com.softwareag.mws.handler.dispatchmanager" on page 416
"com.softwareag.mws.taskengine" on page 417
"com.softwareag.mws.taskevents" on page 418
"com.softwareag.mws.portlet.action" on page 419
"com.softwareag.mws.portlet.render" on page 419
"com.softwareag.mws.portlet.pcalayout" on page 420
"com.softwareag.mws.cache" on page 421
"com.softwareag.mws.dirservicecache" on page 423
"com.softwareag.mws.connectionpool" on page 424
"com.softwareag.mws.database" on page 425

com.softwareag.mws

My webMethods Server is the core set of components and services required to host the Web interface for webMethods products, including Optimize.

My webMethods Server KPIs measure data collected since the last time the server was started. These KPIs can be found on the Analytics Overview page, listed under the component name “MywebMethodsServer.”

Discovery Mechanism

My webMethods Servers are discovered via Asset Discovery in Optimize.

KPIs

Name	Description
int Events.loginEvents	Total number of successful login attempts to My webMethods Server.
Events.deltaloginEvents	Delta of the total number of successful login attempts to My webMethods Server.
int Events.logoutEvents	Total number of events related to session logout and cleanup.
Events.deltalogoutEvents	Delta of the total number of events related to session logout and cleanup.
int Events.deleteEvents	Total number of items (for example, documents, folders, links, portlets, pages, users, groups, or roles) that were deleted from My webMethods Server.
Events.deltadeleteEvents	Delta of the total number of items (for example, documents, folders, links, portlets, pages, users, groups, or roles) that were deleted from My webMethods Server.
int Events.updateEvents	Total number of items that were updated in My webMethods Server.
Events.deltaupdateEvents	Delta of the total number of items that were updated in My webMethods Server.
int Events.readEvents	Total number of HTTP Read requests.

Name	Description
Events.deltareadEvents	Delta of the total number of HTTP Read requests.
int Events.createEvents	Total number of items that were created in My webMethods Server.
Events.deltacreateEvents	Delta of the total number of items that were created in My webMethods Server.
int Events.loginFailedEvents	Total number of unsuccessful login attempts to My webMethods Server.
Events.deltaloginFailedEvents	Delta of the total number of unsuccessful login attempts to My webMethods Server.
int rule_evaluation.numActions	Number of My webMethods Server rule evaluation actions.
float rule_evaluation.averageTime	Average amount of time, in milliseconds, that My webMethods Server took to evaluate rules.
long rule_evaluation.peakTime	Longest amount of time, in milliseconds, that My webMethods Server took to evaluate a rule.
long rule_evaluation.totalTime	Total amount of time, in milliseconds, that My webMethods Server took to evaluate rules.
rule_evaluation.deltanumActions	Delta of the number of My webMethods Server rule evaluation actions.
rule_evaluation.deltatotalTime	Delta of the total amount of time, in milliseconds, that My webMethods Server took to evaluate rules.
int Locale_Evaluation.numActions	Number of actions related to evaluating a client's locale.

Name	Description
long Locale_Evaluation.totalTime	Total amount of time, in milliseconds, that My webMethods Server spent evaluating locales.
long Locale_Evaluation.peakTime	Longest amount of time, in milliseconds, that My webMethods Server spent evaluating a locale.
float Locale_Evaluation.averageTime	Average amount of time, in milliseconds, that My webMethods Server spent evaluating locales.
Locale_Evaluation.deltanumActions	Delta of the number of actions related to evaluating a client's locale.
Locale_Evaluation.deltatotalTime	Delta of the total amount of time, in milliseconds, that My webMethods Server spent evaluating locales.
int Sessions.activeUsers	Number of users currently logged in to My webMethods.
int Sessions.totalSessions	Number of current sessions, including sessions in use by My webMethods Server System users.
float Http_Request.averageTime	Average amount of time, in milliseconds, that My webMethods Server spent processing HTTP requests.
long Http_Request.totalTime	Total amount of time, in milliseconds, that My webMethods Server spent processing HTTP requests.
int Http_Request.numActions	Number of HTTP requests received.
long Http_Request.peakTime	Longest amount of time, in milliseconds, that My webMethods Server spent processing HTTP requests.
Http_Request.deltanumActions	Delta of the number of HTTP requests received.

Name	Description
Http_Request.deltatotalTime	Delta of the total amount of time, in milliseconds, that My webMethods Server spent processing HTTP requests.
long Memory.maxMemory	Maximum amount of memory, in megabytes, available for use.
long Memory.freeMemory	Amount of free memory, in megabytes, that is available for use.

com.softwareag.mws.directory

The My webMethods Server system directory service contains information about My webMethods users, groups, and roles.

System directory service KPIs measure the key operations that track the lookup of users, groups, and roles and the evaluation of groups and roles since the last time the server was started. These KPIs can be found on the Analytics Overview page, listed under the component name "MWSDirectory."

Discovery Mechanism

Automatically discovered via My webMethods Server discovery.

KPIs

Name	Description
long peakTime	Longest amount of time, in milliseconds, that My webMethods Server spent performing a system directory service operation.
float averageTime	Average amount of time, in milliseconds, that My webMethods Server spent performing a system directory service operation.
int numActions	Number of system directory service operations that My webMethods Server performed.

Name	Description
long totalTime	Total amount of time, in milliseconds, that My webMethods Server spent performing a system directory service operation.
deltanumActions	Delta of the number of system directory service operations that My webMethods Server performed.
deltatotalTime	Delta of the total amount of time, in milliseconds, that My webMethods Server spent performing a system directory service operation.

com.softwareag.mws.handler.sessionmanager

The My webMethods Server session manager handles key aspects of the HTTP session lifecycle.

Session manager KPIs measure data collected since the last time the server was started. These KPIs can be found on the Analytics Overview page, listed under the component name "Handler.sessionManager."

Discovery Mechanism

Automatically discovered via My webMethods Server discovery.

KPIs

Name	Description
long peakTime	Longest amount of time, in milliseconds, that My webMethods Server maintained an HTTP session.
float averageTime	Average amount of time, in milliseconds, that My webMethods Server maintained an HTTP session.
int numActions	Number of HTTP requests that My webMethods Server processed.

Name	Description
long totalTime	Total amount of time, in milliseconds, that My webMethods Server maintained an HTTP session.
deltanumActions	Delta value of the number of HTTP requests that My webMethods Server processed.
deltatotalTime	Delta value of the total amount of time, in milliseconds, that My webMethods Server maintained an HTTP session.

com.softwareag.mws.handler.authmanager

The My webMethods Server authentication manager handles tasks related to obtaining authentication information from incoming HTTP requests.

Authentication manager KPIs measure data collected since the last time the server was started. These KPIs can be found on the Analytics Overview page, listed under the component name "Handler.authManager."

Discovery Mechanism

Automatically discovered via My webMethods Server discovery.

KPIs

Name	Description
long peakTime	Longest amount of time, in milliseconds, that My webMethods Server spent obtaining authentication information from an HTTP request.
float averageTime	Average amount of time, in milliseconds, that My webMethods Server spent obtaining authentication information from an HTTP request.
int numActions	Number of authentication requests that My webMethods Server processed.

Name	Description
long totalTime	Total amount of time, in milliseconds, that My webMethods Server spent obtaining authentication information from an HTTP request.
delta numActions	Delta value of the number of authentication requests that My webMethods Server processed.
deltatotalTime	Delta value of the total amount of time, in milliseconds, that My webMethods Server spent obtaining authentication information from an HTTP request.

com.softwareag.mws.handler.presmanager

The My webMethods Server presentation manager handles tasks related to rendering HTML webpages.

Presentation manager KPIs measure data collected since the last time the server was started. These KPIs can be found on the Analytics Overview page, listed under the component name "Handler.presManager."

Discovery Mechanism

Automatically discovered via My webMethods Server discovery.

KPIs

Name	Description
long peakTime	Longest amount of time, in milliseconds, that My webMethods Server spent rendering an HTML webpage.
float averageTime	Average amount of time, in milliseconds, that My webMethods Server spent rendering an HTML webpage.

Name	Description
int numActions	Number of HTML webpages that My webMethods Server rendered.
long totalTime	Total amount of time, in milliseconds, that My webMethods Server spent rendering an HTML webpage.
deltanumActions	Delta value of the number of HTML webpages that My webMethods Server rendered.
deltatotalTime	Delta value of the number of HTML webpages that My webMethods Server rendered.

com.softwareag.mws.handler.dispatchmanager

The My webMethods Server dispatch manager handles incoming HTTP requests.

Dispatch manager KPIs measure data collected since the last time the server was started. These KPIs can be found on the Analytics Overview page, listed under the component name "Handler.dispatchManager."

Discovery Mechanism

Automatically discovered via My webMethods Server discovery.

KPIs

Name	Description
long peakTime	Longest amount of time, in milliseconds, that My webMethods Server spent dispatching HTTP requests.
float averageTime	Average amount of time, in milliseconds, that My webMethods Server spent dispatching HTTP requests.
int numActions	Number of HTTP requests that My webMethods Server dispatched.

Name	Description
long totalTime	Total amount of time, in milliseconds, that My webMethods Server spent dispatching HTTP requests.
deltanumActions	Delta of the number of HTTP requests that My webMethods Server dispatched.
deltatotalTime	Delta of the total amount of time, in milliseconds, that My webMethods Server spent dispatching HTTP requests.

com.softwareag.mws.taskengine

The webMethods Task Engine manages activities related to tasks, such as queuing tasks, processing task rules, and searching and retrieving task data.

Task Engine KPIs measure data collected since the last time the server was started. These KPIs can be found on the Analytics Overview page, listed under the component name "MWSTaskEngine."

Discovery Mechanism

Automatically discovered via My webMethods Server discovery.

KPIs

Name	Description
long peakTime	Longest amount of time, in milliseconds, that Task Engine spent processing a task.
float averageTime	Average amount of time, in milliseconds, that Task Engine spent processing a task.
int numActions	Number of tasks that Task Engine processed.
long totalTime	Total amount of time, in milliseconds, that Task Engine spent processing tasks.

Name	Description
deltanumActions	Delta value of the number of tasks that Task Engine processed.
deltatotalTime	Delta value of the total amount of time, in milliseconds, that Task Engine spent processing tasks.

com.softwareag.mws.taskevents

The My webMethods Server task event KPIs relate to tasks that completed, failed, or were queued since the last time the server was started. These KPIs can be found on the Analytics Overview page, listed under the component name “MWSTaskEvents.”

Discovery Mechanism

Automatically discovered via My webMethods Server discovery.

KPIs

Name	Description
int completedEvents	Total number of tasks that My webMethods Server completed.
int errorEvents	Total number of tasks that did not complete successfully.
int queueEvents	Total number of tasks in the queue.
delta completedEvents	Delta of the total number of tasks that My webMethods Server completed.
deltaerrorEvents	Delta of the total number of tasks that did not complete successfully.
deltaqueueEvents	Delta of the total number of tasks that My webMethods Server completed.

com.softwareag.mws.portlet.action

The My webMethods Server portlet action manager handles requests from My webMethods Server system administrators to create custom portal pages using portlets.

Portlet action manager KPIs measure data collected since the last time the server was started. These KPIs can be found on the Analytics Overview page, listed under the component name "Portlet.Action."

Discovery Mechanism

Automatically discovered via My webMethods Server discovery.

KPIs

Name	Description
long peakTime	Longest amount of time, in milliseconds, that My webMethods Server spent handling portlet requests.
float averageTime	Average amount of time, in milliseconds, that My webMethods Server spent handling portlet requests.
int numActions	Number of portlet requests that My webMethods Server handled.
long totalTime	Total amount of time, in milliseconds, that My webMethods Server spent handling portlet requests.
delta numActions	Delta of the number of portlet requests that My webMethods Server handled.
delta totalTime	Delta of the total amount of time, in milliseconds, that My webMethods Server spent handling portlet requests.

com.softwareag.mws.portlet.render

The My webMethods Server portlet render manager handles the rendering of custom portal pages.

Portlet render manager KPIs measure data collected since the last time the server was started. These KPIs can be found on the Analytics Overview page, listed under the component name "Portlet.Render."

Discovery Mechanism

Automatically discovered via My webMethods Server discovery.

KPIs

Name	Description
long peakTime	Longest amount of time, in milliseconds, that My webMethods Server spent rendering custom portal pages.
float averageTime	Average amount of time, in milliseconds, that My webMethods Server spent rendering custom portal pages.
int numActions	Number of customer portal pages that My webMethods Server rendered.
long totalTime	Total amount of time, in milliseconds, that My webMethods Server spent rendering custom portal pages.
deltanumActions	Delta of the number of customer portal pages that My webMethods Server rendered.
deltatotalTime	Delta of the total amount of time, in milliseconds, that My webMethods Server spent rendering custom portal pages.

com.softwareag.mws.portlet.pcalayout

The My webMethods Server portlet layout manager handles the formatting of custom portal pages.

Portlet layout manager KPIs measure data collected since the last time the server was started. These KPIs can be found on the Analytics Overview page, listed under the component name "Portlet.PCALayout."

Discovery Mechanism

Automatically discovered via My webMethods Server discovery.

KPIs

Name	Description
long peakTime	Longest amount of time, in milliseconds, that My webMethods Server spent formatting custom portal pages.
float averageTime	Average amount of time, in milliseconds, that My webMethods Server spent formatting custom portal pages.
int numActions	Number of customer portal pages that My webMethods Server formatted.
long totalTime	Total amount of time, in milliseconds, that My webMethods Server spent formatting custom portal pages.
deltanumActions	Delta of the number of customer portal pages that My webMethods Server formatted.
deltatotalTime	Delta of the total amount of time, in milliseconds, that My webMethods Server spent formatting custom portal pages.

com.softwareag.mws.cache

The My webMethods Server cache KPIs measure aspects of cache used to hold My webMethods Server objects.

Cache KPIs measure data collected since the last time the server was started. These KPIs can be found on the Analytics Overview page, listed under the component name “MWSCache.”

Note: The following selected cache KPIs are not generating KPI instances when monitoring My webMethods Server 9.0: cacheHits, cacheMisses, deltaCacheHits, and deltaCacheMisses. The following selected KPIs are collecting: monitorscacheSize and maxSize.

Discovery Mechanism

Automatically discovered via My webMethods Server discovery.

KPIs

Name	Description
string cacheSize	Number of entries currently in the cache.
string maxSize	Maximum number of entries the cache can hold.
string cacheHits	Number of times a request to access the cache was processed successfully.
string cacheMisses	Number of times a request to access the cache was not successful, and My webMethods Server instead had to retrieve the data from its source (for example, RDBMS or LDAP).
deltacacheHits	Delta of the number of times a request to access the cache was processed successfully.
deltacacheMisses	Delta of the number of times a request to access the cache was not successful, and My webMethods Server instead had to retrieve the data from its source (for example, RDBMS or LDAP).

com.softwareag.mws.dirservicecache

The My webMethods Server directory service cache KPIs describe the types of data that the directory service caches in order to increase efficiency, reduce network traffic, and reduce load on external directory servers such as LDAP when retrieving information about My webMethods Server users, groups, and roles.

Directory service cache KPIs measure data collected since the last time the server was started. These KPIs can be found on the Analytics Overview page, listed under the component name "DirServiceCache."

Discovery Mechanism

Automatically discovered via My webMethods Server discovery.

KPIs

Name	Description
int authenticateCacheCapacity	Capacity of the authentication cache that My webMethods Server uses while performing authentication operations.
int authenticateCacheSize	Number of entries currently in the authentication cache.
int dnCacheCapacity	Capacity of the distinguished name cache.
int dnCacheSize	Number of entries currently in the distinguished name cache.
int uriCacheCapacity	Capacity of the uniform resource identifier (URI) cache.
int uriCacheSize	Number of entries currently in the URI cache.
int queryCacheCapacity	Capacity of the LDAP query cache.
int queryCacheSize	Number of entries currently in the LDAP query cache.

com.softwareag.mws.connectionpool

The My webMethods Server connection pool KPIs measure aspects of the connection pool used to establish connections to the server.

Connection pool KPIs measure data collected since the last time the server was started. These KPIs can be found on the Analytics Overview page, listed under the component name “MWSConnectionPool.”

Discovery Mechanism

Automatically discovered via My webMethods Server discovery.

KPIs

Name	Description
int initialSize	Initial size of the connection pool, as set on the Data Source Administration page in My webMethods.
boolean closed	Flag indicating whether the connection pool is closed.
int idleSize	Number of connections that can remain idle in the connection pool.
int checkinCalls	Number of times a connection was checked back into the connection pool after use.
int highWaterMark	Maximum number of created JDBC connections.
int createCalls	Number of new JDBC connections that have been created.
int maxSize	Maximum number of active connections allowed for this connection pool.
int poolSize	Current size of the connection pool.

Name	Description
int checkoutCalls	Number of connections that are currently checked out of the connection pool for use. These connections are considered "active" or "in use."
int checkedOutSize	Number of connections that can be checked out of the connection pool for use.
int minSize	Minimum number of active connections allowed for this connection pool.
deltacheckinCalls	Delta of the number of times a connection was checked back into the connection pool after use.
deltacheckoutCalls	Delta of the number of connections that are currently checked out of the connection pool for use. These connections are considered "active" or "in use."
deltacreateCalls	Delta of the number of new JDBC connections that have been created.

com.softwareag.mws.database

The My webMethods Server database KPIs measure My webMethods Server database query activity, including execution of SQL requests.

Database KPIs measure data collected since the last time the server was started. These KPIs can be found on the Analytics Overview page, listed under the component name "MWSDatabase."

Discovery Mechanism

Automatically discovered via My webMethods Server discovery.

KPIs

Name	Description
long peakTime	Longest amount of time, in milliseconds, that My webMethods Server spent processing database queries.
float averageTime	Average amount of time, in milliseconds, that My webMethods Server spent processing database queries.
int numActions	Number of database queries that My webMethods Server handled.
long totalTime	Total amount of time, in milliseconds, that My webMethods Server spent processing database queries.
deltanumActions	Delta of the number of database queries that My webMethods Server handled.
deltatotalTime	Delta of the total amount of time, in milliseconds, that My webMethods Server spent processing database queries.

Infrastructure Data Collector KPI Definitions

wm.infrastructuredc.InfrastructureDataCollector

The interface for the Infrastructure Data Collector.

Discovery Mechanism

Automatically discovered upon startup of the Infrastructure Data Collector.

Adapter KPI Definitions

Adapter KPI definitions are as follows:

KPI Definition

["wm.brokerdc.CustomAdapter" on page 427](#)

["com.wm.isextdc.AdapterService" on page 429](#)

["com.wm.artextdc.ARTConnection" on page 432](#)

["com.wm.artextdc.ARTListener" on page 434](#)

["com.wm.artextdc.ARTListenerNotification" on page 435](#)

["com.wm.artextdc.ARTPollingNotification" on page 438](#)

wm.brokerdc.CustomAdapter

A custom adapter is any webMethods Broker client that does not use the webMethodsAdapter Development Kit. A Custom Adapter component can be created for any webMethods Broker client.

