



Informatica®
10.2 HotFix 2

Application Service Guide

Informatica Application Service Guide
10.2 HotFix 2
April 2019

© Copyright Informatica LLC 2014, 2019

This software and documentation are provided only under a separate license agreement containing restrictions on use and disclosure. No part of this document may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording or otherwise) without prior consent of Informatica LLC.

Informatica, the Informatica logo, PowerCenter, and PowerExchange are trademarks or registered trademarks of Informatica LLC in the United States and many jurisdictions throughout the world. A current list of Informatica trademarks is available on the web at <https://www.informatica.com/trademarks.html>. Other company and product names may be trade names or trademarks of their respective owners.

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation is subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License.

Portions of this software and/or documentation are subject to copyright held by third parties. Required third party notices are included with the product.

The information in this documentation is subject to change without notice. If you find any problems in this documentation, report them to us at infa_documentation@informatica.com.

Informatica products are warranted according to the terms and conditions of the agreements under which they are provided. INFORMATICA PROVIDES THE INFORMATION IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.

Publication Date: 2019-04-18

Table of Contents

Preface	20
Informatica Resources.	20
Informatica Network.	20
Informatica Knowledge Base.	20
Informatica Documentation.	20
Informatica Product Availability Matrixes.	21
Informatica Marketplace.	21
Informatica Global Customer Support.	21
 Chapter 1: Analyst Service.....	 22
Analyst Service Overview.	22
Analyst Service Architecture.	23
Configuration Prerequisites.	24
Services Associated with the Analyst Service.	24
Flat File Cache Directory.	24
Export File Directory.	25
Attachments Directory.	25
Keystore File.	25
Exception Management Audit Database.	25
Recycle and Disable the Analyst Service.	26
Properties for the Analyst Service.	26
General Properties for the Analyst Service.	27
Model Repository Service Properties.	27
Logging Options.	28
Human Task Properties.	28
Run-time Properties.	28
Metadata Manager Service Properties.	29
Business Glossary Properties.	29
Custom Properties for the Analyst Service.	29
Custom Images in the Analyst Tool.	29
Process Properties for the Analyst Service.	30
Node Properties for the Analyst Service Process.	30
Analyst Security Options for the Analyst Service Process.	30
Advanced Properties for the Analyst Service Process.	31
Custom Properties for the Analyst Service Process.	31
Environment Variables for the Analyst Service Process.	32
Creating and Configuring the Analyst Service.	32
Creating an Analyst Service.	33

Chapter 2: Content Management Service.....	34
Content Management Service Overview.	34
Master Content Management Service	35
Content Management Service Architecture.	35
Probabilistic Models and Classifier Models.	36
Reference Data Warehouse.	37
Orphaned Reference Data.	37
Deleting Orphaned Tables	38
Recycling and Disabling the Content Management Service.	38
Content Management Service Properties.	39
General Properties.	39
Multi-Service Options.	40
Associated Services and Reference Data Location Properties.	40
File Transfer Options.	41
Logging Options.	41
Custom Properties for the Content Management Service.	41
Content Management Service Process Properties.	42
Content Management Service Security Options.	42
Address Validation Properties.	43
Identity Properties.	46
Advanced Properties.	46
NLP Options.	47
Custom Properties for the Content Management Service Process.	47
Creating a Content Management Service.	47
 Chapter 3: Data Integration Service.....	 49
Data Integration Service Overview.	49
Before You Create the Data Integration Service.	50
Create Required Databases.	50
Create Connections to the Databases.	50
Create the Service Principal Name and Keytab File.	51
Create Associated Services.	51
Creating a Data Integration Service.	51
Data Integration Service Properties.	54
General Properties.	54
Model Repository Properties.	55
Execution Options.	56
Logical Data Object/Virtual Table Cache Properties.	59
Logging Properties.	60
Pass-through Security Properties.	60
Modules.	61
HTTP Proxy Server Properties.	61

HTTP Configuration Properties	62
Result Set Cache Properties.	63
Mapping Service Properties.	63
Profiling Warehouse Database Properties.	63
Advanced Profiling Properties.	64
SQL Properties.	65
Workflow Orchestration Service Properties.	66
Web Service Properties.	66
Custom Properties for the Data Integration Service.	67
Data Integration Service Process Properties.	67
Data Integration Service Security Properties.	68
HTTP Configuration Properties	68
Result Set Cache Properties.	69
Advanced Properties.	69
Logging Options	70
SQL Properties.	70
Custom Properties for the Data Integration Service Process.	70
Environment Variables.	70
Data Integration Service Compute Properties.	70
Execution Options.	71
Environment Variables.	72
Operating System Profiles for the Data Integration Service.	72
Operating System Profile Components.	73
Configuring the Data Integration Service to Use Operating System Profiles.	73
Troubleshooting Operating System Profiles.	75
High Availability for the Data Integration Service.	75
Data Integration Service Restart and Failover.	76
Data Integration Service Recovery.	77
Chapter 4: Data Integration Service Architecture.....	78
Data Integration Service Architecture Overview.	78
Data Integration Service Connectivity.	79
Data Integration Service Components.	80
Service Components.	81
Mapping Service Module.	81
Profiling Service Module.	81
SQL Service Module.	82
Web Service Module.	82
Workflow Orchestration Service Module.	82
Data Object Cache Manager.	83
Result Set Cache Manager.	83
Deployment Manager.	83
Logical Data Transformation Manager.	83

Compute Component.	84
Execution Data Transformation Manager.	84
DTM Resource Allocation Policy.	85
Processing Threads.	85
Output Files.	85
Process Where DTM Instances Run.	87
In the Data Integration Service Process.	88
In Separate DTM Processes on the Local Node.	88
In Separate DTM Processes on Remote Nodes.	89
Single Node.	89
Grid.	90
Logs.	90
 Chapter 5: Data Integration Service Management.	 92
Data Integration Service Management Overview.	92
Enable and Disable Data Integration Services and Processes.	93
Enable, Disable, or Recycle the Data Integration Service.	93
Enable or Disable a Data Integration Service Process.	95
Directories for Data Integration Service Files.	95
Source and Output File Directories.	95
Control File Directories.	97
Log Directory.	97
Output and Log File Permissions.	98
Run Jobs in Separate Processes.	98
DTM Process Pool Management.	99
Rules and Guidelines when Jobs Run in Separate Processes.	100
Maintain Connection Pools.	100
Connection Pool Management.	101
Pooling Properties in Connection Objects.	101
Example of a Connection Pool.	102
Optimize Connection Performance.	102
PowerExchange Connection Pools.	103
PowerExchange Connection Pool Management.	103
Connection Pooling for PowerExchange Netport Jobs.	104
PowerExchange Connection Pooling Configuration.	104
Maximize Parallelism for Mappings and Profiles.	106
One Thread for Each Pipeline Stage.	107
Multiple Threads for Each Pipeline Stage.	108
Maximum Parallelism Guidelines.	109
Enabling Partitioning for Mappings and Profiles.	110
Optimize Cache and Target Directories for Partitioning.	110
Result Set Caching.	111
Data Object Caching.	111

Cache Tables.	112
Data Object Caching Configuration.	112
Data Object Cache Management.	113
Configure User-Managed Cache Tables.	115
Persisting Virtual Data in Temporary Tables.	116
Temporary Table Implementation.	117
Temporary Table Operations.	117
Rules and Guidelines for Temporary Tables.	119
Content Management for the Profiling Warehouse.	119
Creating and Deleting Profiling Warehouse Content.	119
Database Management.	120
Purge.	120
Tablespace Recovery.	122
Database Statistics.	123
Web Service Security Management.	124
HTTP Client Filter.	125
Pass-through Security.	125
Pass-Through Security with Data Object Caching.	126
Adding Pass-Through Security	126
Chapter 6: Data Integration Service Grid.	128
Data Integration Service Grid Overview.	128
Grid Configuration by Job Type.	129
Before You Configure a Data Integration Service Grid.	130
Grid for Jobs that Run in the Service Process.	130
Example Grid that Runs Jobs in the Service Process.	131
Rules and Guidelines for Grids that Run Jobs in the Service Process.	132
Configuring a Grid that Runs Jobs in the Service Process.	132
Grid for Jobs that Run in Local Mode.	135
Example Grid that Runs Jobs in Local Mode.	137
Rules and Guidelines for Grids that Run Jobs in Local Mode.	137
Configuring a Grid that Runs Jobs in Local Mode.	138
Grid for Jobs that Run in Remote Mode.	141
Supported Node Roles.	141
Job Types.	142
Example Grid that Runs Jobs in Remote Mode.	143
Rules and Guidelines for Grids that Run Jobs in Remote Mode.	144
Recycle the Service When Jobs Run in Remote Mode.	144
Configuring a Grid that Runs Jobs in Remote Mode.	145
Logs for Jobs that Run in Remote Mode.	148
Override Compute Node Attributes to Increase Concurrent Jobs.	149
Grid and Content Management Service.	150
Maximum Number of Concurrent Jobs on a Grid.	151

Editing a Grid.	152
Deleting a Grid.	153
Troubleshooting a Grid.	153

Chapter 7: Data Integration Service Applications..... 156

Data Integration Service Applications Overview.	156
Applications View.	157
Applications.	157
Application State	157
Application Properties.	157
Deploying an Application.	158
Enabling an Application	159
Renaming an Application.	159
Starting an Application.	160
Backing Up an Application.	160
Restoring an Application.	160
Refreshing the Applications View	161
Logical Data Objects.	161
Physical Data Objects.	162
Mappings.	162
SQL Data Services.	164
SQL Data Service Properties.	164
Enabling an SQL Data Service.	167
Renaming an SQL Data Service.	168
Web Services.	168
Web Service Properties.	168
Enabling a Web Service.	170
Renaming a Web Service.	171
Workflows.	171
Workflow Properties.	171
Enabling a Workflow.	171
Starting a Workflow.	172

Chapter 8: Metadata Manager Service..... 173

Metadata Manager Service Overview.	173
Configuring a Metadata Manager Service.	174
Creating a Metadata Manager Service.	175
Metadata Manager Service Properties.	176
Database Connect Strings.	179
Overriding the Repository Database Code Page.	179
Creating and Deleting Repository Content.	180
Creating the Metadata Manager Repository.	180
Restoring the PowerCenter Repository	180

Deleting the Metadata Manager Repository	181
Enabling and Disabling the Metadata Manager Service.	181
Metadata Manager Service Properties.	182
General Properties.	182
Metadata Manager Service Properties.	183
Database Properties.	184
Configuration Properties.	187
Connection Pool Properties.	187
Advanced Properties.	188
Custom Properties for the Metadata Manager Service.	189
Configuring the Associated PowerCenter Integration Service.	189
Privileges for the Associated PowerCenter Integration Service User.	190
Chapter 9: Model Repository Service.....	192
Model Repository Service Overview.	192
Model Repository Architecture.	193
Model Repository Objects.	193
Model Repository Connectivity.	193
Model Repository Database Requirements.	194
IBM DB2 Database Requirements.	195
IBM DB2 Version 9.1.	196
Microsoft SQL Server Database Requirements.	196
Oracle Database Requirements.	196
Enable and Disable Model Repository Services and Processes.	196
Enable, Disable, or Recycle the Model Repository Service.	197
Enable or Disable a Model Repository Service Process.	197
Properties for the Model Repository Service.	198
General Properties for the Model Repository Service.	198
Repository Database Properties for the Model Repository Service.	199
Search Properties for the Model Repository Service.	201
Advanced Properties for the Model Repository Service.	201
Cache Properties for the Model Repository Service.	202
Versioning Properties for the Model Repository Service.	202
Custom Properties for the Model Repository Service.	203
Properties for the Model Repository Service Process.	203
Node Properties for the Model Repository Service Process.	203
High Availability for the Model Repository Service.	206
Model Repository Service Restart and Failover.	206
Model Repository Service Management.	206
Content Management for the Model Repository Service	207
Model Repository Backup and Restoration.	207
Security Management for the Model Repository Service.	209
Search Management for the Model Repository Service	209

Repository Log Management for the Model Repository Service.	211
Audit Log Management for the Model Repository Service	211
Cache Management for the Model Repository Service.	212
Version Control for the Model Repository Service.	213
How to Configure and Synchronize a Model Repository with a Version Control System.	213
Repository Object Administration.	216
Objects View.	216
Locked Object Administration.	217
Versioned Object Administration.	217
Troubleshooting Team-based Development.	218
Creating a Model Repository Service.	218
Chapter 10: PowerCenter Integration Service.....	220
PowerCenter Integration Service Overview.	220
Creating a PowerCenter Integration Service.	221
Enabling and Disabling PowerCenter Integration Services and Processes.	223
Enabling or Disabling a PowerCenter Integration Service Process.	223
Enabling or Disabling the PowerCenter Integration Service.	223
Operating Mode.	224
Normal Mode.	225
Safe Mode.	225
Running the PowerCenter Integration Service in Safe Mode.	225
Configuring the PowerCenter Integration Service Operating Mode.	227
PowerCenter Integration Service Properties.	228
General Properties.	228
PowerCenter Integration Service Properties.	229
Advanced Properties.	231
Operating Mode Configuration.	233
Compatibility and Database Properties.	233
Configuration Properties.	235
HTTP Proxy Properties.	237
Custom Properties for the PowerCenter Integration Service.	237
Operating System Profiles for the PowerCenter Integration Service.	238
Operating System Profile Components.	238
Configuring Operating System Profiles.	238
Troubleshooting Operating System Profiles.	239
Associated Repository for the PowerCenter Integration Service.	239
PowerCenter Integration Service Processes.	240
Code Pages.	240
Directories for PowerCenter Integration Service Files.	241
Directories for Java Components.	242
General Properties.	242
Custom Properties for the PowerCenter Integration Service Process.	244

Environment Variables.	244
Configuration for the PowerCenter Integration Service Grid.	246
Creating a Grid.	247
Configuring the PowerCenter Integration Service to Run on a Grid.	248
Configuring the PowerCenter Integration Service Processes.	248
Resources.	248
Editing and Deleting a Grid.	251
Troubleshooting a Grid.	251
Load Balancer for the PowerCenter Integration Service	251
Configuring the Dispatch Mode.	252
Service Levels.	254
Configuring Resources.	255
Calculating the CPU Profile.	255
Defining Resource Provision Thresholds.	255
Chapter 11: PowerCenter Integration Service Architecture.....	257
PowerCenter Integration Service Architecture Overview.	257
PowerCenter Integration Service Connectivity.	258
PowerCenter Integration Service Process.	258
Load Balancer.	260
Dispatch Process.	261
Resources.	261
Resource Provision Thresholds.	262
Dispatch Mode.	262
Service Levels.	263
Data Transformation Manager (DTM) Process.	263
Processing Threads.	264
Thread Types.	265
Pipeline Partitioning.	267
DTM Processing.	267
Reading Source Data.	267
Blocking Data.	268
Block Processing.	268
Grids.	269
Workflow on a Grid.	269
Session on a Grid.	270
System Resources.	271
CPU Usage.	271
DTM Buffer Memory.	271
Cache Memory.	271
Code Pages and Data Movement Modes.	272
ASCII Data Movement Mode.	272
Unicode Data Movement Mode.	272

Output Files and Caches.	273
Workflow Log.	274
Session Log.	274
Session Details.	274
Performance Detail File.	274
Reject Files.	275
Row Error Logs.	275
Recovery Tables Files.	275
Control File.	275
Email.	275
Indicator File.	276
Output File.	276
Cache Files.	276

Chapter 12: High Availability for the PowerCenter Integration Service..... 278

High Availability for the PowerCenter Integration Service Overview.	278
Resilience.	278
PowerCenter Integration Service Client Resilience.	279
External Component Resilience.	279
Restart and Failover.	280
Running on a Single Node.	280
Running on a Primary Node.	281
Running on a Grid.	282
Recovery.	282
Stopped, Aborted, or Terminated Workflows.	283
Running Workflows.	283
Suspended Workflows.	283
PowerCenter Integration Service Failover and Recovery Configuration.	283

Chapter 13: PowerCenter Repository Service..... 285

PowerCenter Repository Service Overview.	285
Creating a Database for the PowerCenter Repository.	286
Creating the PowerCenter Repository Service.	286
Before You Begin.	286
Creating a PowerCenter Repository Service.	286
Database Connect Strings.	288
PowerCenter Repository Service Properties.	289
Node Assignments.	289
General Properties.	289
Repository Properties.	290
Database Properties.	290
Advanced Properties.	292
Metadata Manager Service Properties.	293

Custom Properties for the PowerCenter Repository Service.	294
PowerCenter Repository Service Process Properties.	294
Custom Properties for the PowerCenter Repository Service Process.	294
Environment Variables.	294
High Availability for the PowerCenter Repository Service.	295
Resilience.	295
Restart and Failover.	296
Recovery.	296
Chapter 14: PowerCenter Repository Management.	297
PowerCenter Repository Management Overview.	297
PowerCenter Repository Service and Service Processes.	298
Enabling and Disabling a PowerCenter Repository Service.	298
Enabling and Disabling PowerCenter Repository Service Processes.	299
Operating Mode.	300
Running a PowerCenter Repository Service in Exclusive Mode.	300
Running a PowerCenter Repository Service in Normal Mode.	301
PowerCenter Repository Content.	301
Creating PowerCenter Repository Content.	301
Deleting PowerCenter Repository Content.	302
Upgrading PowerCenter Repository Content.	302
Enabling Version Control.	303
Managing a Repository Domain.	303
Prerequisites for a PowerCenter Repository Domain.	303
Building a PowerCenter Repository Domain.	304
Promoting a Local Repository to a Global Repository.	304
Registering a Local Repository.	305
Viewing Registered Local and Global Repositories.	306
Moving Local and Global Repositories.	307
Managing User Connections and Locks.	307
Viewing Locks.	307
Viewing User Connections.	308
Closing User Connections and Releasing Locks.	309
Sending Repository Notifications.	309
Backing Up and Restoring the PowerCenter Repository.	310
Backing Up a PowerCenter Repository.	310
Viewing a List of Backup Files.	311
Restoring a PowerCenter Repository.	311
Copying Content from Another Repository.	312
Repository Plug-in Registration.	313
Registering a Repository Plug-in.	313
Unregistering a Repository Plug-in.	313
Audit Trails.	314