Custom Adapter components are contained in webMethods Broker components.

Discovery Mechanism

Auto-discovery via webMethods Broker.

KPIs

Name	Description
string AdapterType	Type of resource to which the adapter is connected.
string Name	Adapter system name.
string ApplicationName	Application name for the adapter.
string ClientGroup	Client group name.
struct ClientStats	Statistics for the adapter's webMethods Broker client. Includes

Name	Description
	queue and document processing statistics.
date now	Current time on the webMethods Broker host.
date createTime	Time when the client was created.
Int cumulativeNumEventsDelivered	Number of events delivered by this client.
Int cumulativeNumEventsPublished	Total number of events published by this client.
Int cumulativeNumEventsRetrieved	Number of events retrieved by this client.
int numEventsPublished	Delta value of the number of events published by this client.
int numEventsDelivered	Delta value of the number of events delivered by this client.
int numEventsRetrieved	Delta value of the number of events retrieved by this client.
date lastEventPublishTime	Time when the last event was published by this client.
date lastEventDeliveryTime	Time when the last event was delivered by this client.
date lastEventQueueTime	Time when the last event was queued for this client.
date lastEventRetrieveTime	Time when the last event was retrieved by this client.
int queueLength	Number of events in the client queue.
int queueByteSize	Number of bytes worth of events in client queue.

Name	Description
date queueHighestLengthTime	Time when QueueHighestLength was last set, Forever (zero date and time) if not yet queued.
int ObjectStatus	Indicates Operational or Failed status of the component.
boolean ContainsResourceMessage	Flag indicating a resource message, such as a log message or exception message, has been received from the monitored resource.
int MessageSeverity	Severity level of the resource message: Critical = 1 Major = 2 Minor = 3 Warning = 4 Info = 5 Ok = 6 Indeterminate = 7

com.wm.isextdc.AdapterService

This interface defines KPIs for Integration Server adapter service components.

Discovery Mechanism

Automatically discovered via Integration Server.

KPIs

Name	Description
string UniversalNameLocalPart	Local part of the service universal name.

Name	Description
string UniversalNameNSPart	Namespace part of the service universal name.
boolean WebService	True if this is a Web service.
string SystemName	Service system name.
string Type	Output template type. Possible values are <code>Java</code> and <code>Flow</code> .
boolean Stateless	True for a stateless service.
string ACL	Access control list.
boolean EnforceInternalACL	True if ACLs are enforced for nested service invocations.
string OutputTemplate	Service output template file.
string TemplateType	Possible values are <code>XML</code> and <code>HTML</code> .
boolean Caching	True if caching is enabled.
boolean CachingPrefetch	True if caching prefetch is enabled.
int CachingExpiration	Result cache expiration interval, in seconds.
int PrefetchActivate	Cache prefetch for the service result.
long CurrentlyRunning	Current number of requests being processed.
long CumulativeRunTime	Cumulative run time for this service, in milliseconds.
long deltaCumulativeRunTime	Delta of the cumulative run time for this service, in milliseconds.
long Count	Total number of service requests.

Name	Description
long deltaCount	Delta of the total number of service requests.
long RetryCount	Number of the times the service failed and was retried.
long deltaRetryCount	Delta of the number of the times the service failed and was retried.
string JavaClass	Java class that implements this service.
string JavaMethod	Java method that implements this service.
date LastRan	Last time the service was invoked.
string LoadError	Last error loading the service.
long EMAILProtocolCumulativeRunTime	The cumulative run time for this service invoked by EMAIL Protocol.
long FTPProtocolCumulativeRunTime	The cumulative run time for this service invoked by FTP Protocol.
long HTTPProtocolCumulativeRunTime	The cumulative run time for this service invoked by HTTP Protocol.
long HTTPSProtocolCumulativeRunTime	The cumulative run time for this service invoked by HTTPS Protocol.
long SOAPProtocolCumulativeRunTime	The cumulative run time for this service invoked by SOAP Protocol.
long BROKEREXPORTProtocolCumulativeRunTime	The cumulative run time for this service invoked by Broker Transport Protocol.
long EMAILProtocolNumInvokes	The number of times this service has been invoked by EMAIL protocol.
long FTPProtocolNumInvokes	The number of times this service has been invoked by FTP protocol.

Name	Description
long HTTPProtocolNumInvokes	The number of times this service has been invoked by HTTP protocol.
long HTTPSProtocolNumInvokes	The number of times this service has been invoked by HTTPS protocol.
long SOAPProtocolNumInvokes	The number of times this service has been invoked by SOAP protocol.
long BROKEREXPORTProtocolNumInvokes	The number of times this service has been invoked by webMethods Broker Transport protocol.
int ObjectStatus	Indicates Operational or Failed status of the component.
boolean ContainsResourceMessage	Flag indicating a resource message, such as a log message or exception message, has been received from the monitored resource.
int MessageSeverity	Severity level of the resource message: <ul style="list-style-type: none"> ■ Critical = 1 ■ Major = 2 ■ Minor = 3 ■ Warning = 4 ■ Info = 5 ■ Ok = 6 ■ Indeterminate = 7

com.wm.artextdc.ARTConnection

The ART Connection component monitors a configured connection of an adapter instance to a resource instance.

Discovery Mechanism

Auto-discovered with discovery of IS.

KPIs

Name	Description
string AdapterType	Locale-sensitive information about the type of adapter used by this connection.
int TotalConnections	Total number of connections existing currently in a pool for the resource (busy connections and free connections).
int BusyConnections	Number of connections currently in use by adapter instances.
int FreeConnections	Number of connections currently available.
long CompletionCount	Number of successful connections to the resource.
long deltaCompletionCount	Delta of the number of successful connections to the resource.
long FailureCount	Number of failed connections to the resource.
long deltaFailureCount	Delta of the number of failed connections to the resource.
string ConnectionType	Locale-sensitive name for adapter connection type.
int ObjectStatus	Indicates Operational or Failed status of the component.
boolean ContainsResourceMessage	Flag indicating a resource message, such as a log message or exception message, has been received from the monitored resource.
boolean ResourceConnectionStatus	Indicates Operational or Failed status of a connection to a resource.

Name	Description
boolean ResourceConnectionError	Indicates whether an error was detected in between poll cycles.
int MessageSeverity	Severity level of the resource message: Critical = 1 Major = 2 Minor = 3 Warning = 4 Info = 5 Ok = 6 Indeterminate = 7

com.wm.artextdc.ARTListener

ART Listeners are configured to handle data pushed from a resource to an Integration Server.

Discovery Mechanism

Automatic discovery when Integration Server is discovered.

KPIs

Name	Description
string ConnectionName	Locale-sensitive name for the connection the listener uses to connect to the resource for polling.
date CreationDate	When the listener was created.
long ExecutionCount	Total number of times the notification has been executed.
long deltaExecutionCount	Delta of the total number of times the notification has been executed.

Name	Description
long ExecutionTime	Sum time of all notification executions that have occurred.
long FailureCount	Number of failed executions.
long deltaFailureCount	Delta of the number of failed executions.
int ObjectStatus	Indicates Operational or Failed status of the component.
boolean ContainsResourceMessage	Flag indicating a resource message, such as a log message or exception message, has been received from the monitored resource.
int MessageSeverity	Severity level of the resource message: Critical = 1 Major = 2 Minor = 3 Warning = 4 Info = 5 Ok = 6 Indeterminate = 7

com.wm.artextdc.ARTListenerNotification

ART Listener notifications are configured to handle events from a resource.

Discovery Mechanism

Automatic discovery when Integration Server is managed.

KPIs

Name	Description
double AverageNotificationTime	Average running time (totalTime/count).

Name	Description
long CompletionCount	Number of successful notifications.
long deltaCompletionCount	Delta of the umber of successful notifications.
long ConsecutiveCompletions	Number of consecutive completions without a failure.
long deltaConsecutiveCompletions	Delta of the number of consecutive completions without a failure.
long ConsecutiveFailures	Number of consecutive completions without a successful completion.
long deltaConsecutiveFailures	Delta of the number of consecutive completions without a successful completion.
date CreationDate	Creation date for the listener notification.
long FailureCount	Number of failed notifications.
long deltaFailureCount	Delta of the number of failed notifications.
string ListenerName	Locale-sensitive information about the listener with which this notification is associated.
long NotificationCount	Total number of times the listener notification has been executed.
long deltaNotificationCount	Delta of the total number of times the listener notification has been executed.
string NotificationType	Listener notification type (synchronous or asynchronous).

Name	Description
string PublishableDocumentName	Name of the document the asynchronous listener notification publishes.
string RequestDocumentName	Name of the document associated with listener notification that is input to service.
string ReplyDocumentName	Name of the document associated with service output.
string ServiceName	TotalNotificationTime divided by NotificationCount.
long TotalNotificationTime	Total number of milliseconds a notification has been running.
long deltaTotalNotificationTime	Delta of the total number of milliseconds a notification has been running.
int ObjectStatus	Indicates Operational or Failed status of the component.
boolean ContainsResourceMessage	Flag indicating a resource message, such as a log message or exception message, has been received from the monitored resource.
int MessageSeverity	Severity level of the resource message: Critical = 1 Major = 2 Minor = 3 Warning = 4 Info = 5 Ok = 6 Indeterminate = 7

com.wm.artextdc.ARTPollingNotification

ART Polling notifications use connections to poll the resource.

Discovery Mechanism

Automatically discovered via Integration Server.

KPIs

Name	Description
string ConnectionName	Locale-sensitive name for the connection the polling notification uses to connect to the resource for polling.
string PollingInterval	Interval at which to poll the resource.
boolean PollingOverlap	Whether to configure polling notifications to overlap.
boolean PollingImmediate	Polling configured to start immediately after notification is enabled.
date CreationDate	Duration of polling notification has been running.
long NotificationCount	Total number of times the polling notification has been executed.
long deltaNotificationCount	Delta of the total number of times the polling notification has been executed.
long CompletionCount	Number of successful invocations.
long deltaCompletionCount	Delta of the number of successful invocations.
long FailureCount	Number of failed invocations.
long deltaFailureCount	Delta of the number of failed invocations.

Name	Description
long ConsecutiveCompletions	Number of consecutive completions without a failure.
long deltaConsecutiveCompletions	Delta of the number of consecutive completions without a failure.
long ConsecutiveFailures	Number of consecutive failures without a successful completion.
long deltaConsecutiveFailures	Delta of the number of consecutive failures without a successful completion.
long TotalNotificationTime	Total number of milliseconds a notification has been running.
long deltaTotalNotificationTime	Delta of the total number of milliseconds a notification has been running.
double AverageNotificationTime	TotalNotificationTime divided by NotificationCount.
int ObjectStatus	Indicates Operational or Failed status of the component.
boolean ContainsResourceMessage	Flag indicating a resource message, such as a log message or exception message, has been received from the monitored resource.
int MessageSeverity	Severity level of the resource message: Critical = 1 Major = 2 Minor = 3 Warning = 4 Info = 5 Ok = 6 Indeterminate = 7

Terracotta KPI Definitions

The KPIs in this section enable you to monitor a Terracotta Management Server and by extension, individual Terracotta servers.

com.softwareag.tc.connectiongroup

This interface defines KPIs for Terracotta Server connection group components.

Discovery Mechanism

Automatically discovered via Terracotta server.

KPIs

Name	Description
boolean ObjectStatus	Indicates whether a component is online or offline.
long ServerSize	The number of servers using a connection.
int Client Size	The total number of clients using a connection.
int InfoEventCount	The total number of information level events.
int DebugEventCount	The total number of debug level events.
int WarnEventCount	The total number of warning level events.
int ErrorEventCount	The total number of error level events.
int CriticalEventCount	The total number of Critical level events.

com.softwareag.tc.server

This interface defines KPIs for Terracotta Server connection group components.

Discovery Mechanism

Automatically discovered via Terracotta server.

KPIs

Name	Description
boolean Object Status	The total number of information level events.
int LiveObjectCount	Shows the total number of live objects in the cluster, mirror group, server, or clients. If the trend for the total number of live objects goes up continuously, clients in the cluster will eventually run out of memory and applications might fail. Upward trends indicate a problem with application logic, garbage collection, or the tuning of one or more clients.
long WriteOperationRate	<p>Shows the number of completed writes (or mutations) in the TSA or selected server. Operations can include evictions and expirations. Large-scale eviction or expiration operations can cause spikes in the operations rate (see the corresponding evictions and expirations statistical graphs). This rate is low in read-mostly situations, indicating that there are few writes and little data to evict. If this number drops or deviates regularly from an established baseline, it might indicate issues with network connections or overloaded servers.</p> <p>When clients are selected, this statistic is reported as the Write Transaction Rate, tracking client-to-server write transactions.</p> <p>A measure of how many objects (per second) are being faulted in from the TSA in response to application requests. Faults from off-heap or disk occur when an object is not available in a server's on-heap cache. Flushes occur when the heap</p>

Name	Description
	<p>or off-heap cache must clear data due to memory constraints. Objects being requested for the first time, or objects that have been flushed from off-heap memory before a request arrives, must be faulted in from disk. High rates could indicate inadequate memory allocation at the server.</p>
long ReadOperationRate	<p>Shows the number of completed read operations (or mutations) in the TSA or selected server. Operations can include evictions and expirations. Large-scale eviction or expiration operations can cause spikes in the operations rate (see the corresponding evictions and expirations statistical graphs). This rate is low in read-mostly situations, indicating that there are few writes and little data to evict. If this number drops or deviates regularly from an established baseline, it might indicate issues with network connections or overloaded servers.</p> <p>When clients are selected, this statistic is reported as the Read Transaction Rate, tracking client-to-server read transactions. A measure of how many objects (per second) are being faulted in from the TSA in response to application requests. Faults from off-heap or disk occur when an object is not available in a server's on-heap cache. Flushes occur when the heap or off-heap cache must clear data due to memory constraints. Objects being requested for the first time, or objects that have been flushed from off-heap memory before a request arrives, must be faulted in from disk. High rates could indicate inadequate memory allocation at the server.</p>
long EvictionRate	<p>Shows the number of entries being evicted from the cluster, mirror group, or server. The number of elements being evicted per second from the cache.</p>

Name	Description
	Evicted elements are expired or evicted according to a usage algorithm when size limits are exceeded.
long ExpirationRate	Shows the number of expired entries found (and being evicted) on the TSA, mirror group, or server. The number of elements per second reaching expiration in the cache. Expired elements are not automatically evicted.
int DataUsedSize	Shows the amount of the data storage that is currently in use.
int DataMaxSize	Shows the total amount of data storage. This is the configured amount (dataStorage size in the tc-config.xml) that can be stored in BigMemory, both in off-heap DRAM and in any SSD flash drive, if BigMemory Hybrid is configured.
int OffheapUsedSize	Shows the amount of off-heap memory currently in use (containing data).
int OffheapReservedSize	Shows the amount of off-heap memory that is currently available.
int OffheapMaxSize	Shows the configured maximum amount of off-heap memory (offheap size in the tc-config.xml).

Terracotta Connection

Connection to the Terracotta Management Server or Terracotta Server through which infradc_service script can discover the other Terracotta components.

Name	Description
boolean ObjectStatus	Indicates whether a component is online or offline.

Ehcache

Ehcache is a standards-based cache for boosting performance, offloading your database, and simplifying scalability.

Name	Description
boolean ObjectStatus	Indicates whether a component is online or offline.
int Secured	Indicates secured status. 1 equals yes, 0 equals no.
int SSLEnabled	Indicates if SSL is enabled. 1 equals yes, 0 equals no.
int NeedClientAuth	Indicates if client authentication is needed. 1 equals yes, 0 equals no.
int SampleHistorySize	Historic information samples of cache size.
int SampleIntervalSeconds	Historic frequency information in seconds.

TSA

The Terracotta Server Array (TSA) provides the platform for Terracotta products and the backbone for Terracotta clusters.

Name	Description
boolean ObjectStatus	Indicates whether TSA is online or offline.
long ServerSize	Server size in bytes.
long ActiveServerSize	Active server size in bytes.
int ClientSize	Client size in bytes.

Name	Description
int ErrorEventCount	Count of all error events.
int CriticalEventCount	Count of all critical events.
int InfoEventCount	Count of all informational events.
int WarnEventCount	Count of all warning events.
int DebugEventCount	Count of all debug events.
int Secured	Indicates secured status. 1 equals yes, 0 equals no.
int SSLEnabled	Indicates if SSL is enabled. 1 equals yes, 0 equals no.
int NeedClientAuth	Indicates if client authentication is needed. 1 equals yes, 0 equals no.
int SampleHistorySize	Historic information samples of cache size.
int SampleIntervalSeconds	Historic frequency information in seconds.

TSA Server

Server instances which are part of a cluster TSA.

Name	Description
boolean ObjectStatus	Indicates whether the TSA server is online or offline.
long WriteOperationRate	Rate at witch operations are written in seconds.
long ReadOperationRate	Rate at witch operations are read in seconds.

Name	Description
long EvictionRate	Rate of evictions in seconds.
long ExpirationRate	Rate of expirations in seconds.
int LiveObjectCount	Number of live objects count.
int DataUsedSize	Size of data used in bytes.
int DataReservedSize	Size of data reserved in bytes.
int DataMaxSize	Size of data max allowed in bytes.
int OffheapUsedSize	Size of Off heap space used.
int OffheapReservedSize	Size of Off heap reserved space.
int OffheapMaxSize	Size of Off heap max space allowed.

CacheManager

A container for Ehcache, which maintains all aspects of caches life cycle. The creation and removal of caches and the access to them is controlled by a named Cache Manager.

Name	Description
boolean ObjectStatus	Indicates whether the CacheManager is online or offline.
int CacheAverageGetTime	Average time it takes to get the cache.
int CacheAverageSearchTime	Average time it takes to search the cache.
long CacheSearchRate	Rate at which the cache is searched.
long CacheHitRate	Rate at which the cache objects are found.

Name	Description
long CacheOnDiskHitRate	Rate at which the cache on disk objects are found.
long CacheOffHeapHitRate	Rate at which the cache Off Heap objects are found.
long CacheInMemoryHitRate	Rate at which the cache in memory objects are found.
long CacheMissRate	Rate at which the cache objects are not found.
long CacheOnDiskMissRate	Rate at which the cache objects are not found.
long CacheOffHeapMissRate	Rate at which the cache Off Heap objects are not found.
long CacheInMemoryMissRate	Rate at which the cache in memory objects are not found.
int CachePutrate	Rate at which the objects are written to cache.
long CacheUpdateRate	Rate at which the objects are updated in cache.
long CacheRemoveRate	Rate at which the objects are removed from cache.
long CacheEvictionRate	Rate at which the objects are evicted from cache.
long CacheExpirationRate	Rate at which the objects expire in cache.
long TransactionCommitRate	Rate at which the objects are committed to cache.
int TransactionCommittedCount	Object count committed to cache.

Name	Description
long TransactionRollbackRate	Rate at which transactions are rolled back from cache.
int TransactionRolledBackCount	Object count of rolled back transactions.
int TransactionTimeOutCount	Object count of time out transactions.
int WriteQueueLength	Queue length written to cache.

Cache

Terracotta Cache is Ehcache, which has elements and is managed by cache manager.

Name	Description
boolean ObjectStatus	Indicates whether the cache is online or offline.
int Size	Size of cache in bytes.
int OnDiskSize	Size of cache on disk in bytes.
int InMemorySize	Size of cache in memory.
int OffHeapSize	Size of Off Heap space in bytes.
int LocalDiskSize	Local disk size of cache in bytes.
int LocalOffHeapSize	Size of Local Off Heap space in bytes.
int LocalHeapSize	Size of Local Heap space in bytes.
int AverageGetTime	Average time it takes to process a get cache object request.
int AverageSearchTime	Average time to process a search object request.

Name	Description
long CacheSearchRate	Rate at which it takes to search the cache.
long CacheHitRate	Rate at which objects are successfully found in cache.
long CacheOnDiskHitRate	Rate at which objects are successfully found on disk cache.
long CacheInMemoryHitRate	Rate at which objects are successfully found in memory cache.
long CacheOffHeapHitRate	Rate at which objects are successfully found in Off Heap cache.
int CacheHitRatio	Ratio at which objects are successfully found in cache.
int CacheHitCount	Count of objects that are successfully found in cache.
int OnDiskHitCount	Count of objects that are successfully found on disk cache.
int OffHeapHitCount	Count of objects that are successfully found in Off Heap cache.
int InMemoryHitCount	Count of objects that are successfully found in memory cache.
long CacheMissRate	Rate at which objects are not found in memory cache.
long CacheOnDiskMissRate	Rate at which objects are not found on disk cache.
long CacheInMemoryMissRate	Rate at which objects are not found in memory cache.
long CacheOffHeapMissRate	Rate at which objects are not found in Off Heap cache.

Name	Description
long RemoveElementMissRate	Rate at which elements are not removed from cache.
long PutIfAbsentMissRate	Rate at which elements not found in cache are not written to cache.
long ReplaceOneArgMissRate	Rate at which elements with one argument are not replaced.
long ReplaceTwoArgMissRate	Rate at which elements with two arguments are not replaced.
int CacheMissCount	Count of objects not found in cache.
int OnDiskMissCount	Count of objects not found on disk.
int OffHeapMissCount	Count of objects not found in Off Heap cache.
int InMemoryMissCount	Count of objects not found in memory cache.
int RemoveElementMissCount	Count of objects not removed from cache.
int PutIfAbsentMissCount	Count of objects not found in cache, not written to cache.
int ReplaceOneArgMissCount	Count of objects with one argument not replaced.
int ReplaceTwoArgMissCount	Count of objects with two arguments not replaced.
int CacheMissCountExpired	Count of objects which should have expired from cache but didn't.
long CachePutRate	Rate at which objects are written to cache.

Name	Description
long CacheRemoveRate	Rate at which objects are removed from cache.
long CacheUpdateRate	Rate at which objects in cache are updated
long CacheExpirationRate	Rate at which objects in cache have expired.
long CacheEvictionRate	Rate at which objects in cache have been evicted.
int PutCount	Count of objects written to cache.
int UpdateCount	Count of objects updated in cache.
int RemovedCount	Count of objects removed from cache.
int ExpiredCount	Count of objects expired in cache.
int EvictedCount	Count of objects evicted from cache.
long RemoveElementSuccessRate	Rate at which elements are successfully removed from cache.
int RemoveElementSuccessCount	Count of elements successfully removed from cache.
long PutIfAbsentSuccessRate	Rate at which objects not found in cache, are written to cache.
int PutIfAbsentSuccessCount	Count of objects missing from cache are written to cache.
long ReplaceOneArgSuccessRate	Rate at which objects with one argument are successfully replaced.
int ReplaceOneArgSuccessCount	Count of objects with one argument successfully replaced.

Name	Description
long ReplaceTwoArgSuccessRate	Rate at which objects with two arguments are successfully replaced.
int ReplaceTwoArgSuccessCount	Count of objects with two arguments successfully replaced.
long NonStopSuccessRate	Rate at which the cache continues to process certain operations when offline or disconnected.
int NonStopSuccessCount	Count of operations the cache continues to process when offline or disconnected.
long NonStopFailureRate	Rate at which the cache doesn't continue to process certain operations when offline or disconnected.
int NonStopFailureCount	Count of operations the cache doesn't continue to process when offline or disconnected.
int NonstopTimeoutRatio	Ratio at which operations have timed out when the cache continues to process when offline or disconnected.
long NonStopTimeoutRate	Rate at which operations have timed out when the cache continues to process when offline or disconnected.
int NonStopTimeoutCount	Count of operations which have timed out when the cache doesn't continue to process when offline or disconnected.
long NonStopRejoinTimeoutRate	Rate at which operations have timed out and rejoined when the cache continues to process when offline or disconnected.
int NonStopRejoinTimeoutCount	Count of operations which have timed out and rejoined when the cache

Name	Description
	doesn't continue to process when offline or disconnected.
long TransactionCommitRate	Rate at which transactions are committed to cache.
int XaCommitCount	Count of XA mode transactions written to cache.
long TransactionRollbackRate	Rate at which transactions are rollback in cache.
int XaRollbackCount	Count of XA mode transactions rolled back in cache.
int XaRecoveredCount	Count of XA mode transactions recovered in cache.
int WriterQueueLength	Maximum elements that can be queued for processing by the write-behind writer.
int WriterMaxQueueSize	Maximum size of elements that can be queued for processing by the write-behind writer.
int WriterConcurrency	Sets whether to notify listeners when an exception occurs on a writer operation.
int CacheClusterOnlineCount	Count of all cache clusters which are online.
int CacheClusterOfflineCount	Count of all cache clusters which are offline.
int CacheClusterRejoinCount	Count of all cache clusters which where rejoined.
int LocalHeapCountBased	Count of all local heaps.

Name	Description
int OverflowToDisk	Overflowed cache data will be written to disk: indicator - on/off.
int OverflowToOffHeap	Overflowed cache data will be written to off heap: indicator - on/off.
int MaxElementsOnDisk	Total maximum elements allowed on disk - value.
int MaxEntriesInCache	Total maximum entries allowed in cache - value.
int MaxEntriesLocalDisk	Total maximum entries allowed on local disk - value.
int MaxEntriesLocalHeap	Total maximum entries allowed in local heap- value.
int MaxBytesLocalDisk	Total maximum bytes allowed on local disk - value.
int MaxBytesLocalOffHeap	Total maximum bytes allowed in off heap - value.
int MaxBytesLocalHeap	Total max bytes allowed in local heap - value.

Presto KPI Definitions

The KPIs in this section enable you to monitor Presto servers.

com.softwareag.presto

Connects over HTTP to the web server and extracts the KPIs from web services, using Presto authentication.

Discovery Mechanism

Presto KPIs are discovered via Asset Discovery in Optimize.

KPIs

Name	Description
boolean ObjectStatus	Status of Presto via presto ping web service.
long JvmCommittedMemory	The committed memory in the JVM in kilobytes.
long JvmMaxMemory	The maximum memory in the JVM in kilobytes.
long JvmUsedMemory	The used memory in the JVM in kilobytes.
int MaxCurrentThreads	The peak number of threads in the JVM.
int NoCurrentThreads	The number of current threads in the JVM.
long NoOfAppsInDepot	The number of approved apps in Presto.
long NoOfAppsInHub	The number of apps in Presto.
long NoOfPendingAppsInDepot	The number of apps pending approval in Presto.
long NoOfAtoms	The number of atom mashups and mashables in Presto.
long NoOfDataSources	The number of data sources defined in Presto.
long NoOfDatabases	The number of database mashups and mashables in Presto.
long NoOfJDBCDrivers	The number of JDBC drivers defined in Presto.
long NoOfMacros	The number of macros in Presto.

Name	Description
long NoOfMashups	The number of mashups in Presto.
long NoOfRESTs	The number of REST mashups and mashables in Presto.
long NoOfRSS	The number of RSS mashups and mashables in Presto.
long NoOfSharePoints	The number of Sharepoint mashups and mashables in Presto.
long NoOfSpreadsheets	The number of Spreadsheet mashups and mashables in Presto.
long NoOfWSDLs	The number of WSDL mashups and mashables in Presto.
long NoOfXMLs	The number of XML mashups and mashables in Presto.

B Built-In Rules

■ Overview	458
■ Infrastructure Data Collector Rules	458
■ ART Rules	459
■ Broker Server Rules	464
■ Custom Adapter Rules	467
■ Territory Rules	467
■ Gateway Rules	468
■ Analytic Engine Rules	468
■ Integration Server Rules	470
■ IS Package Rules	478
■ IS Service Rules	479
■ My webMethods Server Rules	481
■ Trading Networks Rules	491
■ Terracotta Rules	494
■ Presto Rules	496
■ Universal Messaging Rules	497

Overview

The following pre-defined rules are added automatically to the Optimize Rule List the first time the Infrastructure Data Collector is started in your environment.

These rules are provided as examples. They may need to be modified to provide the desired results for your environment.

Infrastructure Data Collector Rules

Infrastructure Data Collector About to Run Out of Memory

Name	Infrastructure Data Collector About to Run Out of Memory	
Type	KPI	
Description	Infrastructure Data Collector about to run out of memory	
Expression	$\left(\frac{\text{InfrastructureDataCollector/Free Memory by Infrastructure Data Collector/Reading_Value}}{\text{InfrastructureDataCollector/Total Memory by Infrastructure Data Collector/Reading_Value}} \right) < (0.2)$	
Related	Component	KPIs
	Infrastructure Data Collector	FreeMemory TotalMemory

Infrastructure Data Collector Thread Count Trend

Name	Infrastructure Data Collector Thread Count Trend	
Type	KPI	
Description	Infrastructure Data Collector Thread Count trending up	
Expression	$\text{InfrastructureDataCollector/ThreadCount by Infrastructure Data Collector/Diagnosis_Trend_Up} = (1.0)$	

Related	Component	KPI
	Infrastructure Data Collector	ThreadCount

Infrastructure Data Collector Failure Status

Name	Infrastructure Data Collector Failure Status	
Type	Event	
Description	Infrastructure Data Collector status is failed.	
Expression	(InfrastructureDataCollector/Object Status by Infrastructure Data Collector/Reading_Value) = (0)	
Related	Component	KPI
	Infrastructure Data Collector	ObjectStatus

ART Rules

ART Adapter Service Critical Resource Message

Name	ART Adapter Service Critical Resource Message	
Type	KPI	
Description	ART Adapter Service Critical Resource Message received.	
Expression	(ArtAdapterService/Contains Resource Message by ART Adapter Service/Reading_Value) = (1) AND (ArtAdapterService/MessageSeverity by ART Adapter Service/Reading_Value) = (1)	
Related	Component	KPIs
	ART Adapter Service	ContainsResourceMessage MessageSeverity

ART Adapter Service Failure Status

Name	ART Adapter Service Failure Status	
Type	Event	
Description	ART Adapter Service status is failed.	
Expression	(ArtAdapterService/Object Status by ART Adapter Service/ Reading_Value) = (0)	
Related	Component	KPI
	ART Adapter Service	ObjectStatus

ART Connection Critical Resource Message

Name	ART Connection Critical Resource Message	
Type	KPI	
Description	ART Connection Critical Resource Message received.	
Expression	(ArtConnection/Contains Resource Message by ART Connection/ Reading_Value) = (1) AND (ArtConnection/MessageSeverity by ART Connection/Reading_Value) = (1)	
Related	Component	KPIs
	ART Connection	ContainsResourceMessage MessageSeverity

ART Connection Failure Status

Name	ART Connection Failure Status	
Type	Event	

Description	ART Connection status is failed.	
Expression	(ArtConnection/Object Status by ART Connection/Reading_Value) = (0)	
Related	Component	KPI
	ART Connection	ObjectStatus

ART Connection Resource Failure

Name	ART Connection Resource Failure	
Type	Event	
Description	Connection to adapter resource is down.	
Expression	(ArtConnection/Object Status by ART Connection/Reading_Value) = (1) AND (ArtConnection/Resource Connection Status by ART Connection/Reading_Value) = (0)	
Related	Component	KPI
	ART Connection	ObjectStatus ResourceConnectionStatus

ART Listener Critical Resource Message

Name	ART Listener Critical Resource Message	
Type	KPI	
Description	ART Listener Critical Resource Message received.	
Expression	(ArtListener/Contains Resource Message by ART Listener/Reading_Value) = (1) AND (ArtListener/MessageSeverity by ART Listener/Reading_Value) = (1)	
Related	Component	KPIs

ART Listener	ContainsResourceMessage MessageSeverity
--------------	--

ART Listener Failure Status

Name	ART Listener Failure Status	
Type	Event	
Description	ART Listener status is failed.	
Expression	(ArtListener/Object Status by ART Listener/Reading_Value) = (0)	
Related	Component	KPI
	ART Listener	ObjectStatus

ART Listener Notification Critical Resource Message

Name	ART Listener Notification Critical Resource Message	
Type	KPI	
Description	ART Listener Notification Critical Resource Message received.	
Expression	(ArtListenerNotification/Contains Resource Message by ART Listener Notification/Reading_Value) = (1)AND (ArtListenerNotification/Message Severity by ART Listener Notification/Reading_Value)= (1)	
Related	Component	KPIs
	ART Listener Notification	ContainsResourceMessage MessageSeverity

ART Listener Notification Failure Status

Name	ART Listener Notification Failure Status	
Type	Event	
Description	ART Listener Notification status is failed.	
Expression	(ArtListenerNotification/Object Status by ART Listener Notification/Reading_Value) = (0)	
Related	Component	KPI
	ART Listener Notification	ObjectStatus

ART Polling Notification Critical Resource Message

Name	ART Polling Notification Critical Resource Message	
Type	KPI	
Description	ART Polling Notification Critical Resource Message received.	
Expression	(ArtPollingNotification/Contains Resource Message by ART Polling Notification/Reading_Value) = (1)AND (ArtPollingNotification/Message Severity by ART Polling Notification/Reading_Value)=(1)	
Related	Component	KPIs
	ART Polling Notification	ContainsResourceMessage MessageSeverity

ART Polling Notification Failure Status

Name	ART Polling Notification Failure Status	
------	---	--

Type	Event				
Description	ART Polling Notification status is failed.				
Expression	(ArtPollingNotification/Object Status by ART Polling Notification/Reading_Value) = (0)				
Related	<table border="1"> <thead> <tr> <th>Component</th> <th>KPI</th> </tr> </thead> <tbody> <tr> <td>ART Polling Notification</td> <td>ObjectStatus</td> </tr> </tbody> </table>	Component	KPI	ART Polling Notification	ObjectStatus
Component	KPI				
ART Polling Notification	ObjectStatus				

Broker Server Rules

Broker Server Critical Resource Message

Name	Broker Server Critical Resource Message				
Type	KPI				
Description	Broker Server Critical Resource Message received.				
Expression	(Broker Server/Contains Resource Message by Broker Server/Reading_Value) = (1) AND (BrokerServer/Message Severity by Broker Server/Reading_Value) = (1)				
Related	<table border="1"> <thead> <tr> <th>Component</th> <th>KPIs</th> </tr> </thead> <tbody> <tr> <td>Broker Server</td> <td>ContainsResourceMessage MessageSeverity</td> </tr> </tbody> </table>	Component	KPIs	Broker Server	ContainsResourceMessage MessageSeverity
Component	KPIs				
Broker Server	ContainsResourceMessage MessageSeverity				

Broker Server Failure Status

Name	Broker Server Failure Status
Type	Event
Description	Broker Server status is failed.

Expression	(Broker Server/Object Status by Broker Server/Reading_Value) = (0)
------------	--

Related	Component	KPI
	Broker Server	ObjectStatus

Broker Server Low Guaranteed Disk Space

Name	Broker Server Low Guaranteed Disk Space
------	---

Type	KPI
------	-----

Description	Broker Server guaranteed disk space is low.
-------------	---

Expression	((Broker Server/SystemStats.GuaranteedSpaceUsed by Broker Server/Reading_Value) / (Broker Server/SystemStats.GuaranteedSpaceMaxby Broker Server/Reading_Value)) >= (.9)
------------	---

Related	Component	KPIs
	Broker Server	SystemStats.GuaranteedSpaceUsed SystemStats.GuaranteedSpaceMax

Broker Rules

Broker Critical Resource Message

Name	Broker Critical Resource Message
------	----------------------------------

Type	KPI
------	-----

Description	Broker Critical Resource Message received.
-------------	--

Expression	(Broker/Contains ResourceMessage by Broker/Reading_Value) = (1) AND (Broker/Message Severity by Broker/Reading_Value) = (1)
------------	---

Related	Component	KPIs
	Broker	ContainsResourceMessage MessageSeverity

Broker Failure Status

Name	Broker Failure Status	
Type	Event	
Description	Broker object status is failed	
Expression	(Broker/Object Status by Broker/Reading_Value) = (0)	
Related	Component	KPI
	Broker	ObjectStatus

Document Type Notification Critical Resource Message

Name	Document Type Notification Critical Resource Message	
Type	KPI	
Description	Document Type Critical Resource Message received.	
Expression	(DocumentType/Contains Resource Message by Document Type/Reading_Value) = (1) AND (DocumentType/MessageSeverity by Document Type/Reading_Value) = (1)	
Related	Component	KPIs
	Document Type	ContainsResourceMessage MessageSeverity

Custom Adapter Rules

Custom Adapter Failure Status

Name	Custom Adapter Failure Status	
Type	Event	
Description	Custom Adapter object status is failed	
Expression	(CustomAdapter/Object Status by Custom Adapter/ Reading_Value) = (0)	
Related	Component	KPI
	Custom Adapter	ObjectStatus

Territory Rules

Territory Critical Resource Message

Name	Territory Critical Resource Message	
Type	KPI	
Description	Territory Critical Resource Message received.	
Expression	(Territory/Contains Resource Message by Territory/ Reading_Value) = (1) AND (Territory/Message Severity by Territory/Reading_Value) = (1)	
Related	Component	KPIs
	Territory	ContainsResourceMessage MessageSeverity

Territory Failure Status

Name	Territory Failure Status	
Type	KPI	
Description	Territory Object Status is failed.	
Expression	$(\text{Territory}/\text{Object Status by Territory}/\text{Reading_Value}) = (0)$	
Related	Component	KPIs
	Territory	ObjectStatus