Repository Performance Tuning.	314
Repository Statistics.	314
Repository Copy, Back Up, and Restore Processes.	314
Chapter 15: PowerExchange Listener Service.....	316
PowerExchange Listener Service Overview.	316
DBMOVER Statements for the Listener Service.	317
Creating a Listener Service.	318
Listener Service Properties.	318
PowerExchange Listener Service General Properties.	319
PowerExchange Listener Service Configuration Properties.	320
Environment Variables for the Listener Service Process.	320
Editing Listener Service Properties.	320
Editing Listener Service General Properties.	321
Editing Listener Service Configuration Properties.	321
Enabling, Disabling, and Restarting the Listener Service.	321
Enabling the Listener Service.	321
Disabling the Listener Service.	321
Restarting the Listener Service.	322
Listener Service Logs.	322
Listener Service Restart and Failover.	322
Chapter 16: PowerExchange Logger Service.....	323
PowerExchange Logger Service Overview.	323
Configuration Statements for the Logger Service.	324
Creating a Logger Service.	324
Properties of the PowerExchange Logger Service.	325
PowerExchange Logger Service General Properties.	325
PowerExchange Logger Service Configuration Properties.	325
Logger Service Management.	327
Configuring Logger Service General Properties.	327
Configuring Logger Service Configuration Properties.	328
Configuring the Logger Service Process Properties.	328
Enabling, Disabling, and Restarting the Logger Service.	328
Enabling the Logger Service.	328
Disabling the Logger Service.	328
Restarting the Logger Service.	329
Logger Service Logs.	329
Logger Service Restart and Failover.	329
Chapter 17: SAP BW Service.....	330
SAP BW Service Overview.	330
Creating the SAP BW Service.	331

Enabling and Disabling the SAP BW Service.	333
Enabling the SAP BW Service.	334
Disabling the SAP BW Service.	334
Configuring the SAP BW Service Properties.	334
General Properties.	334
SAP BW Service Properties.	335
Configuring the Associated Integration Service.	336
Configuring the SAP BW Service Processes.	336
Load Balancing for the SAP BW System and the SAP BW Service.	337
Viewing Log Events.	337

Chapter 18: Search Service..... 338

Search Service Overview.	338
Search Service Architecture.	339
Search Index.	339
Extraction Interval	340
Search Request Process.	340
Search Service Properties.	341
General Properties for the Search Service.	341
Logging Options for the Search Service.	341
Search Options for the Search Service.	342
Custom Properties for the Search Service.	342
Search Service Process Properties.	343
Advanced Properties of the Search Service Process.	343
Environment Variables for the Search Service Process.	343
Custom Properties for the Search Service Process.	344
Creating a Search Service.	344
Enabling the Search Service.	344
Recycling and Disabling the Search Service.	345

Chapter 19: System Services..... 346

System Services Overview.	346
Email Service.	347
Before You Enable the Email Service.	347
Email Service Properties.	347
Email Service Process Properties.	349
Enabling, Disabling, and Recycling the Email Service.	349
Resource Manager Service.	350
Resource Manager Service Architecture.	350
Before You Enable the Resource Manager Service.	351
Resource Manager Service Properties.	351
Resource Manager Service Process Properties.	352
Enabling, Disabling, and Recycling the Resource Manager Service.	352

Scheduler Service.	353
Before You Enable the Scheduler Service.	353
Scheduler Service Properties.	353
Scheduler Service Process Properties.	355
Enabling, Disabling, and Recycling the Scheduler Service.	357
Chapter 20: Test Data Manager Service.	359
Test Data Manager Service Overview	359
Test Data Manager Service Dependencies.	360
Test Data Manager Service Properties.	360
General Properties.	361
Service Properties.	361
TDM Repository Configuration Properties.	362
TDM Server Configuration Properties.	363
Advanced Properties.	364
Database Connection Strings.	364
Configuring the Test Data Manager Service.	364
Creating the Test Data Manager Service.	365
Enabling and Disabling the Test Data Manager Service.	365
Editing the Test Data Manager Service.	366
Create or Upgrade TDM Repository Content.	366
Assigning the Test Data Manager Service to a Different Node.	366
Assigning a New License to the Test Data Manager Service.	366
Deleting the Test Data Manager Service.	367
Chapter 21: Test Data Warehouse Service.	368
Test Data Warehouse Service Overview.	368
Test Data Warehouse Services Dependencies.	368
Test Data Warehouse Service Properties.	369
General Properties.	369
Test Data Warehouse Repository Configuration Properties.	370
Test Data Warehouse Properties.	370
Test Data Warehouse Server Configuration Properties.	371
Creating the Test Data Warehouse Service.	371
Process Properties for the Test Data Warehouse Service.	372
Chapter 22: Web Services Hub.	373
Web Services Hub Overview.	373
Creating a Web Services Hub.	374
Enabling and Disabling the Web Services Hub.	375
Web Services Hub Properties.	376
General Properties.	377
Service Properties.	377

Advanced Properties.	378
Custom Properties for the Web Services Hub.	380
Configuring the Associated Repository.	380
Adding an Associated Repository.	380
Editing an Associated Repository.	381
Chapter 23: Application Service Upgrade.	382
Application Service Upgrade Overview.	382
Privileges to Upgrade Services.	382
Service Upgrade from Previous Versions.	383
Running the Service Upgrade Wizard.	383
Verify the Model Repository Service Upgrade.	384
Object Dependency Graph.	384
Maximum Heap Size.	385
Appendix A: Application Service Databases.	386
Application Service Databases Overview.	386
Set Up Database User Accounts.	387
Data Object Cache Database Requirements.	387
IBM DB2 Database Requirements.	387
Microsoft SQL Server Database Requirements.	387
Oracle Database Requirements.	388
Exception Management Audit Database Requirements.	388
IBM DB2 Database Requirements.	388
Microsoft SQL Server Database Requirements.	389
Oracle Database Requirements.	389
Metadata Manager Repository Database Requirements.	389
IBM DB2 Database Requirements.	390
Microsoft SQL Server Database Requirements.	391
Oracle Database Requirements.	392
Model Repository Database Requirements.	393
IBM DB2 Database Requirements.	393
Microsoft SQL Server Database Requirements.	394
Oracle Database Requirements.	394
PowerCenter Repository Database Requirements.	394
IBM DB2 Database Requirements.	395
Microsoft SQL Server Database Requirements.	395
Oracle Database Requirements.	395
Sybase ASE Database Requirements.	396
Profiling Warehouse Requirements.	396
IBM DB2 Database Requirements.	396
Microsoft SQL Server Database Requirements.	397
Oracle Database Requirements.	397

Reference Data Warehouse Requirements.	398
IBM DB2 Database Requirements.	398
Microsoft SQL Server Database Requirements.	398
Oracle Database Requirements.	399
Workflow Database Requirements.	399
IBM DB2 Database Requirements	399
Microsoft SQL Server Database Requirements.	400
Oracle Database Requirements.	400
Configure Native Connectivity on Service Machines.	401
Install Database Client Software.	402
Configure Database Client Environment Variables on UNIX.	402

Appendix B: Connecting to Databases from Windows. 404

Connecting to Databases from Windows Overview.	404
Connecting to an IBM DB2 Universal Database from Windows.	405
Configuring Native Connectivity.	405
Connecting to an Informix Database from Windows.	405
Configuring ODBC Connectivity.	406
Connecting to Microsoft Access and Microsoft Excel from Windows.	406
Configuring ODBC Connectivity.	406
Connecting to a Microsoft SQL Server Database from Windows.	406
Configuring Native Connectivity.	406
Configuring Custom Properties for Microsoft SQL Server.	407
Connecting to a Netezza Database from Windows.	408
Configuring ODBC Connectivity.	408
Connecting to an Oracle Database from Windows.	408
Configuring Native Connectivity.	409
Connecting to a Sybase ASE Database from Windows.	410
Configuring Native Connectivity.	410
Connecting to a Teradata Database from Windows.	411
Configuring ODBC Connectivity.	411

Appendix C: Connecting to Databases from UNIX. 412

Connecting to Databases from UNIX Overview.	412
Connecting to an IBM DB2 Universal Database from UNIX.	413
Configuring Native Connectivity.	413
Connecting to an Informix Database from UNIX.	415
Configuring ODBC Connectivity.	415
Connecting to Microsoft SQL Server from UNIX.	416
Configuring Native Connectivity.	416
Configuring SSL Authentication through ODBC.	417
Configuring Custom Properties for Microsoft SQL Server.	417
Connecting to a Netezza Database from UNIX.	418

Configuring ODBC Connectivity.	418
Connecting to an Oracle Database from UNIX.	420
Configuring Native Connectivity.	420
Connecting to a Sybase ASE Database from UNIX.	422
Configuring Native Connectivity.	422
Connecting to a Teradata Database from UNIX.	424
Configuring ODBC Connectivity.	424
Connecting to an ODBC Data Source.	426
Sample odbc.ini File.	428
 Appendix D: Updating the DynamicSections Parameter of a DB2 Database..	435
DynamicSections Parameter Overview.	435
Updating the DynamicSections Parameter.	435
Downloading and Installing the DataDirect Connect for JDBC Utility.	435
Running the Test for JDBC Tool.	436
 Index.	437

Preface

The *Informatica Application Service Guide* is written for Informatica users who need to configure application services. The *Informatica Application Service Guide* assumes you have basic working knowledge of Informatica and details of the environment in which the application services run.

Informatica Resources

Informatica Network

Informatica Network hosts Informatica Global Customer Support, the Informatica Knowledge Base, and other product resources. To access Informatica Network, visit <https://network.informatica.com>.

As a member, you can:

- Access all of your Informatica resources in one place.
- Search the Knowledge Base for product resources, including documentation, FAQs, and best practices.
- View product availability information.
- Review your support cases.
- Find your local Informatica User Group Network and collaborate with your peers.

Informatica Knowledge Base

Use the Informatica Knowledge Base to search Informatica Network for product resources such as documentation, how-to articles, best practices, and PAMs.

To access the Knowledge Base, visit <https://kb.informatica.com>. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at KB_Feedback@informatica.com.

Informatica Documentation

To get the latest documentation for your product, browse the Informatica Knowledge Base at https://kb.informatica.com/_layouts/ProductDocumentation/Page/ProductDocumentSearch.aspx.

If you have questions, comments, or ideas about this documentation, contact the Informatica Documentation team through email at infa_documentation@informatica.com.

Informatica Product Availability Matrixes

Product Availability Matrixes (PAMs) indicate the versions of operating systems, databases, and other types of data sources and targets that a product release supports. If you are an Informatica Network member, you can access PAMs at

<https://network.informatica.com/community/informatica-network/product-availability-matrices>.

Informatica Marketplace

The Informatica Marketplace is a forum where you can find solutions that augment, extend, or enhance your Informatica implementations. By leveraging any of the hundreds of solutions from Informatica developers and partners, you can improve your productivity and speed up time to implementation on your projects. You can access Informatica Marketplace at <https://marketplace.informatica.com>.

Informatica Global Customer Support

You can contact a Global Support Center by telephone or through Online Support on Informatica Network.

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:

<http://www.informatica.com/us/services-and-training/support-services/global-support-centers>.

If you are an Informatica Network member, you can use Online Support at <http://network.informatica.com>.

CHAPTER 1

Analyst Service

This chapter includes the following topics:

- [Analyst Service Overview, 22](#)
- [Analyst Service Architecture, 23](#)
- [Configuration Prerequisites, 24](#)
- [Recycle and Disable the Analyst Service, 26](#)
- [Properties for the Analyst Service, 26](#)
- [Custom Images in the Analyst Tool, 29](#)
- [Process Properties for the Analyst Service, 30](#)
- [Creating and Configuring the Analyst Service, 32](#)
- [Creating an Analyst Service, 33](#)

Analyst Service Overview

The Analyst Service is an application service that runs the Analyst tool in the Informatica domain. The Analyst Service manages the connections between the service components and the users who log in to the Analyst tool.

The Analyst Service connects to a Data Integration Service that runs profiles, scorecards, and mapping specifications. The Analyst Service also connects to a Data Integration Service that runs workflows.

The Analyst Service connects to the Model Repository Service to identify a Model repository. The Analyst Service connects to a Metadata Manager Service that enables data lineage analysis on scorecards in the Analyst tool. The Analyst Service connects to a Search Service that enables and manages searches in the Analyst tool.

Additionally, the Analyst Service connects to the Analyst tool, a flat file cache directory to store uploaded flat files, and a business glossary export file directory.

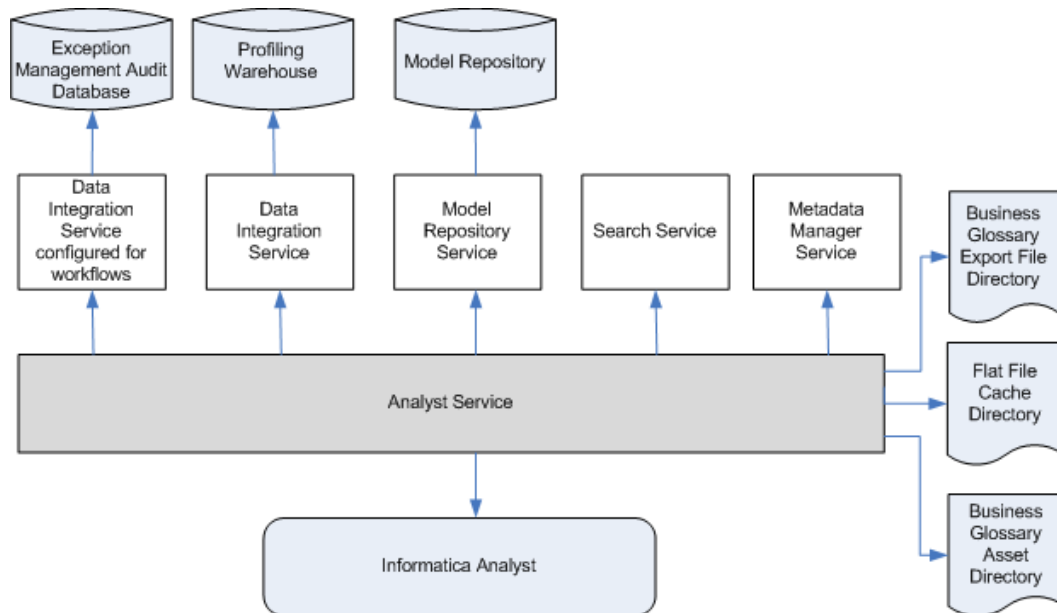
You can use the Administrator tool to create and recycle an Analyst Service in the Informatica domain and to access the Analyst tool. When you recycle the Analyst Service, the Service Manager restarts the Analyst Service.

You can run more than one Analyst Service on the same node. You can associate a Model Repository Service with one Analyst Service. You can associate one Data Integration Service with more than one Analyst Service. The Analyst Service detects the associated Search Service based on the Model Repository Service assigned to the Analyst Service.

Analyst Service Architecture

The Analyst Service connects to application services, databases, and directories.

The following figure shows the Analyst tool components that the Analyst Service connects to in the Informatica domain:



The Analyst Service connects to the following components:

- **Data Integration Services.** The Analyst Service manages the connection to a Data Integration Service that runs profiles, scorecards, and mapping specifications in the Analyst tool. The Analyst Service also manages the connection to a Data Integration Service that runs workflows.
- **Model Repository Service.** The Analyst Service manages the connection to a Model Repository Service for the Analyst tool. The Analyst tool connects to the Model repository database to create, update, and delete projects and objects in the Analyst tool.
- **Search Service.** The Analyst Service manages the connection to the Search Service that enables and manages searches in the Analyst tool. The Analyst Service identifies the associated Search Service based on the Model Repository Service associated with the Analyst Service.
- **Metadata Manager Service.** The Analyst Service manages the connection to a Metadata Manager Service that runs data lineage for scorecards in the Analyst tool.
- **Profiling warehouse database.** The Analyst tool identifies the profiling warehouse database. The Data Integration Service writes profile data and scorecard results to the database.
- **Exception management audit database.** The Analyst Service manages the connection to a database that can store all audit data for the exception management tasks that users work on in the Analyst tool.
- **Flat file cache directory.** The Analyst Service manages the connection to the directory that stores uploaded flat files that you import for reference tables and flat file data sources in the Analyst tool.
- **Business Glossary export file directory.** The Analyst Service manages the connection to the directory that stores the business glossary as a file after you export it from the Analyst tool.
- **Business Glossary asset attachment directory.** The Analyst Service identifies the directory that stores any attachment that an Analyst tool user attaches to a Business Glossary asset.
- **Informatica Analyst.** The Analyst Service defines the URL for the Analyst tool.

Configuration Prerequisites

Before you configure the Analyst Service, you can complete the prerequisite tasks for the service. You can also choose to complete these tasks after you create an Analyst Service.

Perform the following tasks before you configure the Analyst Service:

- Create and enable the associated Data Integration Services, Model Repository Service, and Metadata Manager Service.
- Identify a directory for the flat file cache to upload flat files.
- Identify a directory to export a business glossary.
- Identify a keystore file to configure the Transport Layer Security protocol for the Analyst Service.
- Optionally, create a database to store audit data for the exception management tasks that the Analyst Service identifies.

Services Associated with the Analyst Service

The Analyst Service connects to associated services that you create and enable before you configure the Analyst Service.

The Analyst Service connects to the following associated services:

- **Data Integration Services.** You can associate up to two Data Integration Services with the Analyst Service. Associate a Data Integration Service to run mapping specifications, profiles, and scorecards. Associate a Data Integration Service to run workflows. You can associate the same Data Integration Service to run mapping specifications, profiles, scorecards, and workflows.
- **Model Repository Service.** When you create an Analyst Service, you assign a Model Repository Service to the Analyst Service. You cannot assign the same Model Repository Service to another Analyst Service.
- **Metadata Manager Service.** You can associate a Metadata Manager Service with the Analyst Service to perform data lineage analysis on scorecards.
- **Search Service.** The Analyst Service determines the associated Search Service based on the Model Repository Service associated with the Analyst Service. If you modify the Analyst Service, you must recycle the Search Service.

Flat File Cache Directory

Create a directory for the flat file cache where the Analyst tool stores uploaded flat files. The Data Integration Service must also be able to access this directory.

If the Analyst Service and the Data Integration Service run on different nodes, configure the flat file directory to use a shared directory. If the Data Integration Service runs on primary and back-up nodes or on a grid, each Data Integration Service process must be able to access the files in the shared directory.

For example, you can create a directory named "flatfilecache" in the following mapped drive that all Analyst Service and Data Integration Service processes can access:

```
F:\shared\<Informatica installation directory>\server
```

If the Analyst Service connects to a Data Integration Service that uses operating system profiles, the operating system user specified in the operating system profile must have access to the flat file cache directory. If the Analyst Service and the Data Integration Service run on different nodes, the operating system profiles must be configured for both nodes. The flat file cache directory specified in the operating system profile must be accessible from both nodes.

When you import a reference table or flat file source, the Analyst tool uses the files from this directory to create a reference table or flat file data object.

Export File Directory

Create a directory to store the temporary business glossary files that the business glossary export process creates.

For example, you can create a directory named "exportfiledirectory" in the following location:

```
<InformaticaInstallationDir>\server
```

Attachments Directory

Create a directory to store attachments that the Business Glossary data steward adds to Glossary assets.