Gateway Rules

Gateway Failure Status

Name	Gateway Failure Status	
Type	KPI	
Description	Gateway Object Status is failed.	
Expression	$(\text{Gateway}/\text{Object Status by Gateway}/\text{Reading_Value}) = (0)$	
Related	Component	KPIs
	Gateway	ObjectStatus

Analytic Engine Rules

Analytic Engine About to Run Out of Memory

Name	Analytic Engine About to Run Out of Memory
------	--

Type	KPI	
Description	The Analytic Engine has less than 2% free memory remaining in the virtual machine.	
Expression	AnalyticEngine/ConsumedMemory by Analytic Engine/Reading_Value) / (AnalyticEngine/MaximumMemory by Analytic Engine/Reading_Value)) > (.98)	
Related	Component	KPIs
	Analytic Engine	ConsumedMemory MaximumMemory

Error in Analytic Engine

Name	Error in Analytic Engine	
Type	Event	
Description	An error has occurred in the analytic engine.	
Expression	(AnalyticEngine/Errors by Analytic Engine/Reading_Value) > (0)	
Related	Component	KPIs
	Analytic Engine	Errors

Thread Leak

Name	Thread Leak	
Type	KPI	
Description	A thread leak is occurring.	
Expression	(AnalyticEngine/ThreadCount by Analytic Engine/Reading_Value) > (250)	

Related	Component	KPIs
	Analytic Engine	ThreadCount

Integration Server Rules

Integration Server Completed Requests Abnormal

Name	Integration Server Completed Requests Abnormal	
Type	KPI	
Description	Integration Server completed requests abnormal.	
Expression	(IntegrationServer/NumCompletedRequests by Integration Server/Diagnosis_Above_Normal) > (2.0) OR (IntegrationServer/NumCompletedRequests by Integration Server/Diagnosis_Below_Normal) > (2.0)	
Related	Component	KPI
	Integration Server	NumCompletedRequests

Integration Server Critical Resource Message

Name	Integration Server Critical Resource Message	
Type	KPI	
Description	Integration Server Critical Resource Message received.	
Expression	(IntegrationServer/Contains Resource Message by Integration Server/Reading_Value) = (1) AND (IntegrationServer/MessageSeverity by Integration Server/Reading_Value) = (1)	
Related	Component	KPIs
	Integration Server	ContainsResourceMessage

MessageSeverity

Integration Server Failure Status

Name	Integration Server Failure Status	
Type	Event	
Description	Integration Server status is failed.	
Expression	$(\text{IntegrationServer}/\text{Object Status by Integration Server}/\text{Reading_Value}) = (0)$	
Related	Component	KPI
	Integration Server	ObjectStatus

Integration Server Used Memory Too Big

Name	Integration Server Used Memory Too Big	
Type	KPI	
Description	Integration Server issuing too much memory.	
Expression	$((\text{IntegrationServer}/\text{UsedMemory by Integration Server}/\text{Reading_Value}) / (\text{IntegrationServer}/\text{TotalMemory by Integration Server}/\text{Reading_Value})) \geq (0.8)$	
Related	Component	KPIs
	Integration Server	UsedMemory TotalMemory

IsBrokerTransport Critical Resource Message

Name	IsBrokerTransport Critical Resource Message
------	---

Type	KPI	
Description	ISBrokerTransport Critical Resource Message received.	
Expression	(IsBrokerTransport/Contains Resource Message by IS Broker Transport/Reading_Value) = (1) AND (IsBrokerTransport/Message Severity by IS Broker Transport/Reading_Value) = (1)	
Related	Component	KPIs
	IS Broker Transport	ContainsResourceMessage MessageSeverity

IS Broker Transport Failure Status

Name	IS Broker Transport Failure Status	
Type	Event	
Description	IS Broker Transport status is failed.	
Expression	(ISBrokerTransport/Object Status by IS Broker Transport/Reading_Value) = (0)	
Related	Component	KPI
	IS Broker Transport	ObjectStatus

IS Cluster Critical Resource Message

Name	IS Cluster Critical Resource Message	
Type	KPI	
Description	IS Cluster Critical Resource Message received.	
Expression	(IsCluster/Contains Resource Message by IS Cluster/Reading_Value) = (1) AND (IsCluster/MessageSeverity by IS Cluster/Reading_Value) = (1)	

Related	Component	KPIs
	IS Cluster	ContainsResourceMessage MessageSeverity

IS Cluster Failure Status

Name	IS Cluster Failure Status	
Type	Event	
Description	IS Cluster status is failed.	
Expression	(ISCluster/Object Status by IS Cluster/Reading_Value) = (0)	
Related	Component	KPI
	IS Cluster	ObjectStatus

IS Email Port Critical Resource Message

Name	IS Email Port Critical Resource Message	
Type	KPI	
Description	IS Email Port Critical Resource Message received.	
Expression	(IsEmailPort/Contains Resource Message by IS Email Port/Reading_Value) = (1) AND (IsEmailPort/MessageSeverity by IS Email Port/Reading_Value) = (1)	
Related	Component	KPIs
	IS Email Port	ContainsResourceMessage MessageSeverity

IS Email Port Failure Status

Name	IS Email Port Failure Status	
Type	Event	
Description	IS Email Port status is failed.	
Expression	(ISEmailPort/Object Status by IS Email Port/Reading_Value) = (0)	
Related	Component	KPI
	IS Email Port	ObjectStatus

IS FTP Port Critical Resource Message

Name	IS FTP Port Critical Resource Message	
Type	KPI	
Description	IS FTP Port Critical Resource Message received.	
Expression	(IsFtpPort/Contains ResourceMessage by IS FTP Port/Reading_Value) = (1) AND (IsfTPPort/MessageSeverity by IS FTP Port/Reading_Value) = (1)	
Related	Component	KPIs
	IS FTP Port	ContainsResourceMessage MessageSeverity

IS FTP Port Failure Status

Name	IS FTP Port Failure Status	
Type	Event	

Description	IS FTP Port status is failed.	
Expression	(ISFTPPort/Object Status by IS FTP Port/Reading_Value) = (0)	
Related	Component	KPI
	IS FTP Port	ObjectStatus

IS HTTP Port Critical Resource Message

Name	IS HTTP Port Critical Resource Message	
Type	KPI	
Description	IS HTTP Port Critical Resource Message received.	
Expression	(IsHttpPort/Contains ResourceMessage by IS HTTP Port/Reading_Value) = (1) AND (IsHttpPort/MessageSeverity by IS HTTP Port/Reading_Value) = (1)	
Related	Component	KPIs
	IS HTTP Port	ContainsResourceMessage MessageSeverity

IS HTTP Port Failure Status

Name	IS HTTP Port Failure Status	
Type	Event	
Description	IS HTTP Port status is failed.	
Expression	(ISHTTTPort/Object Status by IS HTTP Port/Reading_Value) = (0)	
Related	Component	KPI
	IS HTTP Port	ObjectStatus

IS HTTPS Port Critical Resource Message

Name	IS HTTPS Port Critical Resource Message	
Type	KPI	
Description	IS HTTPS Port Critical Resource Message received.	
Expression	(IsHttpsPort/Contains Resource Message by IS HTTPS Port/Reading_Value) = (1) AND (IsHttpsPort/MessageSeverity by IS HTTPS Port/Reading_Value) = (1)	
Related	Component	KPIs
	IS HTTPS Port	ContainsResourceMessage MessageSeverity

IS HTTPS Port Failure Status

Name	IS HTTPS Port Failure Status	
Type	Event	
Description	IS HTTPS Port status is failed.	
Expression	(ISHTTPSPort/Object Status by IS HTTPS/Reading_Value) = (0)	
Related	Component	KPI
	IS HTTPS Port	ObjectStatus

IS Socket Port Critical Resource Message

Name	IS Socket Port Critical Resource Message	
Type	KPI	

Description	IS Socket Port Critical Resource Message received.	
Expression	(IsSocketPort/Contains Resource Message by IS Socket Port/Reading_Value) = (1) AND (IsSocketPort/MessageSeverity by IS Socket Port/Reading_Value) = (1)	
Related	Component	KPIs
	IS Socket Port	ContainsResourceMessage MessageSeverity

IS Socket Port Failure Status

Name	IS Socket Port Failure Status	
Type	Event	
Description	IS Socket Port status is failed.	
Expression	(ISocketPort/Object Status by IS Socket Port/Reading_Value) = (0)	
Related	Component	KPI
	IS Socket Port	ObjectStatus

Repository Server Critical Resource Message

Name	Repository Server Critical Resource Message	
Type	KPI	
Description	Repository Server Critical Resource Message received.	
Expression	(RepoV2/Contains ResourceMessage by Repository Server/Reading_Value) = (1) AND (RepoV2/MessageSeverity by Repository Server/Reading_Value) = (1)	
Related	Component	KPIs

Repository Server	ContainsResourceMessage MessageSeverity
-------------------	--

Repository Server Failure Status

Name	Repository Server Failure Status	
Type	Event	
Description	Repository Server status is failed.	
Expression	(RepoV2/Object Status by Repository Server/Reading_Value) = (0)	
Related	Component	KPI
	Repository Server	ObjectStatus

IS Package Rules

Integration Server Package Down

Name	Integration Server Package Down	
Type	Event	
Description	Integration Server package status is failed (while Integration Server status is not failed).	
Expression	(IntegrationServer/ObjectStatus by Integration Server/Reading_Value) = (1) AND (IsPackage/ObjectStatus by IS Package/Reading_Value) = (0)	
Related	Component	KPI
	IS Package	ObjectStatus
	Integration Server	ObjectStatus

Integration Server Package Failure Status

Name	Integration Server Package Failure Status	
Type	Event	
Description	Integration Server Package status is failed.	
Expression	y	
	Component	KPI
	IS Package	ObjectStatus

IS Package Critical Resource Message

Name	IS Package Critical Resource Message	
Type	KPI	
Description	IS Package Critical Resource Message received.	
Expression	(IsPackage/Contains ResourceMessage by IS Package/Reading_Value) = (1) AND (IsPackage/MessageSeverity by IS Package/Reading_Value) = (1)	
Related	Component	KPIs
	IS Package	ContainsResourceMessage MessageSeverity

IS Service Rules

Integration Server Service Failure Status

Name	Integration Server Failure Status
------	-----------------------------------

Type	Event				
Description	Integration Server status is failed.				
Expression	(IntegrationServer/ObjectStatus by Integration Server/ Reading_Value) = (0)				
Related	<table border="1"> <thead> <tr> <th>Component</th> <th>KPI</th> </tr> </thead> <tbody> <tr> <td>IS Service</td> <td>ObjectStatus</td> </tr> </tbody> </table>	Component	KPI	IS Service	ObjectStatus
Component	KPI				
IS Service	ObjectStatus				

Integration Server Service Request Time Trend

Name	Integration Server Service Request Time Trend				
Type	KPI				
Description	Integration Server service request time trending up.				
Expression	(IntegrationServer/CurrentAverageServiceRequestTime by Integration Server/Diagnosis_Trend_Up) = (1.0)				
Related	<table border="1"> <thead> <tr> <th>Component</th> <th>KPI</th> </tr> </thead> <tbody> <tr> <td>IS Service</td> <td>CurrentAverageServiceRequestTime</td> </tr> </tbody> </table>	Component	KPI	IS Service	CurrentAverageServiceRequestTime
Component	KPI				
IS Service	CurrentAverageServiceRequestTime				

IS Service Critical Resource Message

Name	IS Service Critical Resource Message		
Type	KPI		
Description	IS Service Critical Resource Message received.		
Expression	(ISService/Contains ResourceMessage by IS Service/ Reading_Value) = (1) AND (ISService/MessageSeverity by IS Service/Reading_Value) = (1)		
Related	<table border="1"> <thead> <tr> <th>Component</th> <th>KPI</th> </tr> </thead> <tbody> </tbody> </table>	Component	KPI
Component	KPI		

IS Service

ContainsResourceMessage
MessageSeverity

My webMethods Server Rules

DirServiceCache authenticateCacheSize High

Name	DirServiceCache authenticateCacheSize High	
Type	Event	
Description	DirServiceCache authenticateCacheSize is high.	
Expression	$\left(\frac{\text{DirServiceCache/authenticateCacheSize by DirServiceCache/Reading_Value}}{\text{DirServiceCache/authenticateCacheCapacity by DirServiceCache/Reading_Value}} \right) > (0.9)$	
Related	Component	KPI
	My webMethods	dnCacheSizeHigh CacheFailureStatus uriCacheSizeHigh

DirServiceCache dnCacheSizeHigh

Name	DirServiceCache dnCacheSize High	
Type	Event	
Description	Dir Service Cache dnCacheSize is high.	
Expression	$\left(\frac{\text{DirServiceCache/dnCacheSize by DirServiceCache/Reading_Value}}{\text{DirServiceCache/dnCacheCapacity by DirServiceCache/Reading_Value}} \right) > (0.9)$	
Related	Component	KPI
	My webMethods Server	authenticateCacheSizeHigh Failure Status

uriCacheSize High

DirServiceCache Failure Status

Name	DirServiceCache Failure Status	
Type	Event	
Description	DirServiceCache is failed.	
Expression	(MywebMethodsServer/Object Status by MywebMethodsServer/Reading_Value) = (1) AND (DirServiceCache/Object Status by DirServiceCache/Reading_Value) = (0)	
Related	Component	KPI
	My webMethods Server	authenticateCacheSizeHigh dnCacheSizeHigh uriCacheSize High

DirServiceCache uriCacheSize High

Name	DirServiceCache uriCacheSize High	
Type	Event	
Description	DirServiceCache uriCacheSize is high.	
Expression	((DirServiceCache/uriCacheSize by DirServiceCache/Reading_Value) / (DirServiceCache/uriCacheCapacity by DirServiceCache/Reading_Value)) > (0.9)	
Related	Component	KPI
	My webMethods Server	authenticateCacheSizeHigh dnCacheSizeHigh uriCacheSize High Failure Status

Handler.authManager Failure Status

Name	Handler.authManager Failure Status	
Type	Event	
Description	Handler.authManager status is failed.	
Expression	(MywebMethodsServer/Object Status by MywebMethodsServer/Reading_Value) = (1) AND (Handler.authManager/Object Status by Handler.authManager/Reading_Value) = (0)	
Related	Component	KPI
	My webMethods Server	Handler.dispatchManager Failure Status Handler.presManager Failure Status Handler.sessionManager Failure Status

Handler.dispatchManager Failure Status

Name	Handler.dispatchManager Failure Status	
Type	Event	
Description	Handler.dispatchManager status is failed.	
Expression	(MywebMethodsServer/Object Status by MywebMethodsServer/Reading_Value) = (1) AND (Handler.dispatchManager/Object Status by Handler.dispatchManager/Reading_Value) = (0)	
Related	Component	KPI
		Handler.authManager Failure Status Handler.presManager Failure Status Handler.sessionManager Failure Status

Handler.presManager Failure Status

Name	Handler.presManager Failure Status	
Type	Event	
Description	Handler.presManager status is failed.	
Expression	(MywebMethodsServer/Object Status by MywebMethodsServer/Reading_Value) = (1) AND (Handler.presManager/Object Status by Handler.presManager/Reading_Value) = (0)	
Related	Component	KPI
	My webMethods Server	Handler.authManager Failure Status Handler.dispatchManager Failure Status Handler.sessionManager Failure Status

Handler.sessionManager Failure Status

Name	Handler.sessionManager Failure Status	
Type	Event	
Description	Handler.sessionManager status is failed.	
Expression	(MywebMethodsServer/Object Status by MywebMethodsServer/Reading_Value) = (1) AND (Handler.sessionManager/Object Status by Handler.sessionManager/Reading_Value) = (0)	
Related	Component	KPI
	My webMethods Server	Handler.authManager Failure Status Handler.dispatchManager Failure Status Handler.presManager Failure Status

MWSCache cacheSize High

Name	MWSCache cacheSize High	
Type	Event	
Description	MWSCache cacheSize is high.	
Expression	((MWSCache/cacheSize by MWSCache/Reading_Value) / (MWSCache/maxSize by MWSCache/Reading_Value)) > (0.9)	
Related	Component	KPI
	My webMethods Server	Failure Status

MWSCache Failure Status

Name	MWSCache Failure Status	
Type	Event	
Description	MWSCache status is failed.	
Expression	(MywebMethodsServer/Object Status by MywebMethodsServer/Reading_Value) = (1) AND (MWSCache/Object Status by MWSCache/Reading_Value) = (0)	
Related	Component	KPI
	My webMethods Server	cacheSize High

MWSConnectionPool Closed

Name	MWSConnectionPool Closed	
Type	Event	

Description	MWSCollectionPool is closed.	
Expression	(MWSCollectionPool/closed by MWSCollectionPool/Reading_Value) = (1)	
Related	Component	KPI
		Failure Status poolSize High

MWSCollectionPool Failure Status

Name	MWSCollectionPool Failure Status	
Type	Event	
Description	MWSCollectionPool status is failed.	
Expression	(MywebMethodsServer/Object Status by MywebMethodsServer/Reading_Value) = (1) AND (MWSCollectionPool/Object Status by MWSCollectionPool/Reading_Value) = (0)	
Related	Component	KPI
	My webMethods Server	ConnectionPool Closed poolSize High

MWSCollectionPool poolSize High

Name	MWSCollectionPool poolSize High	
Type	Event	
Description	MWSCollectionPool poolSize is high.	
Expression	((MWSCollectionPool/poolSize by MWSCollectionPool/Reading_Value) / (MWSCollectionPool/maxLength by MWSCollectionPool/Reading_Value)) > (0.8)	

Related	Component	KPI
	My webMethods Server	ConnectionPool Closed poolSize High ConnectionPool Failure Status

MWSDatabase FailureStatus

Name	MWSDatabase Failure Status	
Type	Event	
Description	MWSDatabase status is failed.	
Expression	(MywebMethodsServer/Object Status by MywebMethodsServer/Reading_Value) = (1) AND (MWSDatabase/Object Status by MWSDatabase/Reading_Value) = (0)	
Related	Component	KPI

MWSDirectory FailureStatus

Name	MWSDirectory Failure Status	
Type	Event	
Description	MWSDirectory status is failed.	
Expression	(MywebMethodsServer/Object Status by MywebMethodsServer/Reading_Value) = (1) AND (MWSDirectory/Object Status by MWSDirectory/Reading_Value) = (0)	
Related	Component	KPI

MWSTaskEngine FailureStatus

Name	MWSTaskEngine Failure Status	
Type	Event	
Description	MWSTaskEngine status is failed.	
Expression	(MywebMethodsServer/Object Status by MywebMethodsServer/Reading_Value) = (1) AND (MWSTaskEngine/Object Status by MWSTaskEngine/Reading_Value) = (0)	
Related	Component	KPI

MWSTaskEvents FailureStatus

Name	MWSTaskEvents Failure Status	
Type	Event	
Description	MWSTaskEvents status is failed.	
Expression	(MywebMethodsServer/Object Status by MywebMethodsServer/Reading_Value) = (1) AND (MWSTaskEvents/Object Status by MWSTaskEvents/Reading_Value) = (0)	
Related	Component	KPI

MywebMethodsServer FailureStatus

Name	MywebMethodsServer Failure Status	
Type	Event	
Description	MywebMethods Server status is failed.	

Expression	(MywebMethodsServer/Object Status by MywebMethodsServer/Reading_Value) = (0)	
Related	Component	KPI
	My webMethods Server	HTTP_Request averageTime Trending Up Memory.freeMemory low

MywebMethodsServer HTTP_Request averageTime Trending Up

Name	MywebMethodsServer HTTPRequest averageTime Trending Up	
Type	Event	
Description	MywebMethods Server HTTP_Request averageTime is trending up.	
Expression	(MywebMethodsServer/Http_Request.averageTime by MywebMethodsServer/Reading_Trending_Up) = (1)	
Related	Component	KPI
	My webMethods Server	Failure Status Memory.freeMemory low

MywebMethodsServer Memory.freeMemory low

Name	MywebMethodsServer Memory.freeMemory low	
Type	Event	
Description	MywebMethods Server freeMemory is low.	
Expression	((MywebMethodsServer/Memory.freeMemory by MywebMethodsServer/Reading_Value) / (MywebMethodsServer/Memory.maxMemory by MywebMethodsServer/Reading_Value)) < (0.2)	
Related	Component	KPI

My webMethods
Server

HTTP_Request averageTime Trending Up
HTTP_request averageTime Trending Up

Portlet.Action Failure Status

Name	Portlet.Action Failure Status	
Type	Event	
Description	Portlet.Action status is failed.	
Expression	(MywebMethodsServer/Object Status by MywebMethodsServer/Reading_Value) = (1) AND (Portlet.Action/Object Status by Portlet.Action/Reading_Value) = (0)	
Related	Component	KPI

Portlet.PCALayout Failure Status

Name	Portlet.PCALayout Failure Status	
Type	Event	
Description	Portlet PCALayout status is failed.	
Expression	(MywebMethodsServer/Object Status by MywebMethodsServer/Reading_Value) = (1) AND (Portlet.PCALayout/Object Status by Portlet.PCALayout/Reading_Value) = (0)	
Related	Component	KPI

Portlet.Render Failure Status

Name	Portlet.Render Failure Status	
------	-------------------------------	--

Type	Event
Description	Portlet.Render status is failed.
Expression	(MywebMethodsServer/Object Status by MywebMethodsServer/Reading_Value) = (1) AND (Portlet.Render/Object Status by Portlet.Render/Reading_Value) = (0)
Related	Component KPI

Trading Networks Rules

TN Database Connection Pool Failure Status

Name	TN Database Connection Pool Failure Status
Type	Event
Description	TN Database Connection Pool status is failed.
Expression	(TNDBConnectionPool/ObjectStatus by TN DB Connection Pool/Reading_Value) = (0)
Related	Component KPI
	TN Database Connection Pool ObjectStatus

TN DB Connection Pool Critical Resource Message

Name	TN DB Connection Pool Critical Resource Message
Type	KPI
Description	TN DB Connection Pool Critical Resource Message received.
Expression	(TNDBConnectionPool/Contains Resource Message by TN DB Connection Pool/Reading_Value) = (1)

AND(TNDBConnectionPool/Message Severity by TN DB
Connection Pool/Reading_Value)= (1)

Related	Component	KPIs
	TN Database Connection Pool	ContainsResourceMessage MessageSeverity

TN Server Critical Resource Message

Name	TN Server Critical Resource Message	
Type	KPI	
Description	TN Server Critical Resource Message received.	
Expression	(TNServer/Contains Resource Message by TN Server/ Reading_Value) = (1) AND (TNServer/Message Severity by TN Server/Reading_Value) = (1)	
Related	Component	KPIs
	TN Server	ContainsResourceMessage MessageSeverity

TN Server Failure Status

Name	TN Server Failure Status	
Type	Event	
Description	TN Server status is failed.	
Expression	(TNServer/Object Status by TN Server/Reading_Value) = (0)	
Related	Component	KPI
	TN Server	ObjectStatus

TN Server Used Threads Too Big

Name	TN Server Used Threads Too Big	
Type	KPI	
Description	TN Server used threads is too big.	
Expression	$\left(\frac{\text{TNTaskManager/Used Threads by TN Task Manager/Reading_Value}}{\text{TNTaskManager/Max Threads by TN Task Manager/Reading_Value}} \right) \geq (0.9)$	
Related	Component	KPIs
	TN Task Manager	UsedThreads MaxThreads

TN Task Manager Critical Resource Message

Name	TN Task Manager Critical Resource Message	
Type	KPI	
Description	TN Task Manager Critical Resource Message received.	
Expression	$(\text{TNTaskManager/Contains Resource Message by TN Task Manager/Reading_Value}) = (1) \text{ AND } (\text{TNTaskManager/MessageSeverity by TN Task Manager/Reading_Value}) = (1)$	
Related	Component	KPIs
	TN Task Manager	ContainsResourceMessage MessageSeverity

TN Task Manager Failure Status

Name	TN Task Manager Failure Status
------	--------------------------------

Type	Event	
Description	TN Task Manager status is failed.	
Expression	(TNTaskManager/ObjectStatus by TN Task Manager/ Reading_Value) = (0)	
Related	Component	KPI
	TN Task Manager	ObjectStatus

Terracotta Rules

Terracotta Cache Failure Status

Name	Terracotta Cache Failure Status	
Type	Event	
Description	Terracotta Cache status is failed.	
Expression	(TerracottaCacheManager/Object Status by TerracottaCacheManager/Reading_Value) = (1) AND (TerracottaCache/Object Status by TerracottaCache/ Reading_Value) = (0)	
Related	Component	KPI
	Terracotta Cache	ObjectStatus

Terracotta Cache Manager Failure Status

Name	Terracotta Cache Manager Failure Status	
Type	Event	
Description	.Terracotta Cache Manager status is failed.	

Expression	(TerracottaConnection/Object Status by TerracottaConnection/Reading_Value) = (1) AND (TerracottaCacheManager/Object Status by TerracottaCacheManager/Reading_Value) = (0)	
Related	Component	KPI
	Terracotta Cache Manager	ObjectStatus

Terracotta Connection Failure Status

Name	Terracotta Connection Failure Status	
Type	Event	
Description	Terracotta Connection status is failed.	
Expression	(TerracottaConnection/Object Status by TerracottaConnection/Reading_Value) = (0)	
Related	Component	KPI
	Terracotta Connection	ObjectStatus

TSA Server Failure Status

Name	TSA Server Failure Status	
Type	Event	
Description	TSA Server status is failed.	
Expression	(TSA Server/Object Status by TSA/Reading_Value) = (1) AND (TSA Server/Object Status by TSA Server/Reading_Value) = (0)	
Related	Component	KPI
	TSA Server	ObjectStatus

TSA Failure Status

Name	TSA Failure Status	
Type	Event	
Description	TSA status is failed.	
Expression	(TerracottaConnection/Object Status by TerracottaConnection/Reading_Value) = (1) AND (TSA/Object Status by TSA/Reading_Value) = (0)	
Related	Component	KPI
	TSA	ObjectStatus

Presto Rules

Presto Failure Status

Name	PrestoFailure Status	
Type	Event	
Description	Presto status is failed.	
Expression	(Presto/Object Status by Presto/Reading_Value) = (0)	
Related	Component	KPI
	Presto	ObjectStatus

Universal Messaging Rules

Realm Channel Failure Status

Name	Realm Channel Failure Status	
Type	Event	
Description	Realm Channel status is failed.	
Expression	(Realm/Object Status by Realm/Reading_Value) = (1) AND (RealmChannel/Object Status by RealmChannel/Reading_Value) = (0)	
Related	Component	KPI
	Realm Channel	ObjectStatus

Realm Datagroup Failure Status

Name	Realm Datagroup Failure Status	
Type	Event	
Description	Realm Datagroup status is failed.	
Expression	(Realm/Object Status by Realm/Reading_Value) = (1) AND (RealmDatagroup/Object Status by RealmDatagroup/ Reading_Value) = (0)	
Related	Component	KPI
	Realm Datagroup	ObjectStatus

Realm Failure Status

Name	Realm Failure Status	
Type	Event	
Description	Realm status is failed.	
Expression	$(\text{Realm/Object Status by Realm/Reading_Value}) = (0)$	
Related	Component	KPI
	Realm	ObjectStatus

Realm Interface Failure Status

Name	Realm Interface Failure Status	
Type	Event	
Description	Realm Interface status is failed.	
Expression	$(\text{Realm/Object Status by Realm/Reading_Value}) = (1)$ AND $(\text{RealmInterface/Object Status by RealmInterface/Reading_Value}) = (0)$	
Related	Component	KPI
	Realm Interface	ObjectStatus

Realm Queue Failure Status

Name	Realm Queue Failure Status	
Type	Event	
Description	Realm Queue status is failed.	

Expression	(Realm/Object Status by Realm/Reading_Value) = (1) AND (RealmQueue/Object Status by RealmQueue/Reading_Value) = (0)	
Related	Component	KPI
	Realm Queue	ObjectStatus

Realm Threadpool Failure Status

Name	Realm Threadpool Failure Status	
Type	Event	
Description	Realm Threadpool status is failed.	
Expression	(Realm/Object Status by Realm/Reading_Value) = (1) AND (RealmThreadpool/Object Status by RealmThreadpool/ Reading_Value) = (0)	
Related	Component	KPI
	Realm Threadpool	ObjectStatus

C Built-In Services

■ Overview	502
■ Summary of Elements in This Folder	502
■ pub.optimize.monitoring:addEvent (event)	503
■ pub.optimize.monitoring:addEvents (events)	505
■ pub.optimize.monitoring:addDimension (dimension)	506
■ pub.optimize.monitoring:addDimensions (dimensions)	507

Overview

The WmOptimize package contains built-in Java services that enable you to send business information to Optimize, including process statistics, business details, and analytics. This package is available to all webMethods systems that use an Analytic Engine, and when activated on Integration Server and set up for your integration environment, it can send business data to the Analytic Engine.

The services in the WmOptimize package enable you to monitor four types of business data:

- **Dimension data** describes traits or characteristics of data. The package contains services to add single or multiple dimensions. This is non-computational data.
- **Event data** is actual information about business activities. The package contains services to add single or multiple events. This is computational data.
- **Process/step control data** is detailed information about business process start, stop, and step activities. An existing process model is required to send this information.
- **Process execution error data** is information about errors that occurred during execution of business processes.

The WmOptimize services are generally analogous to services for sending process, event, and dimension data available via the Web Service Data Collector using custom programming.

After you install and configure the WmOptimize package, you must connect the desired services to their existing integration environment. Typically this is done using Software AG Designer. For more information, refer to the *Web Services Developer's Guide*.

WmOptimize also contains services that enable you to configure multiple connections to additional Brokers so you can send process information to any of these Brokers.

The WmOptimize package also contains a service that enables you to retrieve data from Optimize Analytic Engine. The service returns an XML string that you can paste into ARIS MashZone to display the data.

Summary of Elements in This Folder

Service and Description

[pub.optimize.monitoring:addEvent \(event\)](#)[pub.optimize.monitoring:addEvents \(events\)](#)

Sends a single event ([addEvent](#)) or multiple events ([addEvents](#)) to the Analytic Engine.

[pub.optimize.monitoring:addDimension \(dimension\)](#)[pub.optimize.monitoring:addDimensions \(dimensions\)](#)

Service and Description

Sends a single dimension (`addDimension`) or multiple dimensions (`addDimensions`) to the Analytic Engine.

`pub.optimize.monitoring:pushExecutionError` (`ExecutionError`)

Pushes data about a process-related error to Optimize.

`pub.optimize.monitoring:pushProcessControlOperation` (`ProcessControlOperation`)

Pushes process state change events to Optimize.

`pub.optimize.monitoring:pushStepControlOperation` (`StepControlOperation`)

Pushes step state change events to Optimize.

`pub.optimize.monitoring:createNewConnection`

Creates connections to additional Brokers.

`pub.optimize.monitoring:destroyConnection`

Deletes a connection to a Broker.

`pub.optimize.mashzone.adapter:retrieveData`

Retrieves data from Optimize Analytic Engine and returns an XML string that you can paste into ARIS MashZone to display the data.

Note that while these entities are referred to as services, they are actually Java methods and could also be referred to as such. These services and their input parameters are explained throughout the rest of this chapter. Refer to the information on input parameters for specific services when connecting and invoking those services in an integration environment.

You can test any of the services within the `WmOptimize` package using Software AG Designer. For more details, see information about testing and debugging services in the *Web Services Developer's Guide*.

`pub.optimize.monitoring:addEvent` (`event`)

Sends a single event (`addEvent`) or multiple events (`addEvents`) to the Analytic Engine.

Input Parameters

Connection **String** Optional. Specifies the JMS server to which information will be sent. If no JMS server is specified, the service uses webMethods Universal Messaging as the default. The webMethods Universal Messaging URL format is as follows:

```
broker://{host}:{port}/{brokerName}/analysis
```

The Broker URL format is as follows:

```
broker://{host}:{port}/{brokerName}/analysis
```

Note: Before you can use this parameter, you must ensure that the specified connection is available on the Connection Settings page as described in ["Configuring the WmOptimize Package" on page 236](#). If the connection does not exist, you can create a connection to the broker URL using the ["pub.optimize.monitoring:createNewConnection" on page 518](#) service.

timestamp **String** Optional. Time to associate with the *eventType*. The time zone parameter of the date must be set to "GMT" in Designer. For input from Integration Server or Designer, the time must be in the following format:

```
yyyy-MM-ddTHH:mm:ss.SSSZ
```

For input from Optimize, the time must be in one of the following formats:

```
yyyy-MM-dd'T'HH:mm:ss.SSS'Z'
yyyy-MM-dd'T'HH:mm:ss.SSSZ
yyyy-MM-dd'T'HH:mm:ss.SSS
yyyy-MM-dd HH:mm:ss.SSS
```

If left blank, this field defaults to the current date and time.

eventType **String** Type of the event (for example, OrderProcess).

eventMap **Document** Container for the list of event attributes.

Parameter	Value
-----------	-------

<i>key</i>	String Attribute name (for example, Revenue).
------------	--

<i>value</i>	String or key/value Attribute value (for example, 500).
--------------	--

Note: If the parameter is of type *date*, you must use the following time format:

```
yyyy-MM-dd'T'HH:mm:ss.SSS'Z'
```

Output Parameters

result **String** “success” or “failure” for the service, and an error message in case of failure.

pub.optimize.monitoring:addEvents (events)

Sends multiple events (addEvents) to the Analytic Engine.

Input Parameters

Connection **String** Optional. Specifies the JMS server to which information will be sent. If no JMS server is specified, the service uses webMethods Universal Messaging as the default. The webMethods Universal Messaging URL format is as follows:

```
broker://{host}:{port}/{brokerName}/analysis
```

The Broker URL format is as follows:

```
broker://{host}:{port}/{brokerName}/analysis
```

Note: Before you can use this parameter, you must ensure that the specified connection is available on the Connection Settings page as described in ["Configuring the WmOptimize Package" on page 236](#). If the connection does not exist, you can create a connection to the broker URL using the ["pub.optimize.monitoring:createNewConnection" on page 518](#) service.

events Container for event data.

timestamp **String** Optional. Time to associate with the *eventType*. The time zone parameter of the date must be set to “GMT” in Designer. For input from Integration Server or Designer, the time must be in the following format:

```
yyyy-MM-ddTHH:mm:ss.SSSZ
```

For input from Optimize, the time must be in one of the following formats:

```
yyyy-MM-dd'T'HH:mm:ss.SSS'Z'
```

```
yyyy-MM-dd'T'HH:mm:ss.SSSZ
```

```
YYYY-MM-dd'T'HH:mm:ss.SSS
```

```
yyyy-MM-dd HH:mm:ss.SSS
```

If left blank, this field defaults to the current date and time.

eventType **String** Type of the event (for example, OrderProcess).

eventMap **Document** Container for the list of event attributes.

Parameter	Value
<i>key</i>	String Attribute name (for example, Revenue).
<i>value</i>	String or key/value Attribute value (for example, 500).

Note: If the parameter is of type `date`, you must use the following time format:

```
yyyy-MM-dd'T'HH:mm:ss.SSS'Z'
```

Output Parameters

result **String** "success" or "failure" for the service, and an error message in case of failure.

pub.optimize.monitoring:addDimension (dimension)

Sends a single dimension (addDimension) to the Analytic Engine.

Input Parameters

Connection **String** Optional. Specifies the JMS server to which information will be sent. If no JMS server is specified, the service uses webMethods Universal Messaging as the default. The webMethods Universal Messaging URL format is as follows:

```
broker://{host}:{port}/{brokerName}/analysis
```

The Broker URL format is as follows:

```
broker://{host}:{port}/{brokerName}/analysis
```

Note: Before you can use this parameter, you must ensure that the specified connection is available on the Connection Settings page as described in "[Configuring the WmOptimize Package](#)" on page 236. If the connection does

not exist, you can create a connection to the broker URL using the "[pub.optimize.monitoring:createNewConnection](#)" on page 518 service.

dimensionName **String** Name of the dimension (for example, *Customer*).

dimensionMap **Document** Container for the list of dimension attributes.

Parameter	Value
<i>key</i>	String Attribute name (for example, <i>Revenue</i>).
<i>value</i>	String or key/value attribute value (for example, <i>500</i>).

Output Parameters

result **String** "success" or "failure" for the service, and an error message in case of failure.

pub.optimize.monitoring:addDimensions (dimensions)

Sends multiple dimensions (addDimensions) to the Analytic Engine.

Input Parameters

connection **String** Optional. Specifies the JMS server to which information will be sent. If no JMS server is specified, the service uses webMethods Universal Messaging as the default. The webMethods Universal Messaging URL format is as follows:

```
broker://{host}:{port}/{brokerName}/analysis
```

The Broker URL format is as follows:

```
broker://{host}:{port}/{brokerName}/analysis
```

Note: Before you can use this parameter, you must ensure that the specified connection is available on the Connection Settings page as described in "[Configuring the WmOptimize Package](#)" on page 236. If the connection does not exist, you can create a connection to the broker URL using the "[pub.optimize.monitoring:createNewConnection](#)" on page 518 service.

dimensions **Document** Container for the list of dimension attributes.

dimensionName **String** Name of the dimension (for example, Customer).

dimensionMap **Document** Container for dimension attribute information.

<u>Parameter</u>	<u>Value</u>
<i>key</i>	String Attribute name (for example, Revenue).
<i>value</i>	String or key/value Attribute value (for example, 500).

Output Parameters

result **String** “success” or “failure” for the service, and an error message in case of failure.

pub.optimize.monitoring:pushExecutionError (ExecutionError)

Pushes data about a process-related error to Optimize.