For example, you can create a directory named "BGattachmentsdirectory" in the following location:

```
<InformaticaInstallationDir>\server
```

Keystore File

A keystore file contains the keys and certificates required if you enable secure communication and use the HTTPS protocol for the Analyst Service.

You can create the keystore file when you install the Informatica services or you can create a keystore file with keytool. keytool is a utility that generates and stores private or public key pairs and associated certificates in a file called a "keystore." When you generate a public or private key pair, keytool wraps the public key into a self-signed certificate. You can use the self-signed certificate or use a certificate signed by a certificate authority.

Note: You must use a certified keystore file. If you do not use a certified keystore file, security warnings and error messages for the browser appear when you access the Analyst tool.

Exception Management Audit Database

Configure the Analyst Service to specify a single audit database for exception management tasks.

An exception management task is an instance of a Human task. When you run a workflow that contains a Human task, the Data Integration Service that the Analyst Service specifies creates instances of the Human task. Analyst tool users can update the data in the task instances. The exception management audit database stores a record of the work that the Analyst tool users perform.

To configure the audit database, identify a database connection and a schema for the audit tables. Set the options on the Human task properties of the Analyst Service in the Administrator tool. Or, run the infacmd as updateServiceOptions command.

If you run infacmd as updateServiceOptions, set the following options:

- HumanTaskDataIntegrationService.exceptionDbName
- HumanTaskDataIntegrationService.exceptionSchemaName

After you set the connection name and schema, create the audit database contents. To create the database contents, use the **Actions** menu options for the Analyst Service in the Administrator tool. Or, run the infacmd as createExceptionAuditTables command.

Note: You can also use the **Actions** menu options to delete the database contents. Or, you can run the infacmd as deleteExceptionAuditTables command.

If you specify a connection and schema and you do not create the database contents, Analyst tool users cannot open the task instances.

If you do not specify a connection and schema, the Analyst Service creates audit tables for each task instance in the database that stores the task instance data. If the Human task data resides in multiple databases, the Analyst Service writes the audit data to the respective databases.

Recycle and Disable the Analyst Service

Disable an Analyst Service to perform maintenance or temporarily restrict users from accessing the Analyst tool. Recycle an Analyst Service to make the Analyst tool available to users.

Use the Administrator tool to recycle and disable the Analyst Service. When you disable the Analyst Service, you also stop the Analyst tool. When you recycle the Analyst Service, you stop and start the service to make the Analyst tool available again.

In the Navigator, select the Analyst Service and click the Disable button to stop the service. Click the Recycle button to start the service.

When you disable the Analyst Service, you must choose the mode to disable it in. You can choose one of the following options:

- Complete. Allows the jobs to run to completion before disabling the service.
- Abort. Tries to stop all jobs before aborting them and disabling the service.
- Stop. Stops all jobs and then disables the service.

Note: The Model Repository Service and the Data Integration Service must be running before you recycle the Analyst Service.

Properties for the Analyst Service

After you create an Analyst Service, you can configure the Analyst Service properties. You can configure Analyst Service properties on the Properties tab in the Administrator tool.

For each service properties section, click **Edit** to modify the service properties.

You can configure the following types of Analyst Service properties:

- General Properties
- Model Repository Service Properties
- Logging Options
- Human Task Properties
- Run-Time Properties
- Metadata Manager Properties
- Business Glossary Export Properties
- Custom Properties

If you update any of the properties, recycle the Analyst Service for the modifications to take effect.

General Properties for the Analyst Service

General properties for the Analyst Service include the name and description of the Analyst Service, and the node in the Informatica domain that the Analyst Service runs on. You can configure these properties when you create the Analyst Service.

You can configure the following general properties for the service:

Name

Name of the service. The name is not case sensitive and must be unique within the domain. It cannot exceed 128 characters or begin with @. It also cannot contain spaces or the following special characters:

`` ~ % ^ * + = { } \ ; : ' " / ? . , < > | ! () []`

You cannot change the name of the service after you create it.

Description

Description of the service. The description cannot exceed 765 characters.

Node

Node on which the service runs. If you change the node, you must recycle the Analyst Service.

License

License object that allows use of the service.

Model Repository Service Properties

Model Repository Service Properties include properties for the Model Repository Service that is associated with the Analyst Service.

The Analyst Service has the following Model Repository Service properties:

Model Repository Service

Model Repository Service associated with the Analyst Service. The Analyst Service manages the connections to the Model Repository Service for Informatica Analyst. You must recycle the Analyst Service if you associate another Model Repository Service with the Analyst Service. If you use advanced approval workflow to publish Glossary assets, you must configure the Model Repository Service properties.

Username

User name of an administrator user in the Informatica domain.

Password

Password of the administrator user in the Informatica domain.

Security Domain

LDAP security domain for the user who manages the Model Repository Service. The security domain field does not appear for users with Native authentication.

Logging Options

Logging options include properties for the severity level for Service logs. Configure the Log Level property to set the logging level. The following values are valid:

- Fatal. Writes FATAL messages to the log. FATAL messages include nonrecoverable system failures that cause the service to shut down or become unavailable.
- Error. Writes FATAL and ERROR code messages to the log. ERROR messages include connection failures, failures to save or retrieve metadata, service errors.
- Warning. Writes FATAL, WARNING, and ERROR messages to the log. WARNING errors include recoverable system failures or warnings.
- Info. Writes FATAL, INFO, WARNING, and ERROR messages to the log. INFO messages include system and service change messages.
- Trace. Write FATAL, TRACE, INFO, WARNING, and ERROR code messages to the log. TRACE messages log user request failures.
- Debug. Write FATAL, DEBUG, TRACE, INFO, WARNING, and ERROR messages to the log. DEBUG messages are user request logs.

The default value is Info.

Human Task Properties

The Human task properties include options to select a Data Integration Service for workflows and to identify an audit trail database for Human task instances.

The Analyst Service has the following Human task properties:

Data Integration Service

Data Integration Service that runs a workflow that creates Human task instances. When a user logs in to the Analyst Service URL, the user can work on any Human task instance that the workflow assigns to them. If the Data Integration Service that you select is not configured to run workflows, select a different Data Integration Service.

Exception Audit Database Connection

Connection name for the database that stores audit trail data for Human task instances.

When a user logs in to the Analyst Service URL and updates a Human task instance, the database stores the update. The database stores audit trail data for all Human task instances that users work on at the current Analyst Service URL.

Exception Audit Database Schema

Name of the schema that defines the audit trail tables in the exception audit database.

Note: If you specify a database connection and a schema for exception audit data, the Analyst Service stores all exception audit data in a single location. If you do not specify a connection and a schema, the Analyst Service creates audit trail tables for a Human task instance in the database that contains the task instance data.

Run-time Properties

Run-time properties include the Data Integration Service associated with the Analyst Service and the flat file cache directory.

The Analyst Service has the following run-time properties:

Data Integration Service

Data Integration Service that enables users to perform data preview, mapping specification, and profile tasks in the Analyst tool. The Analyst Service manages the connection to the Data Integration Service. You must recycle the Analyst Service if you associate another Data Integration Service with the Analyst Service.

Flat File Cache Directory

Directory of the flat file cache where the Analyst tool stores uploaded flat files. The Analyst Service and the Data Integration Service must be able to access this directory. If the Analyst Service and the Data Integration Service run on different nodes, configure the flat file directory to use a shared directory. If the Data Integration Service runs on primary and back-up nodes or on a grid, each Data Integration Service process must be able to access the files in the shared directory.

When you import a reference table or flat file source, the Analyst tool uses the files from this directory to create a reference table or flat file data object. Restart the Analyst Service if you change the flat file location.

Metadata Manager Service Properties

The Metadata Manager Service Properties include the option to select a Metadata Manager Service by name.

Business Glossary Properties

You can configure the following Business Glossary properties:

- Temporary directory to store the Microsoft Excel export file before the Analyst tool makes it available for download via the browser.
- Directory where attachments added to Glossary assets are stored.

Custom Properties for the Analyst Service

Configure custom properties that are unique to specific environments.

You might need to apply custom properties in special cases. When you define a custom property, enter the property name and an initial value. Define custom properties only at the request of Informatica Global Customer Support.

Custom Images in the Analyst Tool

The Analyst tool randomly cycles through a standard set of images in the login page. Each time you open the Analyst tool login page, a different image appears in the background. You can configure the Analyst Service to display custom images instead of the standard set of images.

Configure the **JVM Command Line Options** in the **Advanced Properties** dialog box to add custom images to the Analyst tool. Configure `DBackgroundImageDirectory` to the path where you store the images. The custom images must be in the .png file format with a resolution of 1100 x 745.

Process Properties for the Analyst Service

The Analyst Service runs the Analyst Service process on a node. When you select the Analyst Service in the Administrator tool, you can view the service processes for the Analyst Service on the **Processes** tab. You can view the node properties for the service process in the service panel. You can view the service process properties in the Service Process Properties panel.

Note: You must select the node to view the service process properties in the Service Process Properties panel.

You can configure the following types of Analyst Service process properties:

- Analyst Security Options
- Advanced Properties
- Custom Properties
- Environment Variables

If you update any of the process properties, restart the Analyst Service for the modifications to take effect.

Node Properties for the Analyst Service Process

The Analyst Service process has the following node properties:

Node

Node that the service process runs on.

Node Status

Status of the node. Status can be enabled or disabled.

Process Configuration

Status of the process configured to run on the node.

Process State

State of the service process running on the node. The state can be enabled or disabled.

Analyst Security Options for the Analyst Service Process

The Analyst Service Options include security properties for the Analyst Service process.

The Analyst Service process has the following security properties:

HTTP Port

HTTP port number on which the Analyst tool runs. Use a port number that is different from the HTTP port number for the Data Integration Service. Default is 8085. You must recycle the service if you change the HTTP port number.

Enable Secure Communication

Set up secure communication between the Analyst tool and the Analyst Service.

HTTPS Port

Port number to use for a secure connection to the Informatica Administrator service. Use a different port number than the HTTP port number. You must recycle the service if you change the HTTPS port number.

Keystore File

Path and file name of the keystore file to use for the HTTPS connection to the Informatica Administrator service.

Keystore Password

Password for the keystore file.

SSL Protocol

Informatica recommends that you leave this field blank. The version of TLS enabled depends on the value. A blank field enables the highest version of TLS available. If you enter a value, earlier versions of TLS might be enabled. The behavior is based on the Java version for your environment.

For more information, see the documentation for your Java version.

Advanced Properties for the Analyst Service Process

Advanced properties include properties for the maximum heap size and the Java Virtual Manager (JVM) memory settings.

The Analyst Service process has the following advanced properties:

Maximum Heap Size

Amount of RAM allocated to the Java Virtual Machine (JVM) that runs the Analyst Service. Use this property to increase the performance. Append one of the following letters to the value to specify the units:

- m for megabytes.
- g for gigabytes.

Default is 768 megabytes. Specify 2 gigabytes if you run the Analyst Service on a 64-bit machine.

JVM Command Line Options

Java Virtual Machine (JVM) command line options to run Java-based programs. When you configure the JVM options, you must set the Java SDK classpath, Java SDK minimum memory, and Java SDK maximum memory properties.

To enable the Analyst Service to add custom images to the Analyst tool, add the following property to the JVM Command Line Options:

```
DBackgroundImageDirectory=<directory path>
```

To enable the Analyst Service to communicate with a Hadoop cluster on a particular Hadoop distribution, add the following property to the JVM Command Line Options:

```
-DINFA_HADOOP_DIST_DIR=<Hadoop installation directory>\<HadoopDistributionName>
```

For example, to enable the Analyst Service to communicate with a Hadoop cluster on Cloudera CDH 5.2, add the following property:

```
-DINFA_HADOOP_DIST_DIR=...\services\shared\hadoop\cloudera_cdh5u2
```

Custom Properties for the Analyst Service Process

Configure custom properties that are unique to specific environments.

You might need to apply custom properties in special cases. When you define a custom property, enter the property name and an initial value. Define custom properties only at the request of Informatica Global Customer Support.

Environment Variables for the Analyst Service Process

You can edit environment variables for the Analyst Service process.

The Analyst Service process has the following property for environment variables:

Environment Variables

Environment variables defined for the Analyst Service process.

Environment Variables for MapR

If you use the Analyst tool to run profiles against Hive data objects, configure properties on the Analyst Service to enable communication between the Analyst tool and the cluster, including testing of the Hive connection.

The following table describes the MapR properties:

Property	Value
JAVA_OPTS	<code>-Dhadoop.login=hybrid -Dhttps.protocols=TLSv1.2</code> Applicable to the MapR Kerberos cluster.
MAPR_HOME	Hadoop distribution directory location on the machine that runs the Data Integration Service. For example, <code><Informatica installation directory>/services/shared/hadoop/mapr_5.2.0_TK</code>
MAPR_TICKETFILE_LOCATION	Directory where the MapR Ticket file is stored on the machine that runs the Analyst Service. For example, <code>/export/home/username1/Keytabs_and_krb5conf/Tickets/project1/maprticket_30103</code>
LD_LIBRARY_PATH	The location of Hadoop libraries. For example, <code><Informatica installation directory>/java/jre/lib:<Informatica installation directory>/services/shared/bin:<Informatica installation directory>/server/bin:<Informatica installation directory>/services/shared/hadoop/<MapR location>/lib/native/Linux-amd64-64</code>

Changes take effect when you restart the Analyst Service.

Creating and Configuring the Analyst Service

Use the Administrator tool to create and configure the Analyst Service. After you create the Analyst Service, you can configure the service properties and service process properties. You can enable the Analyst Service to make the Analyst tool accessible to users.

1. Complete the prerequisite tasks for configuring the Analyst Service.

2. Create the Analyst Service.
3. Configure the Analyst Service properties.
4. Configure the Analyst Service process properties.
5. Recycle the Analyst Service.

Creating an Analyst Service

Create an Analyst Service to manage the Informatica Analyst application and to grant users access to Informatica Analyst.

Note: The Analyst service has the same privileges as the user account that creates it. Ensure that the user account does not have privileges to read or modify sensitive files on the system.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. On the Domain Navigator Actions menu, click **New > Analyst Service**.
The **New Analyst Service** window appears.
3. Enter the general properties for the service.
Optionally, click Browse in the **Location** field to enter the location for the domain and folder where you want to create the service. Optionally, click Create Folder to create another folder.
4. Enter the Analyst Security Options for the Analyst Service.
5. Select **Enable Service** to enable the service after you create it.
6. Click **Next**.
7. Enter the Model Repository Service properties.
8. Optionally, enter the Human task properties.
9. Click **Next**.
10. Enter the run-time properties.
11. Optionally, enter the Metadata Manager properties and the Catalog Service properties.
12. Optionally, enter the business glossary export property.
13. Click **Finish**.

If you did not choose to enable the service earlier, you must recycle the service to start it.

CHAPTER 2

Content Management Service

This chapter includes the following topics:

- [Content Management Service Overview, 34](#)
- [Master Content Management Service , 35](#)
- [Content Management Service Architecture, 35](#)
- [Probabilistic Models and Classifier Models, 36](#)
- [Reference Data Warehouse, 37](#)
- [Recycling and Disabling the Content Management Service, 38](#)
- [Content Management Service Properties, 39](#)
- [Content Management Service Process Properties, 42](#)
- [Creating a Content Management Service, 47](#)

Content Management Service Overview

The Content Management Service is an application service that manages reference data. It provides reference data information to the Data Integration Service and to the Developer and Analyst tools. A master Content Management Service maintains probabilistic model and classifier model data files across the domain.

The Content Management Service manages the following types of reference data:

Address reference data

You use address reference data when you want to validate the postal accuracy of an address or fix errors in an address. Use the Address Validator transformation to perform address validation.

Identity populations

You use identity population data when you want to perform duplicate analysis on identity data. An identity is a set of values within a record that collectively identify a person or business. Use a Match transformation or Comparison transformation to perform identity duplicate analysis.

Probabilistic models and classifier models

You use probabilistic or classifier model data when you want to identify the type of information that a string contains. Use a probabilistic model in a Parser or Labeler transformation. Use a classifier model in a Classifier transformation. Probabilistic models and classifier models use probabilistic logic to identify or infer the type of information in the string. Use a Classifier transformation when each input string contains a significant amount of data.

Reference tables

You use reference tables to verify the accuracy or structure of input data values in data quality transformations.

The Content Management Service also compiles rule specifications into mapplets.

Use the Administrator tool to administer the Content Management Service. Recycle the Content Management Service to start it.

Master Content Management Service

When you create multiple Content Management Services on a domain and associate the services with a Model repository, one service operates as the master Content Management Service. The first Content Management Service you create on a domain is the master Content Management Service.

Use the **Master CMS** property to identify the master Content Management Service. When you create the first Content Management Service on a domain, the property is set to True. When you create additional Content Management Services on a domain, the property is set to False.

You cannot edit the **Master CMS** property in the Administrator tool. Use the `infacmd cms UpdateServiceOptions` command to change the master Content Management Service.

Content Management Service Architecture

The Developer tool and the Analyst tool interact with a Content Management Service to retrieve configuration information for reference data and to compile rule specifications.

You associate a Content Management Service with a Data Integration Service and Model Repository Service in a domain. If the Data Integration Service runs a mapping that reads reference data, you must create the Data Integration Service and Content Management Service on the same node. You associate a Data Integration Service with a single Content Management Service.

The Content Management Service must be available when you use the following resources:

Address reference data

The Content Management Service manages configuration information for address reference data. The Data Integration Service maintains a copy of the configuration information. The Data Integration Service applies the configuration information when it runs a mapping that reads the address reference data.

Identity population files

The Content Management Service manages the list of the population files on the node. When you configure a Match transformation or a Comparison transformation, you select a population file from the current list. The Data Integration Service applies the population configuration when it runs a mapping that reads the population files.

Probabilistic model files and classifier model files

The Content Management Service stores the locations of any probabilistic model file and classifier model file on the node. The Content Management Service also manages the compilation status of each model.

Update a probabilistic model or a classifier model on the master Content Management Service machine. When you update a model, the master Content Management Service updates the corresponding model file on any node that you associate with the Model repository.

Note: If you add a node to a domain and you create a Content Management Service on the node, run the `infacmd cms ResyncData` command. The command updates the node with probabilistic model files or classifier model files from the master Content Management Service machine.

Reference tables

The Content Management Service identifies the database that stores data values for the reference table objects in the associated Model repository.

Rule specifications

The Content Management Service manages the compilation of rule specifications into mapplets. When you compile a rule specification in the Analyst tool, the Analyst Service selects a Content Management Service to generate the mapplet. The Analyst tool uses the Model Repository Service configuration to select the Content Management Service.