The input parameter is an instance of ExecutionError.

Input Parameters

connection **String** Optional. Specifies the JMS server to which information will be sent. If no JMS server is specified, the service uses webMethods Universal Messaging as the default. The webMethods Universal Messaging URL format is as follows:

```
broker://{host}:{port}/{brokerName}/analysis
```

The Broker URL format is as follows:

```
broker://{host}:{port}/{brokerName}/analysis
```

Note: Before you can use this parameter, you must ensure that the specified connection is available on the Connection

Settings page as described in ["Configuring the WmOptimize Package" on page 236](#). If the connection does not exist, you can create a connection to the broker URL using the ["pub.optimize.monitoring:createNewConnection" on page 518](#) service.

timestamp

String Optional. The time when the event occurred. The time zone parameter of the date must be set to "GMT" in Designer. For input from Integration Server or Designer, the time must be in the following format: *yyyy-MM-ddTHH:mm:ss.SSSZ*.

For input from Optimize, the time must be in one of the following formats:

```
yyyy-MM-dd' T' HH:mm:ss.SSS' Z'
yyyy-MM-dd' T' HH:mm:ss.SSSZ
yyyy-MM-dd' T' HH:mm:ss.SSS
yyyy-MM-dd HH:mm:ss.SSS
```

stepName

String Optional. Name of the process step. The process step name must match the process step name specified in the model definition for this process. For information about configuring process models, see ["Configuring Business Processes" on page 99](#)

Note: If the Event Enablement flag not set in the wmOptimize configuration, the step name parameter emits a process instance error. If the Event Enablement flag is set, then a process step instance error event is emitted.

processInstanceId

Document Unique, two-part identifier for the process step.

keyValue

The complete set of keys you supply constitutes the unique name Optimize uses internally for the process.

<u>Parameter</u>	<u>Value</u>
<i>name</i>	Document List Set <i>name</i> to the first part of the identifier (for example, ordernumber).
<i>value</i>	Set <i>value</i> to the second part of the identifier (for example, 5). Optimize displays the identifier on the Process Instances page in two separate columns so you can sort by name or by value.

Note: ProcessInstanceID and event data share the same key value pairs.

<i>processName</i>	String Name of the process. The process name must be unique if it is to be used to push operations to Optimize, but this practice is supported only for backward compatibility. Use <i>processExternalId</i> instead, and mark this parameter “optional” in Designer.
<i>processInstanceDisplayName</i>	String Optional. A custom name for the process instance.
<i>type</i>	String The type of error that occurred.
<i>message</i>	String A short description of the error.
<i>messageDetail</i>	String Optional. A long description of the error. If a stack trace is available, store it in this field.
<i>processIteration</i>	Int Optional. Current iteration of the business process execution.

Output Parameters

None.

pub.optimize.monitoring:pushProcessControlOperation (ProcessControlOperation)

Pushes process state change events to Optimize.

The input parameter is an instance of ProcessControlOperation.

Input Parameters

<i>Connection</i>	String Optional. Specifies the JMS server to which information will be sent. If no JMS server is specified, the service uses webMethods Universal Messaging as the default. The webMethods Universal Messaging URL format is as follows: <code>broker://{host}:{port}/{brokerName}/analysis</code>
-------------------	--

The Broker URL format is as follows:

```
broker://{host}:{port}/{brokerName}/analysis
```

Note: Before you can use this parameter, you must ensure that the specified connection is available on the Connection Settings page as described in ["Configuring the WmOptimize Package" on page 236](#). If the connection does not exist, you can create a connection to the broker URL using the ["pub.optimize.monitoring:createNewConnection" on page 518](#) service.

timestamp

String Optional. The time when the event occurred. The time zone parameter of the date must be set to "GMT" in Designer. The time must be in the following format:

```
yyyy-MM-ddTHH:mm:ss.SSSZ
```

For input from Optimize, the time must be in one of the following formats:

```
yyyy-MM-dd'T'HH:mm:ss.SSS'Z'
YYYY-MM-dd'T'HH:mm:ss.SSSZ
yyyy-MM-dd'T'HH:mm:ss.SSS
yyyy-MM-dd HH:mm:ss.SSS
```

operation

String Type of operation.

<u>Set to...</u>	<u>To indicate that a business process has...</u>
start	Starts step. Business data can be passed with this operation.
stop	Stops step. Business data can be passed with this operation.
suspend	Paused. It is assumed that the process will resume execution at a later time. Time elapsed between suspend and resume operations does not contribute to the overall cycle time for the process. The elapsed time will not contribute to a process time-out action.
resume	Resumed execution. The time-out clock will reset.
cancel	Terminated abnormally.

Note: Process instances in the cancel or suspend state do not contribute to cycle time metrics. If a suspended process resumes and completes successfully, it will not contribute the time spent suspended to the cycle time.

parentProcessInstanceId **Document List** Optional. A unique, two-part identifier for the parent process instance to which the executing process instance belongs.

keyValue Optimize displays the identifier on the Process Instances page in two separate columns so you can sort by name or by value. The complete set of keys you supply constitutes the unique name Optimize uses internally for the process.

<u>Parameter</u>	<u>Value</u>
<i>name</i>	Document List Set <i>name</i> to the label for the data (for example, <code>customer number</code>).
<i>value</i>	Set <i>value</i> to the value of the data (for example, <code>138</code>).

processInstanceId **Document List** Unique, two-part identifier for the process instance.

keyValue **Document List**
Optimize displays the identifier on the Process Instances page in two separate columns so you can sort by name or by value. The complete set of keys you supply constitutes the unique name Optimize uses internally for the process.

<u>Parameter</u>	<u>Value</u>
<i>name</i>	Set <i>name</i> to the first part of the identifier (for example, <code>ordernumber</code>).
<i>value</i>	Set <i>value</i> to the second part of the identifier (for example, <code>5</code>).

Note: ProcessInstanceID and event data share the same key value pairs.

<i>processName</i>	String Name of the process. The process name must be unique if it is used to push operations to Optimize, but this practice is supported only for backward compatibility. Use <i>processExternalId</i> instead, and mark this parameter “optional” in Designer.
<i>parentProcessName</i>	String Optional. (Required when <i>ParentProcessInstanceId</i> is specified.) The name of the parent process to which the executing process belongs.
<i>processExternalId</i>	String Optional. Optimize identifier of the business process. Use this parameter instead of <i>processName</i> if the application can access the process metadata. In Designer, this is referred to as ProcessID.
<i>processIteration</i>	Int Optional. Current iteration of the business process execution.
<i>businessData</i>	Document List Optional. Business data about the process instance.

<u>Parameter</u>	<u>Value</u>
<i>time</i>	<p>String Optional. Timestamp passed with event into Optimize for analysis. The time zone parameter of the date must be set to “GMT” in Designer. For input from Integration Server or Designer, the time must be in the following format:</p> <pre>yyyy-MM-ddTHH:mm:ss.SSSZ</pre> <p>For input from Optimize, the time must be in one of the following formats:</p> <pre>yyyy-MM-dd' T' HH:mm:ss.SSS' Z'</pre> <pre>yyyy-MM-dd' T' HH:mm:ss.SSSZ</pre> <pre>yyyy-MM-dd' T' HH:mm:ss.SSS</pre> <pre>yyyy-MM-dd HH:mm:ss.SSS</pre>
<i>event type</i>	<p>String Unique identifier for event data type in Optimize.</p>
<i>Attributes</i>	<p>Document List Optional. Business data about the process instance.</p>

Parameter Value

<i>name</i>	Set <i>name</i> to the label for the data (for example, customer number).
<i>value</i>	Set <i>value</i> to the value of the data (for example, 138).

Output Parameters

result **String** “success” or “failure” for the service, and an error message in case of failure.

pub.optimize.monitoring:pushStepControlOperation (StepControlOperation)

Pushes step state change events to Optimize.

The input parameter is an instance of StepControlOperation.

Input Parameters

Connection

String Optional. Specifies the JMS server to which information will be sent. If no JMS server is specified, the service uses webMethods Universal Messaging as the default. The webMethods Universal Messaging URL format is as follows:

```
broker://{host}:{port}/{brokerName}/analysis
```

The Broker URL format is as follows:

```
broker://{host}:{port}/{brokerName}/analysis
```

Note: Before you can use this parameter, you must ensure that the specified connection is available on the Connection Settings page as described in ["Configuring the WmOptimize Package" on page 236](#). If the connection does not exist, you can create a connection to the broker URL using the

"pub.optimize.monitoring:createNewConnection" on page 518 service.

timestamp

String Optional. The time when the event occurred. The time zone parameter of the date must be set to "GMT" in Designer. For input from Integration Server or Designer, the time must be in the following format:

yyyy-MM-ddTHH:mm:ss.SSSZ

For input from Optimize, the time must be in one of the following formats:

yyyy-MM-dd'T'HH:mm:ss.SSS'Z'

yyyy-MM-dd'T'HH:mm:ss.SSSZ

yyyy-MM-dd'T'HH:mm:ss.SSS

yyyy-MM-dd HH:mm:ss.SSS

operation

String (Required) Type of operation.

<u>Set to...</u>	<u>To indicate that a business process has...</u>
start	Starts step. Business data can be passed with this operation.
stop	Stops step. Business data can be passed with this operation.
log_business_data	Sends business data. Does not affect execution or state of the step.

stepName

String Name of the step. The name of the step must match the step name specified in the process model. If two processes share the same name, neither can be tracked. Use the *stepExternalId* parameter instead, and mark this parameter "optional" in Designer.

stepExternalId

String Optional. Optimize identifier of the process step. Use this parameter instead of *stepName* if the application can access the process metadata.

sourceStepExternalID

String Optional. Optimize identifier of the process source step. Use this parameter instead of *sourceStepName* if the application can access the process metadata.

sourceStepName

String Optional. Use this parameter to provide process tracker with info about transitional path between source and target steps. It is usefull if target step has more than

1 inbound transition from multiple source steps and user wants to track transitions between source and target step.

stepIteration

IntOptional. Current iteration of the business process execution.

parentProcessInstanceId

Document List Optional. A unique, two-part identifier for the parent process instance to which the executing process instance belongs.

keyValue

Record List

Parameter	Value
<i>name</i>	Set <i>name</i> to the label for the data (for example, <i>customer number</i>) and set <i>value</i> to the value of the data (for example, 138).
<i>value</i>	Set <i>value</i> to the value of the data (for example, 138).

processInstanceId

Document List Unique, two-part identifier for the process instance.

keyValue

Document List

Parameter	Value
<i>name</i>	Set <i>name</i> to the first part of the identifier (for example, <i>ordernumber</i>).
<i>Value</i>	<p>Optimize displays the identifier on the Process Instances page in two separate columns so you can sort by name or by value. The complete set of keys you supply constitutes the unique name Optimize uses internally for the process.</p> <p>Set <i>value</i> to the second part of the identifier (for example, 5).</p>

Note ProcessInstanceID and event data share the same key value pairs.

<i>processName</i>	String Name of the process. The process name must be unique if it is to be used to push operations to Optimize, but this practice is supported only for backward compatibility. Use <i>processExternalId</i> instead, and mark this parameter “optional” in Designer.
<i>parentProcess Name</i>	String Optional. (Required when <i>ParentProcessInstanceId</i> is enabled.) The name of the parent process to which the executing process belongs.
<i>processExternalId</i>	String Optional. Optimize identifier of the business process. Use this parameter instead of <i>processName</i> if the application can access the process metadata. In Designer, this is referred to as ProcessID.
<i>processIteration</i>	Int Optional. Current iteration of the business process execution.
<i>globalBusinessData</i>	String (True or False) Indicates whether the data being passed in by the server refers to all future steps or only to specified steps in the process.
<i>businessData</i>	Document List Optional. Business data about the process step.
<i>time</i>	<p>String Optional. Timestamp passed with event into Optimize for analysis. The time zone parameter of the date must be set to “GMT” in Designer. For input from Integration Server or Designer, the time must be in the following format:</p> <p>yyyy-MM-ddTHH:mm:ss.SSSZ</p> <p>For input from Optimize, the time must be in one of the following formats:</p> <p>yyyy-MM-dd' T' HH:mm:ss.SSS' Z' yyyy-MM-dd' T' HH:mm:ss.SSSZ yyyy-MM-dd' T' HH:mm:ss.SSS yyyy-MM-dd HH:mm:ss.SSS</p>
<i>eventType</i>	String Unique identifier for event data type in Optimize.

<i>attributes</i>	Document List
	<u>Parameter</u> <u>Value</u>
<i>name</i>	Set <i>name</i> to the label for the data (for example, customer number) and
<i>value</i>	Set <i>value</i> to the value of the data (for example, 138).

Output Parameters

None.

pub.optimize.monitoring:createNewConnection

Creates connections to additional Brokers.

You can define as many connections as desired and send information to any of the associated JMS servers using the *Connection* parameter in the other WmOptimize services.

Input Parameters

<i>brokerURL</i>	<p>String Defines an additional JMS server to which information will be sent.</p> <p>The webMethods Universal Messaging URL format is as follows: <code>broker://{host}:{port}/{brokerName}/analysis</code></p> <p>The Broker URL format is as follows: <code>broker://{host}:{port}/{brokerName}/analysis</code></p>
<i>useSSL</i>	<p>String Indicates that the connection to the JMS server requires a secure connection.</p>
<i>useEncryption</i>	<p>String Indicates that the connection to the JMS server requires encryption.</p>

<i>KeyStoreFile</i>	String Type the absolute path to the private key file that is located on a server accessible by the component. This field is required only if you enable SSL.
<i>KeyStoreType</i>	String Type the appropriate string to specify the Key Store Type (PKCS12 or JKS). This field is required only if you enable SSL.
<i>distinguishName</i>	String Type the name of the appropriate certificate identifier. This field is required only if you enable SSL.
<i>trustStoreFile</i>	String Type the absolute path to the CA certificate file on a server location accessible by Analytic Engine. This field is required only if you enable SSL.
<i>trustStoreType</i>	String Click the drop down to select the Trust Store Type (PKCS12 or JKS). This field is required only if you enable SSL.
<i>password</i>	String Type the pass phrase to access the specified Key and Trust Stores. This field is required only if you enable SSL.

Output Parameters

<i>result</i>	String “success” or “failure” for the service, and an error message in case of failure.
<i>message</i>	String Optional. Displays an error message in case of failure.

pub.optimize.monitoring:destroyConnection

Severs a connection to a Broker.

Input Parameters

<i>brokerURL</i>	String Defines a Broker to which information will no longer be sent. The specific format is <code>broker://{host}:{port}/{brokerName}/analysis</code> .
------------------	--

Output Parameters

- result* **String** “success” or “failure” for the service, and an error message in case of failure.
- message* **String** Optional. Displays an error message in case of failure.

pub.optimize.mashzone.adapter.retrieveData

Retrieves data from Optimize Analytic Engine and returns an XML string that you can paste into ARIS MashZone to display the data.

Note: The data is returned in GMT time format.

Input Parameters

- dataType* **String** Specifies the type of data you are looking for in Optimize.

<u>Set to...</u>	<u>To look for...</u>
KPI_INSTANCE_READING	KPI instance readings data in Optimize.
PROBLEM	Problems data in Optimize.
KPI_DEFINITION	KPI definitions data in Optimize.
DIMENSION_INSTANCE	Dimension instances data in Optimize.

- keys* **String List** Optional. Specifies a list of strings that act as the keys of the query parameters. The length of the *keys* string list must be equal to the length of the *values* string list, so the two can form valid pairs.

Note: The *keys* parameter is required only for KPI instance readings and dimension instances data. You can omit it when retrieving problems or KPI definitions data.

- values* **String List** Optional. Specifies a list of strings that act as the values of the query parameters. The length of the *keys* string list must be equal to the length of the *values* string list, so the two can form valid pairs.

Note: The *values* parameter is required only for KPI instance readings and dimension instances data. You can omit it when retrieving problems or KPI definitions data.

maxRows **String** Optional. Specifies the maximum number of data rows that you want the service to return. If you do not specify a value or specify a value less than 0, the service will use the default value, which is 1000. If you specify a value equal to 0, the service returns only the table header.

Output Parameters

xmlString **String** The XML string you can paste into the ARIS MashZone user interface to display the data.

D Optimize Schema Overview

■ Understanding the Optimize Schema	524
---	-----

Understanding the Optimize Schema

Optimize stores system and customer data in the “analysis” database within the applicable customer database product (Oracle, SQL Server, etc.). This data forms the schema that defines the structure of a particular database. As delivered, Optimize has a core schema that provides a basis for adding customer-specific metadata and runtime data. As a user creates event maps, facts, and dimensions to work with their specific business operations, Optimize creates the appropriate database tables automatically based on the implemented event maps. These tables form the customer-specific schema that is unique to each Optimize installation.

Because Optimize table names are derived from user-created data structures, it is impossible to provide an accurate, detailed schema diagram for any given user. However, Optimize follows a consistent taxonomy for naming and relating tables, which can be represented by a star diagram. Below is a graphical representation of a basic Optimize star schema:

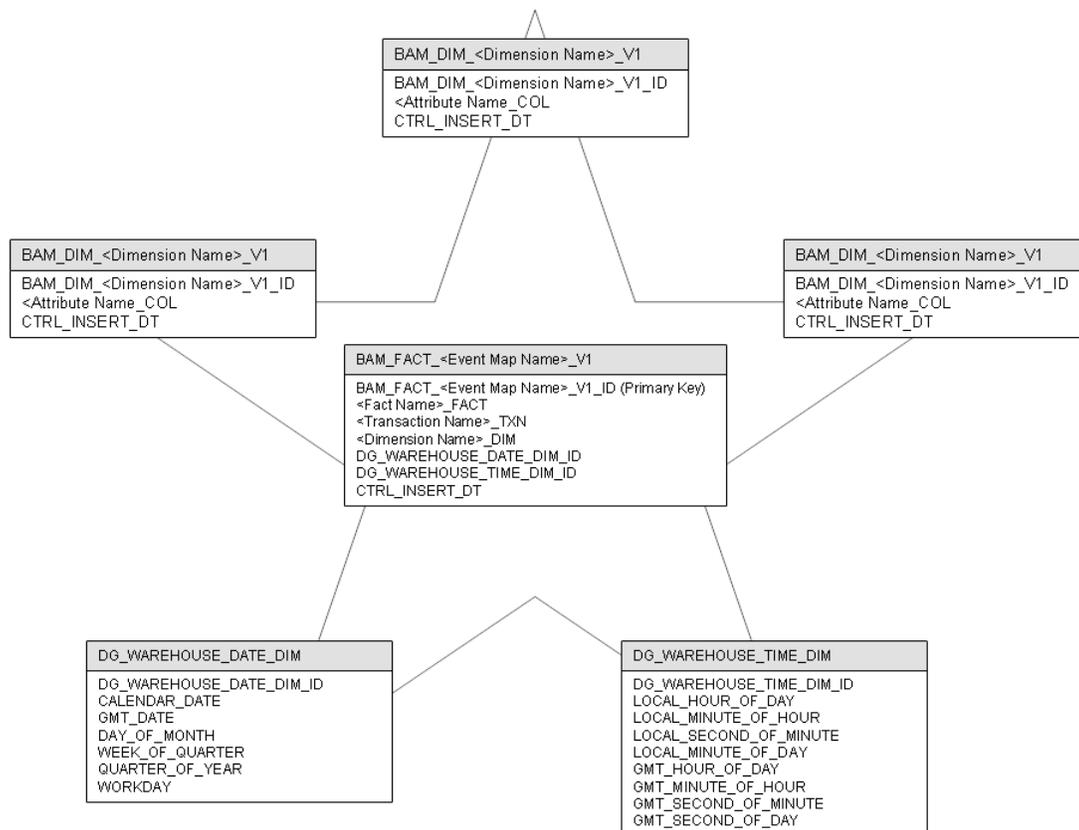


Table Naming and Versioning

Tables and table names in Optimize are derived from the objects they represent, specifically dimensions, facts, and event maps. When you create a dimension, fact, or event map, a new database is created to reflect this new object. There are two types of tables in an Optimize schema: Fact tables and Dimension tables. Note that event maps are metadata, and the tables related to them are beyond the scope of this document.

For example a dimension named Partner would have a corresponding table name of BAM_DIM_PARTNE_V1. If additional Partner dimensions are created, the "V" value is incremented to avoid naming conflicts. To continue our example, if another dimension named PartnershipType were created, the corresponding table would be BAM_DIM_PARTNE_V11. The order of "V" values is V1, V11, V12, V13, V14, and so on.

Fact Tables (BAM_FACT_<Event Map Name>_V1)

Fact tables capture information for every event (that has an event map) generated for Optimize. Each Fact table is associated with one event map and is created after the event map is saved through the Optimize interface in My webMethods. The name of the table is derived from the event map name (not the internal name). All Fact tables contain the following columns:

Column and Description

BAM_FACT <Event Map Name>_V1_ID

Primary key that uniquely identifies an event row.

<Fact Name>_FACT

Fact columns contain measurable values from events. Usually, though not always, this is a number. There can be any number of fact columns. Examples include order amount, transaction count, and failure percentage.

<Transaction Name>_TXN

Transaction columns contain textual information about an event that is not measurable and not a dimension. Examples include order number, transaction date, and phone number. Optimize does not use this information; it is collected solely for potential use by a reporting tool.

<Dimension Name>_DIM

Dimension columns contain integer values that are foreign keys to dimension tables. Associated dimension tables contain more information related to these dimension foreign keys.

Column and Description

DG_WAREHOUSE_DATE_DIM_ID

This column contains information about the date an event occurred, rather than when it was inserted into the database. The column value is a foreign key reference to the DG_WAREHOUSE_DATE_DIM table, which contains different representations of the date value.

DG_WAREHOUSE_TIME_DIM_ID

This column contains information about the time an event occurred, rather than when it was inserted into the database. The column value is a foreign key reference to the DG_WAREHOUSE_TIME_DIM table, which contains different representations of the time value.

CTRL_INSERT_DT

This is the date and time when an event was inserted into the database. If events from several days ago were delayed and entered today, the CTRL_INSERT_DT value would be today, but the DATE_DIM_ID and TIME_DIM_ID (see above), would be from several days ago. Optimize does not use this field.

Dimension Tables (BAM_DIM_<Dimension Name>_V1)

Dimension tables are linked to Fact tables through a foreign key in the Fact table. The specific column that connects the Fact table to the dimension is named after the Dimension table name. For example, a dimension called Region would have a Dimension table called BAM_DIM_REGION_V1. Any Fact table that uses this dimension would have a column named REGION_DIM that contained a value relating back to the Dimension table.

E Glossary

alert

1. A notification triggered by a rule violation. Rule violation alerts can take the form of e-mail messages, or you can specify SNMP alert actions or Web service alert actions.
2. In an Oracle database environment, a notification triggered when the Oracle purge operation completes or encounters an error.

analysis engine

An Analytic Engine component that runs algorithms on business, system, and process data.

Analytic Engine

An Optimize core function that handles the following tasks:

- Receives business, system, and process data from Infrastructure Data Collector.
- Analyzes the data.
- Saves the data and the analysis in the database.
- Sends information about the status of business and system activity to the My webMethods user interface.
- Performs other actions when problems occur, such as sending alerts to specified users.

API

Application programming interface.

application

A webMethods program such as Integration Server running on a system component in your environment.

asset

A container object with connection data. Previously called an object.

BAM

Business Activity Monitoring. BAM allows you to analyze real-time information about the performance of your business, including the volume of business activity and its responsiveness, serious errors that may have occurred, and other KPIs. Using this actionable data, you can eliminate problems and take advantage of business opportunities.

BAM-only process

See externally-executed process.

base rule

A rule. The term *base rule* is used to distinguish a rule from a rule instance.

BPEL

Business Process Execution Language.

BPM

Business Process Management.

BPM-only process

See webMethods-executed process.

Broker

1. An entity that resides on a Broker Server. When a client connects to a Broker Server, the client specifies the Broker with which it wants to interact.
2. See webMethods Broker.

Broker Server

The run-time component of webMethods Broker with which publishers and subscribers interact. It performs the communication-related work of receiving client requests, dispatching requests to the requested resource, and returning responses to clients.

business data

Data that make up a business process, such as revenue and order number.

business KPI

A KPI created by a user to monitor business process data. For example, users can create business KPIs to monitor Revenue by Customer. A business KPI differs from an intrinsic KPI, which is created by Optimize.

business process

A series of interrelated business tasks that are performed in a specific order, using an associated set of business rules; and by multiple systems, people, and partners. Examples of business processes within a corporation might include the following procedures:

- Preparing for a new employee.
- Handling a purchase order.
- Bringing a product from inception to market.
- Delivering a timely and accurate invoice.
- Enabling Vendor Managed Inventory.

You can monitor two types of processes from My webMethods: a webMethods-executed process or an externally-executed process.

business process management

The ability to define, implement, manage, analyze, and optimize business processes, which includes interactions between systems within your own corporation (e.g.,

applications and information stores), people in your corporation, and external business partners.

business process tree

A listing of business process components with branches and nodes that are displayed on the Analytics Overview page in Optimize.

Business Visualization

A set of Optimize analysis functions that present business data as images. Business Visualization provides users with actionable data by highlighting patterns and trends in business processes and key performance indicators (KPIs).

calculation type

One of the following five methods for data aggregation: sum, average, count, last value, and state.

CDC

Common Database Configuration. Platform containing the webMethods database components.

CCS

Central Configuration System. See Central Configuration.

Central Configuration

A webMethods tool that provides a graphical user interface for configuration, enabling you to configure your installation in a central location without manually editing configuration files.

composite KPI

A KPI that performs a mathematical operation (add, subtract, multiply, or divide) on two other KPIs. For example, if individual KPIs monitor a product's operating costs and revenue, a composite KPI can monitor the product's profit by subtracting the operating cost KPI from the revenue KPI.

container

A set of steps that comprise a subprocess within a business process. Subprocess containers can be nested one within another.

control flow

A transition that moves data directly from one step to another through the execution of an automated business process.

custom error

An error that originates in an external system. Examples of custom errors include application failure and database failure when the application or database is external to the system. A custom error can occur at either the process or step level.

data aggregation

The process of combining individual data values into summarized data values. To measure the revenue in dollars generated by the order management process, a Total Order Revenue KPI might sum the dollar amounts of orders received during a ten-minute data collection interval.

data collection interval

A time interval during which Optimize collects data for a KPI.

database component

A grouping of database objects deployed as a single or set of database SQL scripts. It is the lowest-level grouping of a set of scripts. A database component is defined by a database component definition file.

database component definition

An XML file that defines all the information related to a specific database component. Each database component has one and only one XML file.

database connection pool

A collection of pre-established connections to a database. A database connection pool avoids the overhead involved in making a new database connection every time an application or server requires access to the database.

DCA

Data collection agent.

DCC

Database Component Configurator. A Java program used to install database components found in the Common Database Configuration (CDC) platform by executing SQL commands over a JDBC connection. This includes creation, removal, and migration of database components. This program replaces the Database Component Installer (DCI).

DCI

Database Component Installer. This program has been replaced by the DCC (Database Component Configurator).

defect

A failure to meet a business rule (that is, a rule violation).

derived name

A user-defined name applied to all instances of a rule. If a derived name is specified in the Rule Information settings for the rule, all instance names will be derived from this name. Use variables in the derived name definition to keep all instances of the rule from having the same name.

destination step

A destination step relates to wait-time standardized error types. For a complete description, see wait-time standardized error type.

dimension

A category, or a way to analyze business data or system data by slicing the data into smaller components. Organizations define their own dimensions. Typical examples of dimensions are customer, region and sales person, product and manufacturer, and Broker Server and Broker.

dimension hierarchy (formerly KPI hierarchy)

See hierarchy.

discovery

A collection of instructions used by the Infrastructure Data Collector to discover assets.

display interval

The displayed aggregation interval used on KPI Instance graphs. Data is aggregated to the display interval to accommodate larger time frames on the graph.

environment

A grouping of webMethods product components that share common configuration settings. A default Optimize environment contains the following logical servers:

- An Analytic Engine
- A Web Service Data Collector
- A webMethods Broker
- A My webMethods Server

You can use Central Configuration to set up and configure an environment.

event

1. A simple structure containing a set of data attributes.
2. In Business Visualization, a rule violation.

event handler

An Analytic Engine component that receives business data and system data from the Web Service Data Collectors and other webMethods products. The Analytic Engine writes this data to the database in a format that business intelligence software can use to generate reports.

event map

The knowledge of what each attribute in an event (definition 1) means. An event map associates business data, such as dimensions, with a particular business process.

event rule

A rule that evaluates data as it is collected rather than waiting until the end of a KPI data collection interval. An event rule takes dimensions into account, tracks its state of compliance, and persists in the database. In general, event rules evaluate data in situations that require immediate response. For example, an event rule might track whether an application has unexpectedly shut down.

externally-executed process

A business process that is modeled and monitored using webMethods-provided tools; however, the process is executed by an external application. Compare to webMethods-executed process.

global data

No longer applicable. Global data was deprecated in Optimize version 7.1.2.

hierarchy

In Optimize, an ordered ranking of dimensions. A hierarchy provides additional ways to slice data into smaller components. For example, a sales hierarchy might consist of

the two dimensions of region and salesperson. As another example, a queue-length hierarchy might consist of the two dimensions of Broker Server and Broker.

infrastructure

A set of system components (Integration Server, Broker, SNMP agents).

Infrastructure Data Collector

A BAM component that monitors the system and operational data associated with webMethods run-time components, such as Integration Servers, Broker Servers, Brokers, and adapters, and reports the status of these components on Optimize for Infrastructure or other external tools.

Integration Server

See webMethods Integration Server.

intrinsic KPI

A KPI created by Optimize to monitor business process or system data. For example, Optimize creates KPIs to monitor Instance Count by Process, Cycle Time by Process, and Process Error Count.

KPI

Key performance indicator. A measurement of a business activity that is important to the success of an organization. KPIs monitor metrics, quantitative business and system data, such as revenue, volume of orders, queue length, and cycle time. KPIs help answer questions such as, "How many orders over \$10,000 are stuck in this process?" A KPI defines a way to aggregate event (definition 1) data.

Note: In Optimize, KPIs (more specifically, KPI instances) are sometimes referred to as "monitors."

KPI definition

The characteristics of a KPI, as distinct from a KPI instance.

KPI instance

A measurement of one slice of business data or system data. A Total Order Revenue KPI, which is not sliced by dimensions, has one KPI instance. A Revenue by Customer KPI, which slices revenue by the customer dimension, has multiple KPI instances. With a Revenue by Customer KPI, each KPI instance measures revenue for an individual customer. A Revenue by Region and Sales Person KPI, which slices revenue by a sales hierarchy with two dimensions (region and sales person), has multiple KPI instances. For the Revenue by Region and Sales Person KPI, each KPI instance measures revenue for an individual sales person within a specific region.

KPI rule

A rule that evaluates data over one or more KPI data collection intervals. For example, a Gross Margin Percent Severely Below Normal rule might specify that the gross margin percentage must not go more than one sigma below the average Gross Margin KPI value for three consecutive Gross Margin KPI collection intervals. A KPI rule takes dimensions into account, tracks its state of compliance, and persists in the database.

LDAP

Lightweight Directory Access Protocol. An Internet protocol for accessing information directories containing names, phone numbers, addresses, etc. It allows client programs to query LDAP directory servers about entries using their attributes.

logical server

A representation of a webMethods product component. Each logical server contains subcomponents that can be configured using the Central Configuration tool. During the environment configuration process, each logical server is mapped to one or more physical servers.

mashup

A Web application that integrates content from multiple sources into a seamless user experience.

measure (formerly fact)

A metric with a set of attributes including a unit of measure and a calculation type.

message flow

A transition that sends a message from one step to another. For example, a message flow might send a message from a step in an internal business process to a step in an external business process.

model

A visual overview diagram of an entire business process. Business process models are created in Software AG Designer or other external tools.

monitor

1. A KPI instance.
2. The mechanism Optimize uses to store data for a KPI instance.

Monitor

See webMethods Monitor.

monitor repository

The total number of monitors and the current queue size of the Analytic Engine. A large queue size might indicate that the Analytic Engine is attempting to process too much data.

monitored component

A system component monitored by Infrastructure Data Collector.

monitoring API

An interface to Optimize that provides a standardized means for loading monitor, process, and metadata into Optimize's Analytic Engine within the webMethods product suite.

My webMethods

A Web-based monitoring and administration user interface for managing all products in a webMethods integration environment. When you log in to My webMethods, you are logging in to all webMethods products that are incorporated into My webMethods.

My webMethods Server

The core set of components and services required to host the Web interface for webMethods products including Optimize. Formerly known as Portal.

Optimize

See webMethods Optimize.

Optimize for Infrastructure

A BAM component that enables you to access the resource information collected by Infrastructure Data Collector. You use Optimize for Infrastructure to monitor the status of individual managed objects as well as the overall status of your system.

Optimize for Process

A BAM component that monitors business processes, in addition to managed objects and system status, by tracking KPIs that relate to operational factors such as margin, revenue, customer satisfaction, and inventory levels. Optimize for Process enables you to use Six Sigma on your business processes. Optimize for Process measures the number of defects in a process and helps you systematically determine how to eliminate them. Optimize for Process includes Business Visualization tools that provide at-a-glance mashups of Optimize monitoring information.

Pareto principle

A principle that states that the majority of defects (80 percent) are caused by a few (20 percent) vital components.

pipeline data

Data available only within a given process, rather than retrieved from or sent to external sources. Specifically, pipeline data is all information (documents and other data types) that is available to any given step because that data was introduced upstream in the process.

pool

1. A set of associated steps in a business process. A pool of steps represents either an internal process or an external process.
2. See database connection pool.

portlet

Interchangeable user interface components that are managed and displayed in a Web portal. Portlets produce fragments of markup code that are aggregated into a portal page. Typically a portal page is displayed as a collection of non-overlapping portlet windows, where each portlet window displays a portlet. Hence, a portlet (or collection of portlets) resembles a Web-based application that is hosted in a portal.

Process Analytics

An Optimize function that helps users analyze information about the performance of business processes and rules.

process error

An error that originates within a business process and applies to the process as a whole. Examples of process errors include processes that timed out before they completed successfully as well as processes that completed with errors.

process event

An event that specifies what should happen when an external process encounters a particular situation. For example, if a process contains a manual review step and the system determines that the data it receives in that step is corrupt, a process event might cause Optimize to stop analyzing the process.

process instance

An execution of a business process. In an order process, each order that goes through the steps in the process is a process instance.

Process Tracker

An Analytic Engine component that receives business process data from the Web Service Data Collectors and other webMethods products, and tracks the state of executing processes. As a process executes, the Process Tracker takes measurements about the process instance and its steps, such as wait time and cycle time. This information can be used to track performance of a business process or to alert you when certain conditions exist, such as performance degradation or a failure in process execution.

referenced process

A separate process that displays as a step within another business process.

reminder interval

The number of minutes between alerts. If you want alerts specified for a rule to be repeated, specify this interval in the Attributes settings for the rule.

rule

A condition or set of conditions you apply to a process or to a type of data. You can create a rule to notify you when a process or resource becomes problematic and requires user attention. You can also create a rule to notify you when a business opportunity exists. Optimize supports the following two types of rules: KPI rules, event rules, and threshold rules.

rule evaluation interval

The number of consecutive intervals for which a rule must evaluate to true before Optimize will consider the rule to be violated. For example, if you are writing a rule to detect when a queue length is trending toward being too large, you might specify that Optimize should not consider the rule violated unless the queue length trends upward for five intervals.

rule expression

A logical expression contained in a rule. A simple expression tests a single condition; a complex expression tests two or more conditions. Optimize evaluates the expression to determine whether it is true or false. If the expression evaluates to true, Optimize generates a rule violation.

rule filter

A filter that identifies specific processes, dimensions, or system objects that a rule should monitor.

rule instance

An occurrence of a rule. For example, a rule specifies that revenue by region requires attention when the revenue drops below \$10,000 over a 12-hour period. One instance of that rule occurs during the process of monitoring revenue for a specific region at each 12-hour period.

rule violation

A rule instance that meets the conditions specified by a rule. Under these conditions, a business process or system resource requires user attention, either because a problem has occurred or because an opportunity exists.

rules engine

An Analytic Engine component that compares the data it receives from the Analytic Engine to rules that detect when a process or resource requires user attention.

severity

A user-defined text field associated with a rule. If text is entered into this field, that text appears with the rule in the rule list, and any alerts generated for that rule can be configured to display the severity text.

Six Sigma

A disciplined methodology for improving business process performance by eliminating defects.

SLA

Service Level Agreement. An agreement between two parties on the level of service to be provided.

SNMP

Simple Network Management Protocol. A protocol for monitoring and controlling network devices. It has nearly universal use in TCP/IP networks.

SNMP alert

A user-configured SNMP message sent to a defined SNMP manager upon a rule violation.

SNMP data collector

A component of Infrastructure Data Collector specifically configured to collect operational data from defined SNMP Agents.

SOAP

Simple Object Access Protocol.

source step

A source step relates to wait-time standardized error types.

SQL

Structured Query Language. A database query language that interfaces with relational databases.

Standardized error

User-defined process error tracked by Optimize. Standardized errors use built-in error conditions that can be configured in the Add Standardized Error page error builder.

statistical interval

The time range for historical average KPI values.

step

The basic unit of work in a business process. A step can represent an automated process, a manual step performed by a person, or a visual aid that identifies a task that an external organization performs.