Content Management Service and Operating System Profiles

You might configure the Data Integration Service to run mappings and other objects through an operating system profile. You can assign the operating system profile to an Analyst tool user or a Developer tool user.

If you do not assign the operating system profile to a user, the user might not be able to perform the full range of operations that the Content Management Service enables. To enable Analyst tool users and Developer tool users to perform all operations with the Content Management Service, assign the default operating system profile for the associated Data Integration Service to the users.

Probabilistic Models and Classifier Models

The Model Repository Service reads probabilistic model and classifier model file data from the machine that hosts the master Content Management Service in the domain. When you compile a probabilistic model or classifier model in the Developer tool, you update the model files on the master Content Management Service machine.

If a node in the domain runs a Content Management Service, the node stores local copies of the probabilistic model and classifier model files. You specify the local path to the probabilistic and classifier model files in the **NLP Options** property on the Content Management Service. The master Content Management Service synchronizes the probabilistic model and classifier model files on the domain nodes with the master Content Management Service files every 10 minutes.

To synchronize a Content Management Service machine with the current files from the master Content Management Service machine, run the following command:

```
infacmd cms ResyncData
```

The command updates the machine that hosts the new service with the probabilistic model or classifier model files from the master Content Management Service machine. When you add a Content Management Service to a domain that includes a master Content Management Service, run the `ResyncData` command.

You specify a single type of model file when you run the command. To synchronize probabilistic model files and classifier model files, run the command once for each type of model file.

Synchronization Operations

The master Content Management Service stores a list of the Content Management Services in the domain. When the master Content Management Service synchronizes with the domain services, the master Content Management Service copies the current model files sequentially to each domain node. If a node is unavailable, the master Content Management Service moves the node to the end of the list and synchronizes with the next node on the list. After the synchronization operation copies the files to all available Content Management Service machines, the operation ends.

To verify that a synchronization operation succeeded on a node, browse the directory structure on the node and find the probabilistic or classifier model files. Compare the files with the files on the master Content Management Service machine.

Informatica uses the following directory paths as the default locations for the files:

```
[Informatica_install_directory]/tomcat/bin/ner  
[Informatica_install_directory]/tomcat/bin/classifier
```

The file names have the following extensions:

Probabilistic model files: .ner

Classifier model files: .classifier

Note: The time required to synchronize the model files depends on the number of files on the master Content Management Service machine. The ResyncData command copies model files in batches of 15 files at a time.

Reference Data Warehouse

The reference data warehouse stores data values for the reference table objects you define in a Model repository.

When you add data to a reference table, the Content Management Service writes the data values to a table in the reference data warehouse. For example, when you create a reference table from a flat file, the Content Management Service uses the file structure to define the object metadata in the Model repository. The Content Management Service writes the file data to a table in the reference data warehouse.

The **Reference Data Location** option on the Content Management identifies the reference data warehouse. To update the data warehouse connection, configure this option.

When you specify a reference data warehouse, verify that the database you select stores data for the Model repository only.

Orphaned Reference Data

When you delete a reference table object from the Model repository, the table data remains in the reference data warehouse.

Use the **Purge Orphaned Tables** option on the Content Management Service to delete unused reference tables. The option identifies the tables that store data for reference table objects in the Model repository and deletes all other reference tables from the warehouse. The purge option removes obsolete reference tables and creates additional space in the warehouse.

Before you purge the unused tables, verify the following prerequisites:

- You have the Manage Service privilege on the domain.

- The user name that the Content Management Service uses to communicate with the Model repository has the Administrator role on the associated Model Repository Service.
- All Data Integration Services associated with the Model repository are available.
- There are no data operations in progress on the reference data warehouse.
- The reference data warehouse stores data for the reference table objects in a single Model repository.

Note: The purge operation reads the Model repository that the current Content Management Service identifies, and it deletes any reference table that the Model repository does not use. If the reference data warehouse stores reference data for any other Model repository, the purge operation deletes all tables that belong to the other repository. To prevent accidental data loss, the purge operation does not delete tables if the Model repository does not contain a reference table object.

Deleting Orphaned Tables

To delete unused reference tables from the reference data warehouse, purge orphaned tables.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the master Content Management Service.
3. Click **Manage Actions** > **Purge Orphaned Tables**.

The Content Management Service deletes all reference table data that does not belong to a reference table object in the associated Model repository.

To prevent accidental data loss, the purge operation does not delete tables if the Model repository does not contain a reference table object.

Note: To delete unused reference table at the command prompt, run the `infacmd cms Purge` command.

Recycling and Disabling the Content Management Service

Recycle the Content Management Service to apply the latest service or service process options. Disable the Content Management Service to restrict user access to information about reference data in the Developer tool.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select **Content Management Service** > **Disable** to stop the service.

When you disable the Content Management Service, you must choose the mode to disable it in. You can choose one of the following options:

- **Complete.** Allows the jobs to run to completion before disabling the service.
 - **Abort.** Tries to stop all jobs before aborting them and disabling the service.
3. Click the **Recycle** button to restart the service. The Data Integration Service must be running before you recycle the Content Management Service.

You recycle the Content Management Service in the following cases:

- Recycle the Content Management Service after you add or update address reference data files or after you change the file location for probabilistic or classifier model data files.

- Recycle the Content Management Service and the associated Data Integration Service after you update the address validation properties, reference data location, identity cache directory, or identity index directory on the Content Management Service.

When you update the reference data location on the Content Management Service, recycle the Analyst Service associated with the Model Repository Service that the Content Management Service uses. Open a Developer tool or Analyst tool application to refresh the reference data location stored by the application.

Content Management Service Properties

To view the Content Management Service properties, select the service in the Domain Navigator and click the Properties view.

You can configure the following Content Management Service properties:

- General properties
- Multi-service options
- Associated services and reference data location properties
- File transfer options
- Logging options
- Custom properties

If you update a property, restart the Content Management Service to apply the update.

General Properties

General properties for the Content Management Service include the name and description of the Content Management Service, and the node in the Informatica domain that the Content Management Service runs on. You configure these properties when you create the Content Management Service.

The following table describes the general properties for the service:

Property	Description
Name	Name of the service. The name is not case sensitive and must be unique within the domain. It cannot exceed 128 characters or begin with @. It also cannot contain spaces or the following special characters: ` ~ % ^ * + = { } \ ; : ' " / ? . , < > ! () [] You cannot change the name of the service after you create it.
Description	Description of the service. The description cannot exceed 765 characters.
Node	Node on which the service runs. If you change the node, you must recycle the Content Management Service.
License	License object that allows use of the service.

Multi-Service Options

The Multi-service options indicate whether the current service is the master Content Management Service in a domain.

The following table describes the single property under multi-service options:

Property	Description
Master CMS	Indicates the master status of the service. The master Content Management Service is the first service you create on a domain. The Master CMS property defaults to True when it is the first Content Management Service on a domain. Otherwise, the Master CMS property defaults to False.

Associated Services and Reference Data Location Properties

The Associated Services and Reference Data Location Properties identify the services associated with the Content Management Service. It also identifies the database that stores reference data values for associated reference data objects.

The following table describes the associated services and reference data location properties for the Content Management Service:

Property	Description
Data Integration Service	Data Integration Service associated with the Content Management Service. The Data Integration Service reads reference data configuration information from the Content Management Service. Recycle the Content Management Service if you associate another Data Integration Service with the Content Management Service.
Model Repository Service	Model Repository Service associated with the Content Management Service. Recycle the Content Management Service if you associate another Model Repository Service with the Content Management Service.
Username	User name that the Content Management Service uses to connect to the Model Repository Service. To perform reference table management tasks in the Model repository, the user that the property identifies must have the Model Repository Service Administrator role. The reference table management tasks include purge operations on orphaned reference tables. Not available for a domain with Kerberos authentication.
Password	Password that the Content Management Service uses to connect to the Model Repository Service. Not available for a domain with Kerberos authentication.
Reference Data Location	Database connection name for the database that stores reference data values for the reference data objects defined in the associated Model repository. The database stores reference data object row values. The Model repository stores metadata for reference data objects.

File Transfer Options

The File Transfer Options property identifies a directory on the Informatica services machine that the Content Management Service can use to store data when a user imports data to a reference table.

When you import data to a reference table, the Content Management Service uses a local directory structure as a staging area. The Content Management Service clears the directory when the reference table update is complete.

The following table describes the File Transfer Options property:

Property	Description
Temporary File Location	Path to the directory that stores reference data during the import process.

Logging Options

Configure the Log Level property to set the logging level.

The following table describes the Log Level properties:

Property	Description
Log Level	<p>Configure the Log Level property to set the logging level. The following values are valid:</p> <ul style="list-style-type: none">- Fatal. Writes FATAL messages to the log. FATAL messages include nonrecoverable system failures that cause the service to shut down or become unavailable.- Error. Writes FATAL and ERROR code messages to the log. ERROR messages include connection failures, failures to save or retrieve metadata, service errors.- Warning. Writes FATAL, WARNING, and ERROR messages to the log. WARNING errors include recoverable system failures or warnings.- Info. Writes FATAL, INFO, WARNING, and ERROR messages to the log. INFO messages include system and service change messages.- Trace. Write FATAL, TRACE, INFO, WARNING, and ERROR code messages to the log. TRACE messages log user request failures.- Debug. Write FATAL, DEBUG, TRACE, INFO, WARNING, and ERROR messages to the log. DEBUG messages are user request logs.

Custom Properties for the Content Management Service

Configure custom properties that are unique to specific environments.

You might need to apply custom properties in special cases. When you define a custom property, enter the property name and an initial value. Define custom properties only at the request of Informatica Global Customer Support.

Content Management Service Process Properties

The Content Management Service runs the Content Management Service process on the same node as the service. When you select the Content Management Service in the Administrator tool, you can view the service process for the Content Management Service on the **Processes** tab.

You can view the node properties for the service process on the **Processes** tab. Select the node to view the service process properties.

You can configure the following types of Content Management Service process properties:

- Content Management Service security options
- Address validation properties
- Identity properties
- Advanced properties
- NLP option properties
- Custom properties

If you update any of the Content Management Service process properties, restart the Content Management Service for the modifications to take effect.

Note: The Content Management Service does not currently use the Content Management Service Security Options properties.

Content Management Service Security Options

You can configure the Content Management Service to communicate with other components in the Informatica domain in secure mode.

The following table describes the Content Management Service security options:

Property	Description
HTTP Port	Unique HTTP port number for the Content Management Service. Default is 8105. Recycle the service if you change the HTTP port number.
HTTPS Port	HTTPS port number that the service runs on when you enable the Transport Layer Security (TLS) protocol. Use a different port number than the HTTP port number. Recycle the service if you change the HTTPS port number.
Keystore File	Path and file name of the keystore file that contains the private or public key pairs and associated certificates. Required if you enable TLS and use HTTPS connections for the service.
Keystore Password	Plain-text password for the keystore file.
SSL Protocol	Informatica recommends that you leave this field blank. The version of TLS enabled depends on the value. A blank field enables the highest version of TLS available. If you enter a value, earlier versions of TLS might be enabled. The behavior is based on the Java version for your environment. For more information, see the documentation for your Java version.

Address Validation Properties

Configure address validation properties to determine how the Data Integration Service and the Developer tool read address reference data files. After you update address validation properties, you must recycle the Content Management Service and the Data Integration Service.

The following table describes the address validation properties for the Content Management Service process:

Property	Description
License	License key to activate validation reference data. You might have more than one key, for example, if you use batch reference data and geocoding reference data. Enter keys as a comma-delimited list. The property is empty by default.
Reference Data Location	Location of the address reference data files. Enter the full path to the files. Install all address reference data files to a single location. The property is empty by default.
Full Pre-Load Countries	<p>List of countries for which all batch, CAMEO, certified, interactive, or supplementary reference data is loaded into memory before address validation begins. Enter the three-character ISO country codes in a comma-separated list. For example, enter DEU,FRA,USA. Enter ALL to load all data sets. The property is empty by default.</p> <p>Load the full reference database to increase performance. Some countries, such as the United States, have large databases that require significant amounts of memory.</p>
Partial Pre-Load Countries	<p>List of countries for which batch, CAMEO, certified, interactive, or supplementary reference metadata and indexing structures are loaded into memory before address validation begins. Enter the three-character ISO country codes in a comma-separated list. For example, enter DEU,FRA,USA. Enter ALL to partially load all data sets. The property is empty by default.</p> <p>Partial preloading increases performance when not enough memory is available to load the complete databases into memory.</p>
No Pre-Load Countries	List of countries for which no batch, CAMEO, certified, interactive, or supplementary reference data is loaded into memory before address validation begins. Enter the three-character ISO country codes in a comma-separated list. For example, enter DEU,FRA,USA. Default is ALL.
Full Pre-Load Geocoding Countries	<p>List of countries for which all geocoding reference data is loaded into memory before address validation begins. Enter the three-character ISO country codes in a comma-separated list. For example, enter DEU,FRA,USA. Enter ALL to load all data sets. The property is empty by default.</p> <p>Load all reference data for a country to increase performance when processing addresses from that country. Some countries, such as the United States, have large data sets that require significant amounts of memory.</p>
Partial Pre-Load Geocoding Countries	<p>List of countries for which geocoding reference metadata and indexing structures are loaded into memory before address validation begins. Enter the three-character ISO country codes in a comma-separated list. For example, enter DEU,FRA,USA. Enter ALL to partially load all data sets. The property is empty by default.</p> <p>Partial preloading increases performance when not enough memory is available to load the complete databases into memory.</p>
No Pre-Load Geocoding Countries	List of countries for which no geocoding reference data is loaded into memory before address validation begins. Enter the three-character ISO country codes in a comma-separated list. For example, enter DEU,FRA,USA. Default is ALL.
Full Pre-Load Suggestion List Countries	<p>List of countries for which all suggestion list reference data is loaded into memory before address validation begins. Enter the three-character ISO country codes in a comma-separated list. For example, enter DEU,FRA,USA. Enter ALL to load all data sets. The property is empty by default.</p> <p>Load the full reference database to increase performance. Some countries, such as the United States, have large databases that require significant amounts of memory.</p>

Property	Description
Partial Pre-Load Suggestion List Countries	<p>List of countries for which the suggestion list reference metadata and indexing structures are loaded into memory before address validation begins. Enter the three-character ISO country codes in a comma-separated list. For example, enter DEU,FRA,USA. Enter ALL to partially load all data sets. The property is empty by default.</p> <p>Partial preloading increases performance when not enough memory is available to load the complete databases into memory.</p>
No Pre-Load Suggestion List Countries	<p>List of countries for which no suggestion list reference data is loaded into memory before address validation begins. Enter the three-character ISO country codes in a comma-separated list. For example, enter DEU,FRA,USA. Default is ALL.</p>
Full Pre-Load Address Code Countries	<p>List of countries for which all address code lookup reference data is loaded into memory before address validation begins. Enter the three-character ISO country codes in a comma-separated list. For example, enter DEU,FRA,USA. Enter ALL to load all data sets. The property is empty by default.</p> <p>Load the full reference database to increase performance. Some countries, such as the United States, have large databases that require significant amounts of memory.</p>
Partial Pre-Load Address Code Countries	<p>List of countries for which the address code lookup reference metadata and indexing structures are loaded into memory before address validation begins. Enter the three-character ISO country codes in a comma-separated list. For example, enter DEU,FRA,USA. Enter ALL to partially load all data sets. The property is empty by default.</p> <p>Partial preloading increases performance when not enough memory is available to load the complete databases into memory.</p>
No Pre-Load Address Code Countries	<p>List of countries for which no address code lookup reference data is loaded into memory before address validation begins. Enter the three-character ISO country codes in a comma-separated list. For example, enter DEU,FRA,USA. Default is ALL.</p>
Preloading Method	<p>Determines how the Data Integration Service preloads address reference data into memory. The MAP method and the LOAD method both allocate a block of memory and then read reference data into this block. However, the MAP method can share reference data between multiple processes. Default is MAP.</p>
Max Result Count	<p>Maximum number of addresses that address validation can return in suggestion list mode. Set a maximum number in the range 1 through 100. Default is 20.</p>
Memory Usage	<p>Number of megabytes of memory that the address validation library files can allocate. Default is 4096.</p>
Max Address Object Count	<p>Maximum number of address validation instances to run at the same time. Default is 3. Set a value that is greater than or equal to the Maximum Parallelism value on the Data Integration Service.</p>
Max Thread Count	<p>Maximum number of threads that address validation can use. Set to the total number of cores or threads available on a machine. Default is 2.</p>

Property	Description
Cache Size	<p>Size of cache for databases that are not preloaded. Caching reserves memory to increase lookup performance in reference data that has not been preloaded.</p> <p>Set the cache size to LARGE unless all the reference data is preloaded or you need to reduce the amount of memory usage.</p> <p>Enter one of the following options for the cache size in uppercase letters:</p> <ul style="list-style-type: none"> - NONE. No cache. Enter NONE if all reference databases are preloaded. - SMALL. Reduced cache size. - LARGE. Standard cache size. <p>Default is LARGE.</p>
SendRight Report Location	<p>Location to which an address validation mapping writes a SendRight report and any log file that relates to the report. You generate a SendRight report to verify that a set of New Zealand address records meets the certification standards of New Zealand Post. Enter a local path on the machine that hosts the Data Integration Service that runs the mapping.</p> <p>By default, address validation writes the report file to the <code>bin</code> directory of the Informatica installation. If you enter a relative path, the Content Management Service appends the path to the <code>bin</code> directory.</p>

Rules and Guidelines for Address Reference Data Preload Options

If you run a mapping that reads address reference data, verify the policy that the Data Integration Service uses to load the data into memory. To configure the policy, use the preload options on the address validation process properties. The Data Integration Service reads the preload options from the Content Management Service when an address validation mapping runs.

Consider the following rules and guidelines when you configure the preload options on the Content Management Service:

- By default, the Content Management Service applies the ALL value to the options that indicate no data preload. If you accept the default options, the Data Integration Service reads the address reference data from files in the directory structure when the mapping runs.
- The address validation process properties must indicate a preload method for each type of address reference data that a mapping specifies. If the Data Integration Service cannot determine a preload policy for a type of reference data, it ignores the reference data when the mapping runs.
- The Data Integration Service can use a different method to load data for each country. For example, you can specify full preload for United States suggestion list data and partial preload for United Kingdom suggestion list data.
- The Data Integration Service can use a different preload method for each type of data. For example, you can specify full preload for United States batch data and partial preload for United States address code data.
- Full preload settings supersede partial preload settings, and partial preload settings supersede settings that indicate no data preload.

For example, you might configure the following options:

Full Pre-Load Geocoding Countries: DEU

No Pre-Load Geocoding Countries: ALL

The options specify that the Data Integration Service loads German geocoding data into memory and does not load geocoding data for any other country.

- The Data Integration Service loads the types of address reference data that you specify in the address validation process properties. The Data Integration Service does not read the mapping metadata to identify the address reference data that the mapping specifies.

Identity Properties

The identity properties specify the location of the identity population files and the default locations of the temporary files that identity match analysis can generate. The locations on each property are local to the Data Integration Service that runs the identity match mapping. The Data Integration Service must have write access to each location.

The following table describes the identity properties:

Property	Description
Reference Data Location	<p>Path to the directory that contains the identity population files.</p> <p>The path identifies a parent directory. Install the population files to a directory with the name <code>default</code> below the directory that the property specifies.</p>
Cache Directory	<p>Path to the directory that contains the temporary data files that the Data Integration Service generates during identity analysis. The Data Integration Service creates the directory at run time if the Match transformation in the mapping does not specify the directory.</p> <p>The property sets the following default path:</p> <p><code>./identityCache</code></p> <p>You can specify a relative path, or you can specify a fully qualified path to a directory that the Data Integration Service can write to. The relative path is relative to the <code>tomcat/bin</code> directory on the Data Integration Service machine.</p>
Index Directory	<p>Path to the directory that contains the temporary index files that the Data Integration Service generates during identity analysis. Identity match analysis uses the index to sort records into groups before match analysis. The Data Integration Service creates the directory at run time if the Match transformation in the mapping does not specify the directory.</p> <p>The property sets the following default location:</p> <p><code>./identityIndex</code></p> <p>You can specify a relative path, or you can specify a fully qualified path to a directory that the Data Integration Service can write to. The relative path is relative to the <code>tomcat/bin</code> directory on the Data Integration Service machine.</p>

Advanced Properties

The advanced properties define the maximum heap size and the Java Virtual Manager (JVM) memory settings.

The following table describes the advanced properties for service process:

Property	Description
Maximum Heap Size	<p>Amount of RAM allocated to the Java Virtual Machine (JVM) that runs the service. Use this property to increase the memory available to the service. Append one of the following letters to the value to specify the units:</p> <ul style="list-style-type: none">- b for bytes- k for kilobytes- m for megabytes- g for gigabytes <p>Default is 512 megabytes.</p>
JVM Command Line Options	<p>Java Virtual Machine (JVM) command line options to run Java-based programs. When you configure the JVM options, you must set the Java SDK classpath, Java SDK minimum memory, and Java SDK maximum memory properties.</p>

Note: If you use Informatica Developer to compile probabilistic models, increase the default maximum heap size value to 3 gigabytes.

NLP Options

The NLP Options property provides the location of probabilistic model and classifier model files on the Informatica services machine. Probabilistic models and classifier models are types of reference data. Use the models in transformations that perform Natural Language Processing (NLP) analysis.

The following table describes the NLP Options property:

Property	Description
NER File Location	Path to the probabilistic model files. The property reads a relative path from the following directory in the Informatica installation: <code>/tomcat/bin</code> The default value is <code>./ner</code> , which indicates the following directory: <code>/tomcat/bin/ner</code>
Classifier File Location	Path to the classifier model files. The property reads a relative path from the following directory in the Informatica installation: <code>/tomcat/bin</code> The default value is <code>./classifier</code> , which indicates the following directory: <code>/tomcat/bin/classifier</code>

Custom Properties for the Content Management Service Process

Configure custom properties that are unique to specific environments.

You might need to apply custom properties in special cases. When you define a custom property, enter the property name and an initial value. Define custom properties only at the request of Informatica Global Customer Support.

Creating a Content Management Service

Before you create a Content Management Service, verify that the domain contains a Data Integration Service and Model Repository Service. You must also know the connection name of a database that the Content Management Service can use to store reference data.

Create a Content Management Service to manage reference data properties and to provide the Developer tool with information about installed reference data.

1. On the **Manage** tab, select the **Services and Nodes** view.
2. Select the domain name.
3. Click **Actions > New > Content Management Service**.
The **New Content Management Service** window appears.
4. Enter a name and optional description for the service.
5. Set the location for the service. You can create the service in a folder on the domain. Click **Browse** to create a folder.

6. Select the node that you want the service to run on.
7. Specify a Data Integration Service and Model Repository Service to associate with the Content Management Service.
8. Enter a username and password that the Content Management Service can use to connect to the Model Repository Service.
9. Select the database that the Content Management Service can use to store reference data.
10. Click **Next**.
11. Optionally, select **Enable Service** to enable the service after you create it.
Note: Do not configure the Transport Layer Security properties. The properties are reserved for future use.
12. Click **Finish**.
If you did not choose to enable the service, you must recycle the service to start it.

CHAPTER 3

Data Integration Service

This chapter includes the following topics:

- [Data Integration Service Overview, 49](#)
- [Before You Create the Data Integration Service, 50](#)
- [Creating a Data Integration Service, 51](#)
- [Data Integration Service Properties, 54](#)
- [Data Integration Service Process Properties, 67](#)
- [Data Integration Service Compute Properties, 70](#)
- [Operating System Profiles for the Data Integration Service, 72](#)
- [High Availability for the Data Integration Service, 75](#)

Data Integration Service Overview

The Data Integration Service is an application service in the Informatica domain that performs data integration tasks for Informatica Analyst and Informatica Developer. It also performs data integration tasks for external clients.

When you preview or run mappings, profiles, SQL data services, and web services in the Analyst tool or the Developer tool, the application client sends requests to the Data Integration Service to perform the data integration tasks. When you start a command from the command line or an external client to run mappings, SQL data services, web services, and workflows in an application, the command sends the request to the Data Integration Service.

The Data Integration Service performs the following tasks:

- Runs mappings and generates mapping previews in the Developer tool.
- Runs profiles and generates previews for profiles in the Analyst tool and the Developer tool.
- Runs scorecards for the profiles in the Analyst tool and the Developer tool.
- Runs SQL data services and web services in the Developer tool.
- Runs mappings in a deployed application.
- Runs workflows in a deployed application.
- Caches data objects for mappings and SQL data services deployed in an application.
- Runs SQL queries that end users run against an SQL data service through a third-party JDBC or ODBC client tool.

- Runs web service requests against a web service.

Create and configure a Data Integration Service in the Administrator tool. You can create one or more Data Integration Services on a node. Based on your license, the Data Integration Service can be highly available.

Before You Create the Data Integration Service

Before you create the Data Integration Service, complete the prerequisite tasks for the service.

Perform the following tasks before you create the Data Integration Service:

- Set up the databases that the Data Integration Service connects to.
- Create connections to the databases.
- If the domain uses Kerberos authentication and you set the service principal level at the process level, create a keytab file for the Data Integration Service.
- Create the associated Model Repository Service.

Create Required Databases

The Data Integration Service can connect to multiple relational databases. The databases that the service can connect to depend on the license key generated for your organization. When you create the Data Integration Service, you provide connection information to the databases.

Create the following databases before you create the Data Integration Service:

Data object cache database

Stores cached logical data objects and virtual tables. Data object caching enables the Data Integration Service to access pre-built logical data objects and virtual tables. You need a data object cache database to increase performance for mappings, SQL data service queries, and web service requests.

Profiling warehouse

Stores profiling information, such as profile results and scorecard results. You need a profiling warehouse to perform profiling and data discovery.

Workflow database

Stores all run-time metadata for workflows, including Human task metadata.

For more information about the database requirements, see [Appendix A, “Application Service Databases” on page 386](#).

The Data Integration Service uses native database drivers to connect to the data object cache database, the profiling warehouse, and source and target databases. To establish native connectivity between the service and a database, install the database client software for the database that you want to access. For more information, see [“Configure Native Connectivity on Service Machines” on page 401](#).

Create Connections to the Databases

The Data Integration Service uses connections to access the databases. You specify the connection details when you create the service.

When you create the database connection in the Administrator tool, specify the database connection properties and test the connection.

The following table describes the database connections that you must create before you create the Data Integration Service:

Database Connection	Description
Data object cache database	To access the data object cache, create the data object cache connection for the Data Integration Service.
Workflow database	To store run-time metadata for workflows, create the workflow database connection for the Data Integration Service.
Profiling warehouse database	<p>To create and run profiles and scorecards, create the profiling warehouse database connection for the Data Integration Service. Use this instance of the Data Integration Service when you configure the run-time properties of the Analyst Service.</p> <p>You can create the following types of profiles when you use a JDBC connection for the profiling warehouse:</p> <ul style="list-style-type: none">- Column profile- Rule profile- Data domain discovery profile- Enterprise discovery profile without enabling the foreign key discovery <p>You can also create scorecards when you use a JDBC connection for the profiling warehouse.</p> <p>Note: To use the Microsoft SQL Server database as the profiling warehouse, choose ODBC as the provider type, and clear the use DSN option in the Microsoft SQL Server connection properties dialog box when you configure the Microsoft SQL Server connection.</p>

Create the Service Principal Name and Keytab File

If the Informatica domain uses Kerberos authentication and you set the service principal level for the domain to process level, the domain requires an SPN and keytab file for each application service that you create in the domain.

Before you enable a service, verify that an SPN and a keytab file are available for the service. Kerberos cannot authenticate the application service if the service does not have a keytab file in the Informatica directory.

For more information about creating the service principal names and keytab files, see the *Informatica Security Guide*.

Create Associated Services

The Data Integration Service connects to the Model Repository Service to perform jobs such as running mappings, workflows, and profiles.

Create the Model Repository Service before you create the Data Integration Service. When you create the Data Integration Service, you provide the name of the Model Repository Service. You can associate the same Model Repository Service to multiple Data Integration Services.

Creating a Data Integration Service

Use the service creation wizard in the Administrator tool to create the service.

1. In the Administrator tool, click the **Manage** tab.

2. Click the **Services and Nodes** view.
3. In the Domain Navigator, select the domain.
4. Click **Actions > New > Data Integration Service**.
The **New Data Integration Service** wizard appears.
5. On the **New Data Integration Service - Step 1 of 14** page, enter the following properties:

Property	Description
Name	Name of the service. The name is not case sensitive and must be unique within the domain. It cannot exceed 128 characters or begin with @. It also cannot contain spaces or the following special characters: ` ~ % ^ * + = { } \ ; : ' " / ? . , < > ! () []
Description	Description of the service. The description cannot exceed 765 characters.
Location	Domain and folder where the service is created. Click Browse to choose a different folder. You can move the service after you create it.
License	License object that allows use of the service.
Assign	Select Node to configure the service to run on a node. If your license includes grid, you can create a grid and assign the service to run on the grid after you create the service.
Node	Node on which the service runs.
Backup Nodes	If your license includes high availability, nodes on which the service can run if the primary node is unavailable.
Model Repository Service	Model Repository Service to associate with the service.
Username	User name that the service uses to access the Model Repository Service. Enter the Model repository user that you created.
Password	Password for the Model repository user.
Security Domain	LDAP security domain for the Model repository user. The field appears when the Informatica domain contains an LDAP security domain. Not available for a domain with Kerberos authentication.

6. Click **Next**.
The **New Data Integration Service - Step 2 of 14** page appears.
7. Enter the HTTP port number to use for the Data Integration Service.
8. Accept the default values for the remaining security properties. You can configure the security properties after you create the Data Integration Service.
9. Select **Enable Service**.
The Model Repository Service must be running to enable the Data Integration Service.
10. Verify that the **Move to plugin configuration page** is not selected.
11. Click **Next**.
The **New Data Integration Service - Step 3 of 14** page appears.

12. Set the **Launch Job Options** property to one of the following values:
- In the service process. Configure when you run SQL data service and web service jobs. SQL data service and web service jobs typically achieve better performance when the Data Integration Service runs jobs in the service process.
 - In separate local processes. Configure when you run mapping, profile, and workflow jobs. When the Data Integration Service runs jobs in separate local processes, stability increases because an unexpected interruption to one job does not affect all other jobs.

If you configure the Data Integration Service to run on a grid after you create the service, you can configure the service to run jobs in separate remote processes.

13. Accept the default values for the remaining execution options and click **Next**.

The **New Data Integration Service - Step 4 of 14** page appears.

14. If you created the data object cache database for the Data Integration Service, click **Select** to select the cache connection. Select the data object cache connection that you created for the service to access the database.

15. Accept the default values for the remaining properties on this page and click **Next**.

The **New Data Integration Service - Step 5 of 14** page appears.

16. For optimal performance, enable the Data Integration Service modules that you plan to use.

The following table lists the Data Integration Service modules that you can enable:

Module	Description
Web Service Module	Runs web service operation mappings.
Mapping Service Module	Runs mappings and previews.
Profiling Service Module	Runs profiles and scorecards.
SQL Service Module	Runs SQL queries from a third-party client tool to an SQL data service.
Workflow Orchestration Service Module	Runs workflows.

17. Click **Next**.

The **New Data Integration Service - Step 6 of 14** page appears.

You can configure the HTTP proxy server properties to redirect HTTP requests to the Data Integration Service. You can configure the HTTP configuration properties to filter the web services client machines that can send requests to the Data Integration Service. You can configure these properties after you create the service.

18. Accept the default values for the HTTP proxy server and HTTP configuration properties and click **Next**.

The **New Data Integration Service - Step 7 of 14** page appears.

The Data Integration Service uses the result set cache properties to use cached results for SQL data service queries and web service requests. You can configure the properties after you create the service.

19. Accept the default values for the result set cache properties and click **Next**.

The **New Data Integration Service - Step 8 of 14** page appears.

20. If you created the profiling warehouse database for the Data Integration Service, select the Profiling Service module.

21. If you created the workflow database for the Data Integration Service, select the Workflow Orchestration Service module.
22. Verify that the remaining modules are not selected.
You can configure properties for the remaining modules after you create the service.
23. Click **Next**.
The **New Data Integration Service - Step 11 of 14** page appears.
24. If you created the profiling warehouse database for the Data Integration Service, click **Select** to select the database connection. Select the profiling warehouse connection that you created for the service to access the database.
25. Select whether or not content exists in the profiling warehouse database.
If you created a new profiling warehouse database, select **No content exists under specified connection string**.
26. Click **Next**.
The **New Data Integration Service - Step 12 of 14** page appears.
27. Accept the default values for the advanced profiling properties and click **Next**.
The **New Data Integration Service - Step 14 of 14** page appears.
28. If you created the workflow database for the Data Integration Service, click **Select** to select the database connection. Select the workflow database connection that you created for the service to access the database.
29. Click **Finish**.
The domain creates and enables the Data Integration Service.
After you create the service through the wizard, you can edit the properties or configure other properties.

Data Integration Service Properties

To view the Data Integration Service properties, select the service in the Domain Navigator and click the Properties view. You can change the properties while the service is running, but you must restart the service for the properties to take effect.

General Properties

The general properties of a Data Integration Service includes name, license, and node assignment.

The following table describes the general properties for the service:

General Property	Description
Name	Name of the service. The name is not case sensitive and must be unique within the domain. It cannot exceed 128 characters or begin with @. It also cannot contain spaces or the following special characters: ` ~ % ^ * + = { } \ ; : ' " / ? . , < > ! () [] You cannot change the name of the service after you create it.
Description	Description of the service. The description cannot exceed 765 characters.

General Property	Description
License	License object that allows use of the service.
Assign	Node or grid on which the Data Integration Service runs.
Node	Node on which the service runs.
Grid	Name of the grid on which the Data Integration Service runs if the service runs on a grid. Click the grid name to view the grid configuration.
Backup Nodes	If your license includes high availability, nodes on which the service can run if the primary node is unavailable.

Model Repository Properties

The following table describes the Model repository properties for the Data Integration Service:

Property	Description
Model Repository Service	Service that stores run-time metadata required to run mappings and SQL data services.
User Name	User name to access the Model repository. The user must have the Create Project privilege for the Model Repository Service. Not available for a domain with Kerberos authentication.
Password	User password to access the Model repository. Not available for a domain with Kerberos authentication.

Execution Options

The following table describes the execution options for the Data Integration Service:

Property	Description
Use Operating System Profiles and Impersonation	<p>Runs mappings, workflows, and profiling jobs with operating system profiles.</p> <p>In a Hadoop environment, the Data Integration Service uses the Hadoop impersonation user to run mappings, workflows, and profiling jobs.</p> <p>You can select this option if the Data Integration Service runs on UNIX or Linux. To apply changes, restart the Data Integration Service.</p>
Launch Job Options	<p>Runs jobs in the Data Integration Service process, in separate DTM processes on the local node, or in separate DTM processes on remote nodes. Configure the property based on whether the Data Integration Service runs on a single node or a grid and based on the types of jobs that the service runs.</p> <p>Choose one of the following options:</p> <ul style="list-style-type: none">- In the service process. Configure when you run jobs on a single node or on a grid where each node has both the service and compute roles.- In separate local processes. Configure when you run jobs on a single node or on a grid where each node has both the service and compute roles.- In separate remote processes. Configure when you run mapping, profile, and workflow jobs on a grid where nodes have a different combination of roles. If you choose this option when the Data Integration Service runs on a single node, then the service runs jobs in separate local processes. You cannot run SQL data service or web service jobs in separate remote processes. <p>Default is in separate local processes.</p> <p>If the Data Integration Service uses operating system profiles, configure to run jobs in separate local processes.</p> <p>Note: If the Data Integration Service runs on UNIX and is configured to run jobs in separate local or remote processes, verify that the host file on each node with the compute role contains a localhost entry. Otherwise, jobs that run in separate processes fail.</p>
Maximum On-Demand Execution Pool Size	<p>Maximum number of on-demand jobs that can run concurrently. Jobs include data previews, profiling jobs, REST and SQL queries, web service requests, and mappings run from the Developer tool. The Data Integration Service immediately runs on-demand jobs if enough resources are available. Otherwise, the Data Integration Service rejects the job. Default is 10.</p>
Maximum Native Batch Execution Pool Size	<p>Maximum number of deployed jobs that can run concurrently in the native environment. The Data Integration Service moves native mapping jobs from the queue to the native job pool when enough resources are available. Default is 10.</p>
Maximum Hadoop Batch Execution Pool Size	<p>Maximum number of deployed jobs that can run concurrently in the Hadoop environment. The Data Integration Service moves Hadoop jobs from the queue to the Hadoop job pool when enough resources are available. Default is 100.</p>