Step Absolute Wait Time

See wait-time standardized error type.

step error

An error that applies to a step within a business process. Examples of step errors include duplicate step processing, step timeouts, and steps executed out of sequence.

Step Relative Wait Time

See wait-time standardized error type.

sticky rule

A rule that when violated does not return to compliance until the rule is manually resolved. You can make a rule sticky to signal a serious violation that requires user attention to remedy the situation. For example, if you create a rule to detect that a Broker Server is offline, defining the rule as sticky causes the alert to occur once and the rule to remain in violation until the server is restarted and the rule is resolved.

subprocess

A process within another business process. The steps within a subprocess are often grouped within a container. The subprocess is treated like a step within the process that contains it.

swarm line

A transition that shows the relative volume of process instances traveling down a path in a business process. The number of dots in the swarm line indicates the relative volume moving along the transition.

swimlane

A set of associated steps within a pool.

system component

Any item of information technology (IT) infrastructure equipment in your environment that can be monitored.

system data

Data about equipment or applications, such as queue length and whether a managed component is online or offline.

system error

Built-in error that is automatically associated with a process by Process Tracker.

system KPI

A KPI created by a user to monitor system data. For example, users can create system KPIs to monitor Integration Server Total Memory. A system KPI differs from an intrinsic KPI, which is created by Optimize.

system tree

A listing of system components with branches and nodes that is displayed on the Analytics Overview page in Optimize.

threshold rule

A rule that evaluates data as it is collected (for example, queue size greater than 1000 or order amount less than \$500). A threshold rule does not take dimensions into account, does not track its state of compliance, and does not persist in the database. A threshold rule is like a trigger that produces an alert.

transition

The flow of data and control from one step to another in a business process. The two basic types of transitions are control flow and message flow.

transaction

(In an event map) An attribute that you do not want to measure but would like to store in the database to use in a rule expression or to associate with a fact that arrives with the same event.

trend

In Optimize, one of 16 different types of diagnoses (above normal, below normal, staying in upward trend, staying in downward trend, etc.) on all KPI instances over the KPI's data collection interval. You can use this information to watch for trends in performance and to help determine ways to improve business processes and system components that continue to perform outside their normal range.

wait-time standardized error type

Wait-time error types are intended for configuring errors related to the time that elapses leading to the start of a specified step in a process instance (the destination step). Wait-time error types are always applied to the destination step. The two available wait-time error types are Step Relative Wait Time and Step Absolute Wait Time. A Step Relative Wait Time error type relates to the time that elapses between the completion of one step in a process instance (the source step) and the start of a subsequent, specified step in that same process instance (the destination step). A Step Absolute Wait Time error type relates to the time that elapses between the beginning of a process instance and the start of a specified step in that process instance.

Web services

Self-contained applications that exchange data over the Web or another network and are based on open standards such as SOAP, TCP/IP, HTTP, Java, and XML.

Web Service Data Collector

Also referred to as WSDC. An interface to Optimize through which client programs can send any of the following information:

- Process and operational data about applications or IT equipment in your enterprise, such as databases, printers, or disk drives.
- Process data, such as the start and end time of each step in process instances.
- Business data, such as customers, order quantities, and revenues.

Web services layer

An Analytic Engine component that sends the results of the analysis to the My webMethods Server. The My webMethods user interface presents the results, along with icons that indicate that a rule violation has occurred or that Optimize has diagnosed a trend.

webMethods Broker

The primary “message backbone” product in a webMethods integration environment. Broker facilitates asynchronous, message-based communication using the publish-and-subscribe model.

webMethods Integration Server

The central run-time product in a webMethods integration environment. Integration Server provides an environment for the orderly, efficient, and secure execution of business processes and services.

webMethods Monitor

An administrative tool used to examine instances of running or completed business processes, services, integrations, and documents. Monitor allows you to view data about business processes, services, and documents that other webMethods products have previously logged.

webMethods Optimize

The Business Activity Monitoring (BAM) product in a webMethods integration environment. Optimize facilitates real-time insight into key business process and system resource activities, and provides actionable information that enables business users to make informed decisions, quickly address problems, and enhance business operations through continuous process improvement.

webMethods-executed process

A business process that is modeled, executed, and monitored using webMethods-provided tools. Compare to externally-executed process.

WSDC

See Web Service Data Collector.

XML

Extensible Markup Language. A flexible markup language for structuring, storing, and sending information.

Index

A

- About page 50
- Action Name list, configuring 57
- actions
 - sending SNMP alerts 66
 - triggering a Web service 282
- adapter
 - KPI definitions 426
- adapters
 - webMethods 33
- Add BusinessProcesses page 137
- Add Environment page 54
- Add KPI Filter dialog box 133
- Add Tracked Process dialog box 131, 131, 132
- Add/Edit Dimension page 145
- Add/Edit Event Map page 151, 156, 160, 163
- Add/Edit EventMap page 164
- Add/Edit KPI page 169, 171
- Add/EditPool panel 56
- addDimension method 260
- addDimension method 506, 507
- addDimensions method 260, 506, 507
- addEvent method 257, 503, 505
- addEvents method 257, 503, 505
- AddKPI Privileges page 178
- administrator tasks 37
- alerts
 - sending by way of SNMP 66
- alerttemplate
 - creating 63
- Analysis database component 319, 330, 335
- Analysis database component set 319
- analysis engine
 - description 29
- Analytic Engine
 - adjusting RAM setting 91
 - database activities 324
 - description 29
 - diagnoses 34
 - log files 232
 - logical server 55
 - optimizing performance 79
 - starting on a UNIX system 40
 - starting on a Windows system 43
 - stopping 44
 - viewing information about 49

- AnalyticEngine
 - e-mail alerts 58
- attribute definitions
 - Broker 361
- attributes
 - adding to dimension 146, 148
 - data maintenance 333
 - dimensions 146, 148
 - e-mail alerts 58
 - monitored components 229
 - transaction 34, 153, 157, 538
- Attributespanel 229

B

- Broker 27
- Broker auto-discovery 201
- Broker Servers
 - monitoring, description 32
- Broker Server
 - logical server 55
- BrokerServers
 - monitoring, description 198
- business process
 - models 36
- Business Activity Monitoring
 - components 26
 - overview 26
- business analyst tasks 37
- Business Data page 141
- business process
 - analysis enabled 102
 - associating with facts 150
 - associating with dimensions 150
 - dimensions 34
 - editing 103
 - event maps 34, 172
 - execution enabled 102
 - externally executed 100
 - externally-executed 35
 - identifying processes to monitor 101
 - KPI hierarchies 34
 - KPIs 34
 - types of 35, 100, 136
 - webMethods-executed 35, 100
- Business Processes page 101, 103
- Business Processes panel 101
- businessdata, configuring 140

C

- cache file, importing 70
- cache configuration
 - defining 95
 - defining 95, 95, 95
- Certification Authority certificate, importing 70
- changing port for InfraDC 46
- ClusterPortnumber 55
- communication (port) settings, specifying 68
- components
 - monitored 225
- components
 - Business Activity Monitoring 26
 - database component sets 318
 - Optimize 27, 48
- composite KPIs
 - defining 167
- configuring
 - data
 - hierarchies 164
- configuration settings
 - changing 57
- configuring
 - Action Name list on Edit Rule page 57
 - alert SNMP settings 66
 - Analytics Overviewpage display hierarchy 126
 - communication (port) settings 68
 - data
 - business 140
 - dimensions 143
 - events 150
 - KPI hierarchies 164
 - KPIs 167
 - system 142
 - KPIs 140
 - Web service actions 282, 282

D

- data analysis 30
- data collection 30
 - Web services 32
- data collection intervals 34
- Data Collection Services
 - logical server 55
- data collectionintervals 170
- data definition on 30
- data level security

- KPI 177
- data maintenance
 - attributes 333
 - settings 80, 333
- data purge
 - DB2 332
 - Oracle 331
 - SQL Server 331
- Data Purge database component 319, 330
- data storage 30
- database activities
 - Analytic Engine 324
 - connect to Optimize database componentsets 322
 - mapping to database connectionpools 326
- database component sets 318
 - Analysis 319
 - Optimize 322
 - Process Audit Log 319
 - Process Tracker 319
- database components
 - Analysis 319, 335
 - Data Purge 319, 330
 - Distributed Locking 319
 - IS Core Audit Log 319
 - Operation Management 319
 - Optimize for Process 30
 - order of installation 318
 - Process Audit Log 319
 - Process Engine 319
 - Process Tracker 319, 334, 335
- database connection pools 322
 - mapping 56, 326
- database connectionpools
 - defining 325
- Database PoolConfiguration page 325
- databaseconnection pools
 - configuring 322
- DB2
 - data purge 332
 - database schema URL options 325
- default environment 54
- Define Environments page 54, 58, 282
- Define Environmentspage 326
- defining data 30
- Designer 255
- diagnoses
 - description 34
- dialog boxes

- Add KPI Filter 133
- AddTracked Process 131, 131, 132
- Select KPI to Add 132
- dimensions
 - adding and editing 145
 - adding and editing 147
 - associating with business processes 150
 - attributes 146, 148
 - collecting data 147, 149
 - defining 143
 - description 34
 - system-defined 147
 - viewing 144
- Dimensions page 144
- Distributed Locking database component 319
- documentation
 - using effectively 21
- E**
- Edit Component Type page 228
- Edit Environment page 58, 282
- Edit Environment page 55
- Edit Hierarchical page 166
- Edit Role page 136, 177
- Editing the Component Tree 133
- EditProcess page 103
- EditRole page 178
- e-mail alerts
 - admin address 59
 - configuring 58
 - creating an alert template 63
 - default sender 59
 - default text encoding 59
 - Oracle purge operation 338
 - specifying SMTP mail server 58
 - template for e-mail body and subject 59
 - template attribute 59
- endpoints
 - mapping 56
- Environment Information panel 54
- environments
 - adding 54
 - default 54
 - deploying 56
 - description 48, 54
 - validating 56
- event rules
 - description 36
- event handler
 - description 29
- event maps
 - description 34, 172
 - viewing 150
- events
 - mapping 150
- externally executed business process 100
- externally-executed business process 35
- F**
- facts
 - associating with business process 153, 157
 - associating with business process 150
- Field Mappings 156, 160, 163
- G**
- Gateway
 - KPI definitions 373
- H**
- hierarchies (KPI), defining 164
- historical average values
 - modes of operation 78
- hosts
 - defining 56
- I**
- identifying processes to monitor 101
- individual KPIs
 - defining 167
- Infra DC
 - changing port for 46
- InfraDC
 - description 32, 198
 - scaling 197
 - starting 46
- Infrastructure
 - overview 26
- Infrastructure Data Collector
 - SNMP data 31
 - starting 46
- Integration Server
 - KPI definitions 375, 408
- Integration Servers
 - monitored components 225
 - monitoring, description 32
 - services 33

IntegrationServers
 monitoring, description 198
 IS Core Audit Log database component 319

J

Java Database Connectivity 322
 JDBC connectionpools
 configuring 322
 JMS queue 31, 182, 196, 196
 JNDI settings 81
 Journal Logging 233

K

key performance indicators (KPIs)
 data level security 177
 key performance indicators (KPIs)
 hierarchies 164
 key performance indicators (KPIs)
 description 34
 dimensions, defining 143
 granting access 177
 hierarchies 34
 key performance indicators(KPIs)
 individual 167
 key performance indicators (KPIs)
 composite 167
 events, mapping 150
 key performance indicators (KPIs) 26
 key performance indicators (KPIs)
 configuring 140
 defining 167
 KPI definitions
 adapter 426
 Gateway 373
 Integration Server 375, 408
 Territory 371
 Trading Networks 400
 webMethods Broker 361
 KPI deletion script 179
 KPI Hierarchies page 165
 KPI InstanceDetail page 199
 KPI rules
 description 36
 KPIs 26
 deleting 179

L

log files 232

log information
 specifying 338
 logical servers
 adding 55
 configuring 55
 mapping 56
 renaming 55

M

mail
 attributes 58
 Mail Settings 58
 mapping events 150
 measures
 description 34
 methods
 pushExecutionError 277
 methods
 addDimension 260, 506, 507
 addDimensions 260, 506, 507
 addEvent 257, 503, 505
 addEvents 257, 503, 505
 pushExecutionError 278, 508
 pushProcessControlOperation 510
 pushProcessControlOperation 262, 265
 pushStepControlOperation 269, 273, 514
 MIB configuration file 185
 Monitor 102
 monitor data
 publishing 255
 Monitored Components page 226
 monitored components
 attributes 229
 description 225
 displaying list of 226
 monitoredcomponents
 configuring 228
 monitoring
 large numbers of objects 197
 Monitoring API 31, 182, 196, 196
 monitors
 description 34
 types 35
 My webMethodsServer
 stopping 45
 My webMethods
 user interface 29
 My webMethods Server

- logical server 55
 - starting on a UNIX system 40
 - starting on a Windows system 43
 - viewing information about 50
- O**
- Objects panel 230
 - OPERATION_PARAMETER table 338
 - Operation Management database component 319
 - OPERATION_LOG table 338, 338
 - OPERATION_MGR table 338, 338
 - OPERATION_PARAMETER table 338
 - OPERATION_PARAMETER table 335
 - Optimize
 - implementation (typical) 27
 - main components 27
 - starting on a UNIX system 40
 - starting on a Windows system 42
 - stopping 44
 - Optimize Central Configurator
 - e-mail alerts 58
 - Optimize components
 - viewing information about 48
 - Optimize for Infrastructure
 - alerts 199
 - notifications 199
 - Optimize for Infrastructure, description 199
 - Optimize for Process
 - database components 30
 - overview 26
 - Optimize for Process
 - SixSigma use in 26
 - Oracle
 - data purge 331
 - required permissions 330
 - Overview Configuration page 126
- P**
- pages
 - Add/Edit Event Map 160
 - pages
 - About 50
 - Add BusinessProcesses 137
 - Add Environment 54
 - Add KPI Privileges 178
 - Add/Edit KPI 171
 - Add/Edit Event Map 151, 156, 163, 164
 - Add/Edit Dimension 145
 - Add/Edit KPI 169
 - Business Data 141
 - Business Processes 101, 103
 - DatabasePool Configuration 325
 - Define Environments 54, 58, 282, 326
 - Dimensions 144
 - Edit Component Type 228
 - Edit Environment 55, 58, 282
 - Edit Hierarchy 166
 - Edit Process 103
 - Edit Role 136, 177, 178
 - KPI Hierarchies 165
 - MonitoredComponents 226
 - OverviewConfiguration 126
 - Roles 136, 178
 - System Data 142
 - System Information 48
 - System Settings 70
 - panels
 - Add/Edit Pool 56
 - Attributes 229
 - Business Processes 101
 - Environment Information 54
 - Objects 230
 - Process Diagram 105
 - ServerSettings 70
 - SNMP Alert Settings 66
 - performance
 - optimizing Analytic Engine 79
 - port, changing for InfraDC 46
 - ports, specifying 68
 - PredictionEngine
 - viewing information about 49
 - privileges, SYSDBA 331
 - Process Audit Log database component 319
 - Process Audit Log database component set 319
 - Process Engine 35, 101, 101
 - Process Engine database component 319
 - Process Tracker
 - description 29
 - settings 334
 - Process Tracker database component 319, 335
 - Process Tracker database component set 319
 - Process Tracker database component 334
 - ProcessDiagram panel 105
 - processes 100
 - ProcessTracker database component 330
 - purge
 - configuring 333

- running 331
- purge attributes
 - setting 333
- purge data
 - DB2 332
 - Oracle 331
 - SQL Server 331
- purge data from database components 330
- purge operation
 - setting parameters 335
 - specifying people to notify 338
 - viewing events 338
- pushExecutionError method 277, 278, 508
- pushProcessControlOperation method 510
- pushProcessControlOperation method 262, 265
- pushStepControlOperation method 514
- pushStepControlOperation method 269, 273

R

- RAM, adjusting for the Analytic Engine and Web Service Data Collectors 91
- Roles page 136, 178
- rule filters
 - description 36
- rule violations
 - alerting 57
- rules
 - alerts and actions 57
 - event rules 36
 - KPI rules 36
 - overview 36
 - threshold rules 36
- rules engine
 - description 29

S

- scaling InfraDC 197
- secure connections between subcomponents 70
- Select KPI to Add dialog box 132
- Server Settings panel 70
- servers
 - configuring 55
 - logical 55
 - mapping 56
- settings
 - configuration 57
 - data maintenance 80, 333
 - JNDI 81

- mail 58
- monitored component 229
- Process Tracker 334
- Six Sigma 26
- SNMP, sending alerts by way of 66
- SNMP Alert Settings panel 66
- SQL Server
 - data purge 331
- SSL 70
- starting
 - InfraDC 46
- statistical intervals
 - description 78
- storing data 30
- System Data page 142
- system data, configuring 142
- System Information page 48
- System Settings page 70

T

- tables
 - OPERATION_LOG 338, 338
 - OPERATION_MGR 338, 338
 - OPERATION_PARAMETER 338
 - OPERATION_PARAMETER 335, 338
- tasks performed by administrators 37
- tasks performed by business analysts 37
- templates
 - alert 59
- Territory
 - KPI definitions 371
- threshold rules
 - description 36
- tracked processes, identifying 101
- Trading Networks
 - KPI definitions 400
- transaction attributes 153, 157, 538

U

- UNIX system
 - starting Optimize 40

W

- Web service actions 282
 - configuring 282
- Web Service Data Collector
 - addDimensions method 260
- Web Service Data Collector 32, 34, 101

- addDimension method 260
- addEvent method 257
- addEvents method 257
- adjusting RAM setting 91
- generating the client using Designer 255
- interfaces 254
- log files 232
- pushExecutionError method 278
- pushExecutionError method 277
- pushProcessControlOperation method 262, 265
- pushStepControlOperation method 269, 273
- starting on a UNIX system 40
- starting on a Windows system 43
- viewing information about 49
- WSDLs 255
- Web Service Data Collector
 - stopping 45
- Web services 32, 33
- Web Services Definition Language 254
- Web services layer
 - description 29
- webMethods Broker 27
- webMethods Broker
 - KPI definitions 361
- webMethods Central Configuration
 - changing settings 57
- webMethods Designer 100, 255
 - designing business process models 36
- webMethods Infrastructure Data Collector. 32
- webMethods Infrastructure Data Collector. 198
- webMethods Monitor 102
- webMethods Process Engine 101
- webMethods-executed business process 35, 100
- webMethods Optimize for Infrastructure. 199
- Windows system
 - starting Optimize 42
- WmOptimize
 - pushExecutionError method 508
- WmOptimize
 - addDimension method 506, 507
 - addDimensions method 506, 507
 - addEvent method 503, 505
 - addEvents method 503, 505
 - pushProcessControlOperation method 510
 - pushStepControlOperation method 514
- wsd2java utility 255
- WSDLs for Web Service Data Collector 255