Property	Description
Maximum Memory Size	<p>Maximum amount of memory, in bytes, that the Data Integration Service can allocate for running all requests concurrently when the service runs jobs in the Data Integration Service process. When the Data Integration Service runs jobs in separate local or remote processes, the service ignores this value. If you do not want to limit the amount of memory the Data Integration Service can allocate, set this property to 0.</p> <p>If the value is greater than 0, the Data Integration Service uses the property to calculate the maximum total memory allowed for running all requests concurrently. The Data Integration Service calculates the maximum total memory as follows:</p> <p>Maximum Memory Size + Maximum Heap Size + memory required for loading program components</p> <p>Default is 0.</p> <p>Note: If you run profiles or data quality mappings, set this property to 0.</p>
Maximum Parallelism	<p>Maximum number of parallel threads that process a single mapping pipeline stage.</p> <p>When you set the value greater than 1, the Data Integration Service enables partitioning for mappings, column profiling, and data domain discovery. The service dynamically scales the number of partitions for a mapping pipeline at run time. Increase the value based on the number of CPUs available on the nodes where jobs run.</p> <p>In the Developer tool, developers can change the maximum parallelism value for each mapping. When maximum parallelism is set for both the Data Integration Service and the mapping, the Data Integration Service uses the minimum value when it runs the mapping.</p> <p>Default is 1. Maximum is 64.</p> <p>This property change does not require a restart of the Data Integration Service.</p> <p>Note: Developers cannot change the maximum parallelism value for each profile. When the Data Integration Service converts a profile job into one or more mappings, the mappings always use Auto for the mapping maximum parallelism.</p>
Hadoop Kerberos Service Principal Name	<p>Service Principal Name (SPN) of the Data Integration Service to connect to a Hadoop cluster that uses Kerberos authentication.</p> <p>Not required when you run the MapR Hadoop distribution. Required for other Hadoop distributions.</p>
Hadoop Kerberos Keytab	<p>The file path to the Kerberos keytab file on the machine on which the Data Integration Service runs.</p> <p>Not required when you run the MapR Hadoop distribution. Required for other Hadoop distributions.</p>
Home Directory	<p>Root directory accessible by the node. This is the root directory for other service directories. Default is <Informatica installation directory>/tomcat/bin. If you change the default value, verify that the directory exists.</p> <p>You cannot use the following characters in the directory path:</p> <p>* ? < > " , []</p> <p>This property change does not require a restart of the Data Integration Service.</p>
Temporary Directories	<p>Directory for temporary files created when jobs are run. Default is <home directory>/disTemp.</p> <p>Enter a list of directories separated by semicolons to optimize performance during profile operations and during cache partitioning for Sorter transformations.</p> <p>You cannot use the following characters in the directory path:</p> <p>* ? < > " , []</p> <p>This property change does not require a restart of the Data Integration Service.</p>

Property	Description
Cache Directory	<p>Directory for index and data cache files for transformations. Default is <home directory>/cache.</p> <p>Enter a list of directories separated by semicolons to increase performance during cache partitioning for Aggregator, Joiner, or Rank transformations.</p> <p>You cannot use the following characters in the directory path:</p> <p>* ? < > " , []</p> <p>This property change does not require a restart of the Data Integration Service.</p>
Source Directory	<p>Directory for source flat files used in a mapping. Default is <home directory>/source.</p> <p>If the Data Integration Service runs on a grid, you can use a shared directory to create one directory for source files. If you configure a different directory for each node with the compute role, ensure that the source files are consistent among all source directories.</p> <p>You cannot use the following characters in the directory path:</p> <p>* ? < > " , []</p> <p>This property change does not require a restart of the Data Integration Service.</p>
Target Directory	<p>Default directory for target flat files used in a mapping. Default is <home directory>/target.</p> <p>Enter a list of directories separated by semicolons to increase performance when multiple partitions write to the flat file target.</p> <p>You cannot use the following characters in the directory path:</p> <p>* ? < > " , []</p> <p>This property change does not require a restart of the Data Integration Service.</p>
Rejected Files Directory	<p>Directory for reject files. Reject files contain rows that were rejected when running a mapping. Default is <home directory>/reject.</p> <p>You cannot use the following characters in the directory path:</p> <p>* ? < > " , []</p> <p>This property change does not require a restart of the Data Integration Service.</p>
Hadoop Staging Directory	<p>The HDFS directory where the Data Integration Services pushes Informatica Hadoop binaries and stores temporary files during processing. Default is /tmp.</p>
Hadoop Staging User	<p>The HDFS user that performs operations on the Hadoop staging directory. The user needs write permissions on Hadoop staging directory. Default is the Data Integration Service user.</p>
Custom Hadoop OS Path	<p>The local path to the Informatica Hadoop binaries compatible with the Hadoop operating system. Required when the Hadoop cluster and the Data Integration Service are on different supported operating systems. Download and extract the Informatica binaries for the Hadoop cluster on the machine that hosts the Data Integration Service. The Data Integration Service uses the binaries in this directory to integrate the domain with the Hadoop cluster. The Data Integration Service can synchronize the following operating systems:</p> <ul style="list-style-type: none"> - SUSE 11 and Redhat 6.5 <p>Changes take effect after you recycle the Data Integration Service.</p> <p>Note: When you install an Informatica EBF, you must also install it in the path of the Hadoop operating system on the Data Integration Service machine.</p>

Property	Description
Data Integration Service Hadoop Distribution Directory	The Hadoop distribution directory on the Data Integration Service node. The contents of the Data Integration Service Hadoop distribution directory must be identical to Hadoop distribution directory on the data nodes. Type <Informatica installation directory/Informatica/services/shared/hadoop/<Hadoop distribution name>.
State Store	<p>The HDFS location on the cluster to store information about the state of the Spark job. Default is <Home directory >/State Store</p> <p>Configure this property when you configure the run-time properties of a streaming mapping. This property change does not require a restart of the Data Integration Service.</p> <p>For more information about this property, see the <i>Informatica Intelligent Streaming User Guide</i> .</p>

Logical Data Object/Virtual Table Cache Properties

The following table describes the data object and virtual table cache properties:

Property	Description
Cache Removal Time	The number of milliseconds that the Data Integration Service waits before cleaning up cache storage after a refresh. Default is 3,600,000.
Cache Connection	The database connection name for the database that stores the data object cache. Select a valid connection object name.
Maximum Concurrent Refresh Requests	Maximum number of cache refreshes that can occur at the same time. Limit the concurrent cache refreshes to maintain system resources.
Enable Nested LDO Cache	<p>Indicates that the Data Integration Service can use cache data for a logical data object used as a source or a lookup in another logical data object during a cache refresh. If false, the Data Integration Service accesses the source resources even if you enabled caching for the logical data object used as a source or a lookup.</p> <p>For example, logical data object LD03 joins data from logical data objects LD01 and LD02. A developer creates a mapping that uses LD03 as the input and includes the mapping in an application. You enable caching for LD01, LD02, and LD03. If you enable nested logical data object caching, the Data Integration Service uses cache data for LD01 and LD02 when it refreshes the cache table for LD03. If you do not enable nested logical data object caching, the Data Integration Service accesses the source resources for LD01 and LD02 when it refreshes the cache table for LD03.</p> <p>Default is False.</p>

Logging Properties

The following table describes the log level properties:

Property	Description
Log Level	<p>Configure the Log Level property to set the logging level. The following values are valid:</p> <ul style="list-style-type: none">- Fatal. Writes FATAL messages to the log. FATAL messages include nonrecoverable system failures that cause the service to shut down or become unavailable.- Error. Writes FATAL and ERROR code messages to the log. ERROR messages include connection failures, failures to save or retrieve metadata, service errors.- Warning. Writes FATAL, WARNING, and ERROR messages to the log. WARNING errors include recoverable system failures or warnings.- Info. Writes FATAL, INFO, WARNING, and ERROR messages to the log. INFO messages include system and service change messages.- Trace. Write FATAL, TRACE, INFO, WARNING, and ERROR code messages to the log. TRACE messages log user request failures.- Debug. Write FATAL, DEBUG, TRACE, INFO, WARNING, and ERROR messages to the log. DEBUG messages are user request logs.

Deployment Options

The following table describes the deployment options for the Data Integration Service:

Property	Description
Default Deployment Mode	<p>Determines whether to enable and start each application after you deploy it to a Data Integration Service. Default Deployment mode affects applications that you deploy from the Developer tool, command line, and Administrator tool.</p> <p>Choose one of the following options:</p> <ul style="list-style-type: none">- Enable and Start. Enable the application and start the application.- Enable Only. Enable the application but do not start the application.- Disable. Do not enable the application.

Pass-through Security Properties

The following table describes the pass-through security properties:

Property	Description
Allow Caching	<p>Allows data object caching for all pass-through connections in the Data Integration Service. Populates data object cache using the credentials from the connection object.</p> <p>Note: When you enable data object caching with pass-through security, you might allow users access to data in the cache database that they might not have in an uncached environment.</p>

Modules

By default, all Data Integration Service modules are enabled. You can disable some of the modules.

You might want to disable a module if you are testing and you have limited resources on the computer. You can save memory by limiting the Data Integration Service functionality. Before you disable a module, you must disable the Data Integration Service.

The following table describes the Data Integration Service modules:

Module	Description
Web Service Module	Runs web service operation mappings.
Mapping Service Module	Runs mappings and previews.
Profiling Service Module	Runs profiles and generate scorecards.
SQL Service Module	Runs SQL queries from a third-party client tool to an SQL data service.
Workflow Orchestration Service Module	Runs workflows.

HTTP Proxy Server Properties

The following table describes the HTTP proxy server properties:

Property	Description
HTTP Proxy Server Host	Name of the HTTP proxy server.
HTTP Proxy Server Port	Port number of the HTTP proxy server. Default is 8080.
HTTP Proxy Server User	Authenticated user name for the HTTP proxy server. This is required if the proxy server requires authentication.
HTTP Proxy Server Password	Password for the authenticated user. The Service Manager encrypts the password. This is required if the proxy server requires authentication.
HTTP Proxy Server Domain	Domain for authentication.

HTTP Configuration Properties

The following table describes the HTTP Configuration Properties:

Property	Description
Allowed IP Addresses	<p>List of constants or Java regular expression patterns compared to the IP address of the requesting machine. Use a space to separate multiple constants or expressions.</p> <p>If you configure this property, the Data Integration Service accepts requests from IP addresses that match the allowed address pattern. If you do not configure this property, the Data Integration Service uses the Denied IP Addresses property to determine which clients can send requests.</p>
Allowed Host Names	<p>List of constants or Java regular expression patterns compared to the host name of the requesting machine. The host names are case sensitive. Use a space to separate multiple constants or expressions.</p> <p>If you configure this property, the Data Integration Service accepts requests from host names that match the allowed host name pattern. If you do not configure this property, the Data Integration Service uses the Denied Host Names property to determine which clients can send requests.</p>
Denied IP Addresses	<p>List of constants or Java regular expression patterns compared to the IP address of the requesting machine. Use a space to separate multiple constants or expressions.</p> <p>If you configure this property, the Data Integration Service accepts requests from IP addresses that do not match the denied IP address pattern. If you do not configure this property, the Data Integration Service uses the Allowed IP Addresses property to determine which clients can send requests.</p>
Denied Host Names	<p>List of constants or Java regular expression patterns compared to the host name of the requesting machine. The host names are case sensitive. Use a space to separate multiple constants or expressions.</p> <p>If you configure this property, the Data Integration Service accepts requests from host names that do not match the denied host name pattern. If you do not configure this property, the Data Integration Service uses the Allowed Host Names property to determine which clients can send requests.</p>
HTTP Protocol Type	<p>Security protocol that the Data Integration Service uses. Select one of the following values:</p> <ul style="list-style-type: none">- HTTP. Requests to the service must use an HTTP URL.- HTTPS. Requests to the service must use an HTTPS URL.- HTTP&HTTPS. Requests to the service can use either an HTTP or an HTTPS URL. <p>When you set the HTTP protocol type to HTTPS or HTTP&HTTPS, you enable Transport Layer Security (TLS) for the service.</p> <p>You can also enable TLS for each web service deployed to an application. When you enable HTTPS for the Data Integration Service and enable TLS for the web service, the web service uses an HTTPS URL. When you enable HTTPS for the Data Integration Service and do not enable TLS for the web service, the web service can use an HTTP URL or an HTTPS URL. If you enable TLS for a web service and do not enable HTTPS for the Data Integration Service, the web service does not start.</p> <p>Default is HTTP.</p>

Result Set Cache Properties

The following table describes the result set cache properties:

Property	Description
File Name Prefix	The prefix for the names of all result set cache files stored on disk. Default is RSCACHE.
Enable Encryption	Indicates whether result set cache files are encrypted using 128-bit AES encryption. Valid values are true or false. Default is true.

Mapping Service Properties

The following table describes Mapping Service Module properties for the Data Integration Service:

Property	Description
Maximum Notification Thread Pool Size	Maximum number of concurrent job completion notifications that the Mapping Service Module sends to external clients after the Data Integration Service completes jobs. The Mapping Service Module is a component in the Data Integration Service that manages requests sent to run mappings. Default is 5.
Maximum Memory Per Request	<p>The behavior of Maximum Memory Per Request depends on the following Data Integration Service configurations:</p> <ul style="list-style-type: none">- The service runs jobs in separate local or remote processes, or the service property Maximum Memory Size is 0 (default). Maximum Memory Per Request is the maximum amount of memory, in bytes, that the Data Integration Service can allocate to all transformations that use auto cache mode in a single request. The service allocates memory separately to transformations that have a specific cache size. The total memory used by the request can exceed the value of Maximum Memory Per Request.- The service runs jobs in the Data Integration Service process, and the service property Maximum Memory Size is greater than 0. Maximum Memory Per Request is the maximum amount of memory, in bytes, that the Data Integration Service can allocate to a single request. The total memory used by the request cannot exceed the value of Maximum Memory Per Request. <p>Default is 536,870,912. Requests include mappings and mappings run from Mapping tasks within a workflow.</p>

Profiling Warehouse Database Properties

The following table describes the profiling warehouse database properties:

Property	Description
Profiling Warehouse Database	The connection to the profiling warehouse. Select the connection object name.
Maximum Ranks	Number of minimum and maximum values to display for a profile. Default is 5.
Maximum Patterns	Maximum number of patterns to display for a profile. Default is 10.

Property	Description
Maximum Profile Execution Pool Size	Maximum number of threads to run profiling. Default is 10.
Maximum DB Connections	Maximum number of database connections for each profiling job. Default is 5.
Profile Results Export Path	Location where the Data Integration Service exports profile results file. If the Data Integration Service and Analyst Service run on different nodes, both services must be able to access this location. Otherwise, the export fails.
Maximum Memory Per Request	Maximum amount of memory, in bytes, that the Data Integration Service can allocate for each mapping run for a single profile request. Default is 536,870,912.

Advanced Profiling Properties

The following table describes the advanced profiling properties:

Property	Description
Pattern Threshold Percentage	Maximum number of values required to derive a pattern. Default is 5.
Maximum # Value Frequency Pairs	Maximum number of value-frequency pairs to store in the profiling warehouse. Default is 16,000.
Maximum String Length	Maximum length of a string that the Profiling Service can process. Default is 255.
Maximum Numeric Precision	Maximum number of digits for a numeric value. Default is 38.
Maximum Concurrent Profile Jobs	The maximum number of concurrent profile threads used to run a profile on flat files and relational sources. If left blank, the Profiling Service plug-in determines the best number based on the set of running jobs and other environment factors.
Maximum Concurrent Columns	Maximum number of columns that you can combine for profiling flat files in a single execution pool thread. Default is 5.
Maximum Concurrent Profile Threads	The maximum number of concurrent execution pool threads used to run a profile on flat files or relational data sources. Default is 1.
Maximum Column Heap Size	Amount of memory to allow each column for column profiling. Default is 64 megabytes.
Reserved Profile Threads	Number of threads of the Maximum Execution Pool Size that are for priority requests. Default is 1.

SQL Properties

The following table describes the SQL properties:

Property	Description
DTM Keep Alive Time	<p>Number of milliseconds that the DTM instance stays open after it completes the last request. Identical SQL queries can reuse the open instance. Use the keep alive time to increase performance when the time required to process the SQL query is small compared to the initialization time for the DTM instance. If the query fails, the DTM instance terminates.</p> <p>Must be greater than or equal to 0. 0 means that the Data Integration Service does not keep the DTM instance in memory. Default is 0.</p> <p>You can also set this property for each SQL data service that is deployed to the Data Integration Service. If you set this property for a deployed SQL data service, the value for the deployed SQL data service overrides the value you set for the Data Integration Service.</p>
Table Storage Connection	<p>Relational database connection that stores temporary tables for SQL data services. By default, no connection is selected.</p>
Maximum Memory Per Request	<p>The behavior of Maximum Memory Per Request depends on the following Data Integration Service configurations:</p> <ul style="list-style-type: none">- The service runs jobs in separate local or remote processes, or the service property Maximum Memory Size is 0 (default). Maximum Memory Per Request is the maximum amount of memory, in bytes, that the Data Integration Service can allocate to all transformations that use auto cache mode in a single request. The service allocates memory separately to transformations that have a specific cache size. The total memory used by the request can exceed the value of Maximum Memory Per Request.- The service runs jobs in the Data Integration Service process, and the service property Maximum Memory Size is greater than 0. Maximum Memory Per Request is the maximum amount of memory, in bytes, that the Data Integration Service can allocate to a single request. The total memory used by the request cannot exceed the value of Maximum Memory Per Request. <p>Default is 50,000,000.</p>
Skip Log Files	<p>Prevents the Data Integration Service from generating log files when the SQL data service request completes successfully and the tracing level is set to INFO or higher. Default is false.</p>

Workflow Orchestration Service Properties

The following table describes the Workflow Orchestration Service properties for the Data Integration Service:

Property	Description
Workflow Connection	<p>The connection name of the database that stores the run-time configuration data for the workflows that the Data Integration Service runs. Select a database on the Connections view.</p> <p>Create the workflow database contents before you run a workflow. To create the contents, use the Actions menu options for the Data Integration Service in the Administrator tool.</p> <p>Note: Recycle the Data Integration Service after you configure the workflow database connection and before you create the workflow database contents.</p>
Maximum Worker Threads	<p>The maximum number of threads that the Data Integration Service can use to run parallel tasks between a pair of inclusive gateways in a workflow. The default value is 10.</p> <p>If the number of tasks between the inclusive gateways is greater than the maximum value, the Data Integration Service runs the tasks in batches that the value specifies. For example, if the Maximum Worker Threads value is 10, the Data Integration Service runs the tasks in batches of ten.</p>

Web Service Properties

The following table describes the web service properties:

Property	Description
DTM Keep Alive Time	<p>Number of milliseconds that the DTM instance stays open after it completes the last request. Web service requests that are issued against the same operation can reuse the open instance. Use the keep alive time to increase performance when the time required to process the request is small compared to the initialization time for the DTM instance. If the request fails, the DTM instance terminates.</p> <p>Must be greater than or equal to 0. 0 means that the Data Integration Service does not keep the DTM instance in memory. Default is 5000.</p> <p>You can also set this property for each web service that is deployed to the Data Integration Service. If you set this property for a deployed web service, the value for the deployed web service overrides the value you set for the Data Integration Service.</p>
Logical URL	<p>Prefix for the WSDL URL if you use an external HTTP load balancer. For example,</p> <p><code>http://loadbalancer:8080</code></p> <p>The Data Integration Service requires an external HTTP load balancer to run a web service on a grid. If you run the Data Integration Service on a single node, you do not need to specify the logical URL.</p>

Property	Description
Maximum Memory Per Request	<p>The behavior of Maximum Memory Per Request depends on the following Data Integration Service configurations:</p> <ul style="list-style-type: none"> - The service runs jobs in separate local or remote processes, or the service property Maximum Memory Size is 0 (default). Maximum Memory Per Request is the maximum amount of memory, in bytes, that the Data Integration Service can allocate to all transformations that use auto cache mode in a single request. The service allocates memory separately to transformations that have a specific cache size. The total memory used by the request can exceed the value of Maximum Memory Per Request. - The service runs jobs in the Data Integration Service process, and the service property Maximum Memory Size is greater than 0. Maximum Memory Per Request is the maximum amount of memory, in bytes, that the Data Integration Service can allocate to a single request. The total memory used by the request cannot exceed the value of Maximum Memory Per Request. <p>Default is 50,000,000.</p>
Skip Log Files	Prevents the Data Integration Service from generating log files when the web service request completes successfully and the tracing level is set to INFO or higher. Default is false.

Custom Properties for the Data Integration Service

Configure custom properties that are unique to specific environments.

You might need to apply custom properties in special cases. When you define a custom property, enter the property name and an initial value. Define custom properties only at the request of Informatica Global Customer Support.

Data Integration Service Process Properties

A service process is the physical representation of a service running on a node. When the Data Integration Service runs on multiple nodes, a Data Integration Service process can run on each node with the service role. You can configure the service process properties differently for each node.

To configure properties for the Data Integration Service processes, click the **Processes** view. Select a node to configure properties specific to that node.

The number of running service processes depends on the following ways that you can configure the Data Integration Service:

Single node

A single service process runs on the node.

Primary and back-up nodes

A service process is enabled on each node. However, only a single process runs at any given time, and the other processes maintain standby status.

Grid

A service process runs on each node in the grid that has the service role.

You can edit service process properties such as the HTTP port, result set cache, custom properties, and environment variables. You can change the properties while the Data Integration Service process is running, but you must restart the process for the changed properties to take effect.

Data Integration Service Security Properties

When you set the HTTP protocol type for the Data Integration Service to HTTPS or both, you enable the Transport Layer Security (TLS) protocol for the service. Depending on the HTTP protocol type of the service, you define the HTTP port, the HTTPS port, or both ports for the service process.

The following table describes the Data Integration Service Security properties:

Property	Description
HTTP Port	Unique HTTP port number for the Data Integration Service process when the service uses the HTTP protocol. Default is 8095.
HTTPS Port	Unique HTTPS port number for the Data Integration Service process when the service uses the HTTPS protocol. When you set an HTTPS port number, you must also define the keystore file that contains the required keys and certificates.

HTTP Configuration Properties

The HTTP configuration properties for a Data Integration Service process specify the maximum number of HTTP or HTTPS connections that can be made to the process. The properties also specify the keystore and truststore file to use when the Data Integration Service uses the HTTPS protocol.

The following table describes the HTTP configuration properties for a Data Integration Service process:

Property	Description
Maximum Concurrent Requests	Maximum number of HTTP or HTTPS connections that can be made to this Data Integration Service process. Default is 200.
Maximum Backlog Requests	Maximum number of HTTP or HTTPS connections that can wait in a queue for this Data Integration Service process. Default is 100.
Keystore File	Path and file name of the keystore file that contains the keys and certificates required if you use HTTPS connections for the Data Integration Service. You can create a keystore file with a keytool. keytool is a utility that generates and stores private or public key pairs and associated certificates in a keystore file. You can use the self-signed certificate or use a certificate signed by a certificate authority. If you run the Data Integration Service on a grid, the keystore file on each node in the grid must contain the same keys.
Keystore Password	Password for the keystore file.
Truststore File	Path and file name of the truststore file that contains authentication certificates trusted by the Data Integration Service. If you run the Data Integration Service on a grid, the truststore file on each node in the grid must contain the same keys.

Property	Description
Truststore Password	Password for the truststore file.
SSL Protocol	Secure Sockets Layer protocol to use. Default is TLS.

Result Set Cache Properties

The following table describes the result set cache properties:

Property	Description
Maximum Total Disk Size	Maximum number of bytes allowed for the total result set cache file storage. Default is 0.
Maximum Per Cache Memory Size	Maximum number of bytes allocated for a single result set cache instance in memory. Default is 0.
Maximum Total Memory Size	Maximum number of bytes allocated for the total result set cache storage in memory. Default is 0.
Maximum Number of Caches	Maximum number of result set cache instances allowed for this Data Integration Service process. Default is 0.

Advanced Properties

The following table describes the Advanced properties:

Property	Description
Maximum Heap Size	<p>Amount of RAM allocated to the Java Virtual Machine (JVM) that runs the Data Integration Service. Use this property to increase the performance. Append one of the following letters to the value to specify the units:</p> <ul style="list-style-type: none"> - b for bytes. - k for kilobytes. - m for megabytes. - g for gigabytes. <p>Default is 640 megabytes.</p> <p>Note: Consider increasing the heap size when the Data Integration Service needs to process large amounts of data.</p> <p>For example, if the Data Integration Service runs workflows that create many Human tasks, increase the heap size to 1024 megabytes.</p>
JVM Command Line Options	Java Virtual Machine (JVM) command line options to run Java-based programs. When you configure the JVM options, you must set the Java SDK classpath, Java SDK minimum memory, and Java SDK maximum memory properties.

Logging Options

The following table describes the logging options for the Data Integration Service process:

Property	Description
Log Directory	Directory for Data Integration Service node process logs. Default is <Informatica installation directory>/logs/node_name>/services/DataIntegrationService/. If the Data Integration Service runs on a grid, use a shared directory to create one directory for log files. Use a shared directory to ensure that if the master service process fails over to another node, the new master service process can access previous log files.

SQL Properties

The following table describes the SQL properties:

Property	Description
Maximum # of Concurrent Connections	Limits the number of database connections that the Data Integration Service can make for SQL data services. Default is 100.

Custom Properties for the Data Integration Service Process

Configure custom properties that are unique to specific environments.

You might need to apply custom properties in special cases. When you define a custom property, enter the property name and an initial value. Define custom properties only at the request of Informatica Global Customer Support.

Environment Variables

You can configure environment variables for the Data Integration Service process.

The following table describes the environment variables:

Property	Description
Environment Variable	Enter a name and a value for the environment variable.

Data Integration Service Compute Properties

You can configure the compute properties that the execution Data Transformation Manager (DTM) uses when it runs jobs.

When the Data Integration Service runs on primary and back-up nodes, you can configure the compute properties differently for each node. When the Data Integration Service runs on a grid, DTM instances run

jobs on each node with the compute role. You can configure the compute properties differently for each node with the compute role.

To configure compute properties for the DTM, click the **Compute** view. Select a node with the compute role to configure properties specific to DTM instances that run on the node.

You can change the compute properties while the Data Integration Service is running, but you must restart the service for the properties to take effect.

Execution Options

The default value for each execution option on the **Compute** view is defined by the same execution option on the **Properties** view. When the Data Integration Service runs on multiple nodes, you can override the execution options to define different values for each node with the compute role. The DTM instances that run on the node use the overridden values.

You can override the following execution options on the **Compute** view:

- Home Directory
- Temporary Directories
- Cache Directory
- Source Directory
- Target Directory
- Rejected Files Directory

When you override an execution option for a specific node, the Administrator tool displays a green checkmark next to the overridden property. The **Edit Execution Options** dialog box displays a reset option next to each overridden property. Select **Reset** to remove the overridden value and use the value defined for the Data Integration Service on the **Properties** view.

The following image shows that the **Temporary Directories** property has an overridden value in the **Edit Execution Options** dialog box:

Edit Execution Options [X]

Fields marked with an asterisk (*) are required.

Home Directory *	<input type="text" value="/"/>	
Temporary Directories *	<input type="text" value="/myTemp"/>	<input type="checkbox"/> Reset
Cache Directory *	<input type="text" value="/cache"/>	
Source Directory *	<input type="text" value="/source"/>	
Target Directory *	<input type="text" value="/target"/>	
Rejected Files Directory *	<input type="text" value="/reject"/>	

[?] [OK] [Cancel]

RELATED TOPICS:

- [“Execution Options” on page 56](#)
- [“Directories for Data Integration Service Files” on page 95](#)

Environment Variables

When a Data Integration Service grid runs jobs in separate remote processes, you can configure environment variables for DTM processes that run on nodes with the compute role.

Note: If the Data Integration Service runs on a single node or on a grid that runs jobs in the service process or in separate local processes, any environment variables that you define on the **Compute** view are ignored.

When a node in the grid has the compute role only, configure environment variables for DTM processes on the **Compute** view.

When a node in the grid has both the service and compute roles, you configure environment variables for the Data Integration Service process that runs on the node on the **Processes** view. You configure environment variables for DTM processes that run on the node on the **Compute** view. DTM processes inherit the environment variables defined for the Data Integration Service process. You can override an environment variable value for DTM processes. Or, you can define specific environment variables for DTM processes.

Consider the following examples:

- You define `EnvironmentVar1=A` on the **Processes** view and define `EnvironmentVar1=B` on the **Compute** view. The Data Integration Service process that runs on the node uses the value A for the environment variable. The DTM processes that run on the node use the value B.
- You define `EnvironmentVar1` on the **Processes** view and define `EnvironmentVar2` on the **Compute** view. The Data Integration Service process that runs on the node uses `EnvironmentVar1`. The DTM processes that run on the node use both `EnvironmentVar1` and `EnvironmentVar2`.

The following table describes the environment variables:

Property	Description
Environment Variable	Enter a name and a value for the environment variable.

Operating System Profiles for the Data Integration Service

An operating system profile is a type of security that the Data Integration Service uses to run mappings, workflows, and profiling jobs. Use operating system profiles to increase security and to isolate the run-time environment for users.

If the Data Integration Service runs on UNIX or Linux, create operating system profiles and configure the Data Integration Service to use operating system profiles.

By default, the Data Integration Service process runs all jobs, mappings, and workflows using the permissions of the operating system user that starts Informatica Services. The Data Integration Service writes output files to a single shared location specified in the Data Integration Service execution options.

When you configure the Data Integration Service to use operating system profiles, the Data Integration Service process runs jobs with the permission of the operating system user you define in the operating

system profile. The operating system profile contains the operating system user name, service process variables, Hadoop impersonation properties, the Analyst Service properties, environment variables, and permissions. The operating system user must have access to the directories you configure in the profile and the directories the Data Integration Service accesses at run time.

Note: If the Analyst Service and the Data Integration Service run on different nodes, the operating system profiles must be configured for both nodes.

Before you run a mapping with a Lookup transformation, Sqoop source, or Sqoop target in the Hadoop run-time environment, verify that the operating system user has read, write, and execute permissions on the following directory:

```
<Informatica installation directory>/tomcat/temp/<Data Integration Service name>/temp
```

Operating System Profile Example

An I.T. organization has some developers that work with sensitive data from Human Resources. The organization needs to restrict other developers in the organization from accessing any HR file or directory that the HR developers own.

The organization enables operating system profiles to limit access to data. Each developer group has an operating system profile. The developers in the HR operating system profile can read and write data in the restricted directories on the UNIX machine.

Operating System Profile Components

Configure the following components in an operating system profile:

- Operating system user name. Specify an operating system user that exists on the system where the Data Integration Service runs. The Data Integration Service uses the system permissions of this operating system user to run mappings, workflows, and profiling jobs.
- Service process variables. Configure service process variables in the operating system profile to specify different output file locations based on the operating system profile that is assigned to the user or group.
- Hadoop impersonation properties. Configure the Data Integration Service to use a Hadoop impersonation user to run mappings, workflows, and profiles in a Hadoop environment.
- Environment variables. Configure environment variables that the Data Integration Services uses at run time.
- Analyst service properties. Configure the flat file cache directory for the Analyst tool to store uploaded flat files.
- Permissions. Configure permissions for users and groups to use operating system profiles.

Configuring the Data Integration Service to Use Operating System Profiles

Configure the Data Integration Service to run mappings, workflows, and profiling jobs with operating system profiles.

The operating system user you define in the operating system profile must have access to the directories you configure in the operating system profile and to the directories the Data Integration Service accesses at run time. For example, pmsuid is a tool that the DTM process, command tasks, and parameter files use to switch between operating system users. You must provide permissions to operating system users to run pmsuid with the permissions of the Data Integration Service administrator user.

Note: If you enable the Data Integration Service to use operating system profiles, you cannot enable cache connection, the SQL Service Module, and the Web Service Module.

Complete the following steps to configure the Data Integration Service to use operating system profiles:

1. Configure system permissions on the files and directories that the operating system profile user needs access at run time.
2. In the Administrator tool, enable the Data Integration Service to use operating system profiles.
3. On the Security page of the Administrator tool, create operating system profiles.

For more information on creating and managing operating system profiles, see the *Informatica Security Guide*.

Configuring System Permissions for the Operating System Profile Users

Configure system permissions on the files and directories that operating system profile users must access at run time.

1. Make sure that the operating system user that starts the Informatica services has sudo permission.
2. On UNIX or Linux, verify that setuid is enabled on the file system that contains the Informatica installation.

If necessary, remount the file system with setuid enabled.

3. Make sure that all the library files in the following directory have at least 755 permissions:

```
<Informatica installation directory>/services/shared/bin
```

4. Make sure that the operating system profile users have 777 permissions on the \$DISTempDir directory and at least 750 permissions on the \$DISLogDir directory.
5. Make sure that the operating system profile users have at least 755 permissions to the directory where the pmsuid file is located and all its parent directories.

The pmsuid file is located in the following directory:

```
<Informatica installation directory>/services/shared/bin
```

6. Set the owner and group of pmsuid to root and set the permissions. Perform the following steps on each node where the Data Integration Service runs:

- a. At the command prompt, switch to the following directory:

```
<Informatica installation directory>/services/shared/bin
```

- b. Enter the following information at the command line to log in as root:

```
su root
```

- c. Enter the following command to create a group for the administrator user:

```
sudo groupadd <group name>
```

- d. Enter the following command to add the administrator user to the group:

```
sudo usermod -G <group name> <Informatica administrator user>
```

The administrator user is the Linux user whose permissions are used for all Informatica services.

- e. Enter the following command to change the owner and group of pmsuid to root and the group that you created:

```
chown root:<group name> pmsuid
```

- f. Set the following permissions:

```
chmod 6710 pmsuid
```

- g. Verify that the permissions for the pmsuid file appear as follows:

```
rws--s---
```

7. Set the umask value of the directories that the operating system profile accesses to 0027 or 0077 for better security.

When you create these directories on UNIX or Linux, the default umask value is set to 0222.

Enabling the Data Integration Service to Use Operating System Profiles

After you configure system permissions for the operating system profile users, enable the Data Integration Service to use operating system profiles.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the Data Integration Service.
3. In the **Properties** view of the Data Integration Service, click **Edit Execution Options**.
4. Select **Use Operating System Profiles and Impersonation**.
A warning message appears that cache connection, the SQL Service Module, and the Web Service Module are not available when the Data Integration Service uses operating system profiles.
5. Restart the Data Integration Service to apply the changes.

Troubleshooting Operating System Profiles

Consider the following troubleshooting tips when you configure the Data Integration Service to use operating system profiles:

After I configured the Data Integration Service to use operating system profiles, the Data Integration Service failed to start.

The Data Integration Service will not start if operating system profiles is enabled on Windows or a grid that includes a Windows node. You can enable operating system profiles on Data Integration Services that run on UNIX or Linux.

Or, *pmsuid* was not configured. To use operating system profiles, you must set the owner and group of *pmsuid* to administrator and enable the setuid bit for *pmsuid*.

High Availability for the Data Integration Service

High availability for the Data Integration Service minimizes interruptions to data integration tasks. High availability enables the Service Manager and the Data Integration Service to react to network failures and failures of the Data Integration Service.

The Data Integration Service has the following high availability features that are available based on your license:

Restart and Failover

When a Data Integration Service process becomes unavailable, the Service Manager tries to restart the process or fails the process over to another node based on the service configuration.

Recovery

When a Data Integration Service process shuts down unexpectedly, the Data Integration Service can automatically recover canceled workflow instances.

For information about configuring a highly available domain, see the *Informatica Administrator Guide*.

Data Integration Service Restart and Failover

When a Data Integration Service process becomes unavailable, the Service Manager restarts the Data Integration Service process on the same node or on a backup node.

The restart and failover behavior depends on the following ways that you can configure the Data Integration Service:

Single node

When the Data Integration Service runs on a single node and the service process shuts down unexpectedly, the Service Manager tries to restart the service process. If the Service Manager cannot restart the process, the process stops or fails.

Primary and backup nodes

When the Data Integration Service runs on primary and backup nodes and the service process shuts down unexpectedly, the Service Manager tries to restart the service process. If the Service Manager cannot restart the process, the Service Manager fails the service process over to a backup node.

A Data Integration Service process fails over to a backup node in the following situations:

- The Data Integration Service process fails and the primary node is not available.
- The Data Integration Service process is running on a node that fails.

Grid

When the Data Integration Service runs on a grid, the restart and failover behavior depends on whether the master or worker service process becomes unavailable.

If the master service process shuts down unexpectedly, the Service Manager tries to restart the process. If the Service Manager cannot restart the process, the Service Manager elects another node to run the master service process. The remaining worker service processes register themselves with the new master. The master service process then reconfigures the grid to run on one less node.

If a worker service process shuts down unexpectedly, the Service Manager tries to restart the process. If the Service Manager cannot restart the process, the master service process reconfigures the grid to run on one less node.

The Service Manager restarts the Data Integration Service process based on domain property values set for the amount of time spent trying to restart the service and the maximum number of attempts to try within the restart period.

The Data Integration Service clients are resilient to temporary connection failures during restart and failover of the service.

Data Integration Service Failover Configuration

When you configure the Data Integration Service to run on multiple nodes, verify that each node has access to the source and output files that the Data Integration Service requires to process data integration tasks such as workflows and mappings. For example, a workflow might require parameter files, input files, or output files.

To access logs for completed data integration tasks after a failover occurs, configure a shared directory for the Data Integration Service process **Logging Directory** property.

Data Integration Service Recovery

The Data Integration Service can recover some workflows that are enabled for recovery. Workflow recovery is the completion of a workflow instance from the point of interruption.

A running workflow instance can be interrupted when an error occurs, when you cancel the workflow instance, when you restart a Data Integration Service, or when a Data Integration Service process shuts down unexpectedly. If you abort the workflow instance, the instance is not recoverable.

The Data Integration Service performs workflow recovery based on the state of the workflow tasks, the values of the workflow variables and parameters during the interrupted workflow instance, and whether the recovery is manual or automatic.

Based on your license, you can configure automatic recovery of workflow instances. If you enable a workflow for automatic recovery, the Data Integration Service automatically recovers the workflow when the Data Integration Service restarts.

If the Data Integration Service runs on a grid and the master service process fails over, all nodes retrieve object state information from the Model repository. The new master automatically recovers workflow instances that were running during the failover and that are configured for automatic recovery.

The Data Integration Service does not automatically recover workflows that are not configured for automatic recovery. You can manually recover these workflows if they are enabled for recovery.

Any SQL data service, web service, mapping, profile, and preview jobs that were running during the failover are not recovered. You must manually restart these jobs.

CHAPTER 4

Data Integration Service Architecture

This chapter includes the following topics:

- [Data Integration Service Architecture Overview, 78](#)
- [Data Integration Service Connectivity, 79](#)
- [Data Integration Service Components, 80](#)
- [Service Components, 81](#)
- [Compute Component, 84](#)
- [Process Where DTM Instances Run, 87](#)
- [Single Node, 89](#)
- [Grid, 90](#)
- [Logs, 90](#)

Data Integration Service Architecture Overview

The Data Integration Service receives requests to run data transformation jobs from client tools. Data transformation jobs include mappings, previews, profiles, SQL queries to an SQL data service, web service operation mappings, and workflows. The Data Integration Service connects to other application services, databases, and third-party applications to access and transform the data.

To perform data transformation jobs, the Data Integration Service starts the following components:

Data Integration Service process

The Data Integration Service starts one or more Data Integration Service processes to manage requests to run jobs, application deployment, job optimizations, and data caches. Multiple service components run within a Data Integration Service process. Each service component performs a specific function to complete a data transformation job.

DTM instance

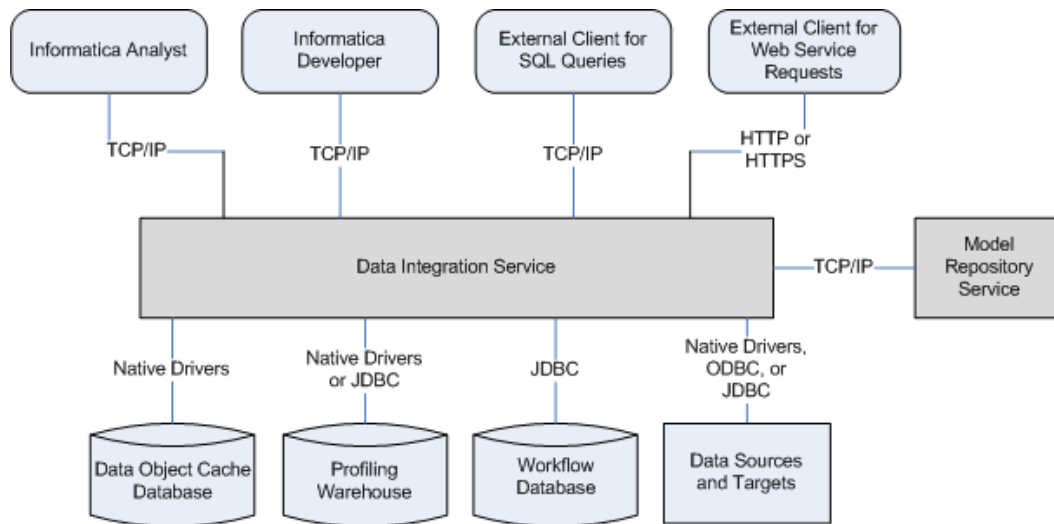
The Data Integration Service starts a DTM instance to run each job. A DTM instance is a specific, logical representation of the execution Data Transformation Manager (DTM). The DTM is the compute component of the Data Integration Service that runs jobs.

The Data Integration Service can run on a single node or on a grid. A grid is an alias assigned to a group of nodes that run jobs. When you run a job on a grid, you improve scalability and performance by distributing jobs to processes running on multiple nodes in the grid.

Data Integration Service Connectivity

The Data Integration Service uses multiple types of connectivity to communicate with client tools, other application services, databases, and applications.

The following image shows an overview of the types of connectivity that the Data Integration Service uses:



The Data Integration Service uses the following types of connectivity:

TCP/IP

The Data Integration Service uses TCP/IP network protocol to communicate with Informatica Analyst (the Analyst tool), Informatica Developer (the Developer tool), and external clients that send SQL queries. The Data Integration Service also uses TCP/IP to communicate with the Model Repository Service.

HTTP or HTTPS

The Data Integration Service uses HTTP or HTTPS to communicate with external clients that send web service requests.

Native drivers

The Data Integration Service uses native drivers to connect to the data object cache database. The Data Integration Service can also use native drivers to connect to the profiling warehouse or to a source or target database or application.

JDBC

The Data Integration Service uses JDBC to connect to the workflow database. The Data Integration Service can also use native JDBC drivers to connect to the profiling warehouse or to a source or target database or application.

ODBC

The Data Integration Service can use ODBC drivers to connect to a source or target database or application.

Data Integration Service Components

The Data Integration Service includes multiple components that complete data transformation jobs.

The Data Integration Service includes the following components:

Service components

Multiple service components run within the Data Integration Service process. The service components manage job requests, application deployment, job optimizations, and data caches. Service components include modules and managers.

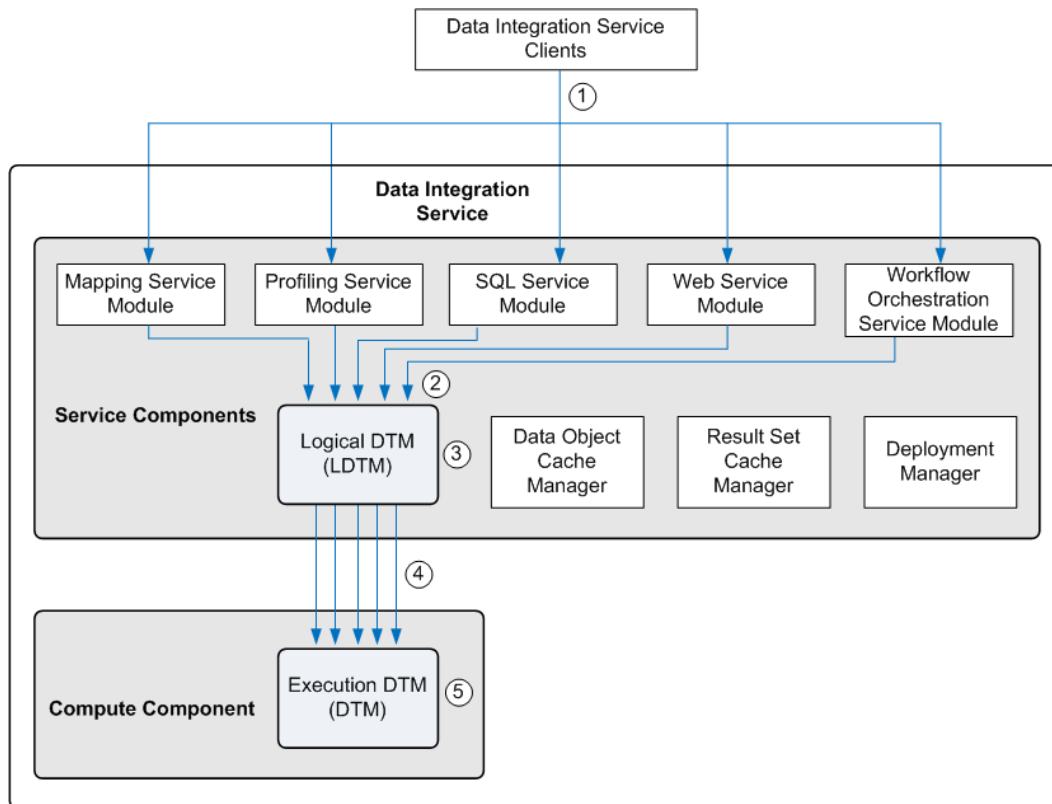
Modules manage the requests from client tools to run data transformation jobs. When a service module receives a request to run a job, the service module sends the job to the logical Data Transformation Manager (LDTM). The LDTM optimizes and compiles the job, and then sends the job to the execution Data Transformation Manager (DTM).

Managers manage application deployment, data caching, and temporary result set caches.

Compute component

The compute component is the execution Data Transformation Manager (DTM) that runs jobs. The DTM extracts, transforms, and loads data to complete a data transformation job such as a preview or mapping.

The following image shows how the Data Integration Service components complete job requests:



1. A Data Integration Service client sends a request to a service module to run a job.
2. The service module sends the job to the LDTM.
3. The LDTM optimizes and compiles the job.

4. The LDTM sends the compiled job to the DTM.
5. The DTM runs the job.

Service Components

The service components of the Data Integration Service include modules that manage requests from client tools. They also include managers that manage application deployment, caches, and job optimizations.

The service components run within the Data Integration Service process. The Data Integration Service process must run on a node with the service role. A node with the service role can run application services.

Mapping Service Module

The Mapping Service Module manages requests to preview data and run mappings.

The following table lists the requests that the Mapping Service Module manages from the different client tools:

Request	Client Tools
Preview source or transformation data based on mapping logic.	Developer tool Analyst tool
Run a mapping.	Developer tool
Run a mapping in a deployed application.	Command line
Preview an SQL data service.	Developer tool
Preview a web service operation mapping.	Developer tool

Sample third-party client tools include SQL Squirrel Client, DBClient, and MySQL ODBC Client.

When you preview or run a mapping, the client tool sends the request and the mapping to the Data Integration Service. The Mapping Service Module sends the mapping to the LDTM for optimization and compilation. The LDTM passes the compiled mapping to a DTM instance, which generates the preview data or runs the mapping.

When you preview data contained in an SQL data service in the Developer tool, the Developer tool sends the request to the Data Integration Service. The Mapping Service Module sends the SQL statement to the LDTM for optimization and compilation. The LDTM passes the compiled SQL statement to a DTM instance, which runs the SQL statement and generates the preview data.

When you preview a web service operation mapping in the Developer tool, the Developer tool sends the request to the Data Integration Service. The Mapping Service Module sends the operation mapping to the LDTM for optimization and compilation. The LDTM passes the compiled operation mapping to a DTM instance, which runs the operation mapping and generates the preview data.

Profiling Service Module

The Profiling Service Module manages requests to run profiles and generate scorecards.

When you run a profile in the Analyst tool or the Developer tool, the application sends the request to the Data Integration Service. The Profiling Service Module converts the profile into one or more mappings. The Profiling Service Module sends the mappings to the LDTM for optimization and compilation. The LDTM passes the compiled mappings to DTM instances that get the profiling rules and run the profile.

When you run a scorecard in the Analyst tool or the Developer tool, the application sends the request to the Data Integration Service. The Profiling Service Module converts the scorecard into one or more mappings. The Profiling Service Module sends the mappings to the LDTM for optimization and compilation. The LDTM passes the compiled mappings to DTM instances that generate a scorecard for the profile.

To create and run profiles and scorecards, you must associate the Data Integration Service with a profiling warehouse. The Profiling Service Module stores profiling data and metadata in the profiling warehouse.

SQL Service Module

The SQL Service Module manages SQL queries sent to an SQL data service from a third-party client tool.

When the Data Integration Service receives an SQL query from a third-party client tool, the SQL Service Module sends the SQL statement to the LDTM for optimization and compilation. The LDTM passes the compiled SQL statement to a DTM instance to run the SQL query against the virtual tables in the SQL data service.

If you do not cache the data when you deploy an SQL data service, a DTM instance is started to run the SQL data service. Every time the third-party client tool sends an SQL query to the virtual database, the DTM instance reads data from the source tables instead of cache tables.

Web Service Module

The Web Service Module manages web service operation requests sent to a web service from a web service client.

When the Data Integration Service receives requests from a web service client, the Web Service Module sends the web service operation mapping to the LDTM for optimization and compilation. The LDTM passes the compiled mapping to a DTM instance that runs the operation mapping. The Web Service Module sends the operation mapping response to the web service client.

Workflow Orchestration Service Module

The Workflow Orchestration Service Module manages requests to run workflows.

When you start a workflow instance in a deployed application, the Data Integration Service receives the request. The Workflow Orchestration Service Module runs and manages the workflow instance. The Workflow Orchestration Service Module runs workflow objects in the order that the objects are connected. The Workflow Orchestration Service Module evaluates expressions in conditional sequence flows to determine whether to run the next task. If the expression evaluates to true or if the sequence flow does not include a condition, the Workflow Orchestration Service Module starts and passes input data to the connected task. The task uses the input data to complete a single unit of work.

When a Mapping task runs a mapping, it sends the mapping to the LDTM for optimization and compilation. The LDTM passes the compiled mapping to a DTM instance to run the mapping.

When a task finishes processing a unit of work, the task passes output data back to the Workflow Orchestration Service Module. The Workflow Orchestration Service Module uses this data to evaluate expressions in conditional sequence flows or uses this data as input for the remaining tasks in the workflow.

Data Object Cache Manager

The Data Object Cache Manager caches data in an application.

When you enable data object caching, the Data Object Cache Manager can cache logical data objects and virtual tables in a database. The Data Object Cache Manager initially caches the data when you enable the application. Optimal performance for the cache depends on the speed and performance of the database.

By default, the Data Object Cache Manager manages the data object cache in the data object cache database. The Data Object Cache Manager creates the cache tables and refreshes the cache. It creates one table for each cached logical data object or virtual table in an application. Objects within an application share cache tables, but objects in different applications do not. If one data object is used in multiple applications, the Data Object Cache Manager creates a separate cache table for each instance of the data object.

Result Set Cache Manager

The Result Set Cache Manager manages cached results for SQL data service queries and web service requests. A result set cache is the result of a DTM instance that runs an SQL query against an SQL data service or a web service request against a web service operation.

When you enable result set caching, the Result Set Cache Manager creates in-memory caches to temporarily store the results of a DTM instance. If the Result Set Cache Manager requires more space than allocated, it stores the data in cache files. The Result Set Cache Manager caches the results for a specified time period. When an external client makes the same request before the cache expires, the Result Set Cache Manager returns the cached results. If a cache does not exist or has expired, the Data Integration Service starts a DTM instance to process the request and then it stores the cached the results.

When the Result Set Cache Manager stores the results by user, the Data Integration Service only returns cached results to the user that ran the SQL query or sent the web service request. The Result Set Cache Manager stores the result set cache for SQL data services by user. The Result Set Cache Manager stores the result set cache for web services by user when the web service uses WS-Security. The Result Set Cache Manager stores the cache by the user name that is provided in the username token of the web service request.

Deployment Manager

The Deployment Manager is the component in Data Integration Service that manages applications. When you deploy an application, the Deployment Manager manages the interaction between the Data Integration Service and the Model Repository Service.

The Deployment Manager starts and stops an application. The Deployment Manager validates the mappings, workflows, web services, and SQL data services in the application and their dependent objects when you deploy the application.

After validation, the Deployment Manager stores application run-time metadata in the Model repository. Run-time metadata includes information to run the mappings, workflows, web services, and SQL data services in the application.

The Deployment Manager creates a separate set of run-time metadata in the Model repository for each application. When the Data Integration Service runs application objects, the Deployment Manager retrieves the run-time metadata and makes it available to the DTM.

Logical Data Transformation Manager

The logical Data Transformation Manager (LDTM) optimizes and compiles jobs.

The LDTM can perform the following optimizations:

Filter data to reduce the number of rows to be processed.

The LDTM applies optimization methods to filter data and reduce the number of rows to be processed. For example, the LDTM can use early selection optimization to move a filter closer to the source. It can use pushdown optimization to push transformation logic to a database. It can use the cost-based optimization method to change the join processing order. When you develop a mapping, you can choose an optimizer level that determines which optimization methods the LDTM can apply to the mapping.

Determine the partitioning strategy to maximize parallel processing.

If you have the partitioning option, the Data Integration Service can maximize parallelism for mappings and profiles. The LDTM dynamically determines the optimal number of partitions for each pipeline stage and the best way to redistribute data across each partition point.

Determine the data movement mode to optimize processing of ASCII characters.

The LDTM determines whether to use the ASCII or Unicode data movement mode for mappings that read from a flat file or relational source. The LDTM determines the data movement mode based on the character sets that the mapping processes. When a mapping processes all ASCII data, the LDTM selects the ASCII mode. In ASCII mode, the Data Integration Service uses one byte to store each character, which can optimize mapping performance. In Unicode mode, the service uses two bytes for each character.

After optimizing a mapping, the LDTM compiles the optimized mapping and makes it available to the execution Data Transformation Manager (DTM) to run.

Compute Component

The compute component of the Data Integration Service is the execution Data Transformation Manager (DTM). The DTM extracts, transforms, and loads data to complete a data transformation job.

The DTM must run on a node with the compute role. A node with the compute role can perform computations requested by application services.

Execution Data Transformation Manager

The execution Data Transformation Manager (DTM) extracts, transforms, and loads data to run a data transformation job such as a preview or mapping.

When a service module in the Data Integration Service receives a request to run a job, the service module sends the request to the LDTM. The LDTM optimizes and compiles the job, and then sends the compiled job to the DTM. A DTM instance is started to run the job and complete the request.

A DTM instance is a specific, logical representation of the DTM. The Data Integration Service runs multiple instances of the DTM to complete multiple requests. For example, the Data Integration Service runs a separate instance of the DTM each time it receives a request from the Developer tool to preview a mapping.

The DTM completes the following types of jobs:

- Run or preview mappings.
- Run mappings in workflows.
- Preview transformations.
- Run or query SQL data services.
- Run web service operations.

- Run or preview data profiles.
- Generate scorecards.

DTM Resource Allocation Policy

The Data Transformation Manager resource allocation policy determines how to allocate the CPU resources for tasks. The DTM uses an on-demand resource allocation policy to allocate CPU resources.

When the DTM runs a mapping, it converts the mapping into a set of tasks such as:

- Initializing and deinitializing pipelines
- Reading data from source
- Transforming data
- Writing data to target

The DTM allocates CPU resources only when a DTM task needs a thread. When a task completes or if a task is idle, the task returns the thread to a thread pool. The DTM reuses the threads in the thread pool for other DTM tasks.

Processing Threads

When the DTM runs mappings, it uses reader, transformation, and writer pipelines that run in parallel to extract, transform, and load data.

The DTM separates a mapping into pipeline stages and uses one reader thread, one transformation stage, and one writer thread to process each stage. Each pipeline stage runs in one of the following threads:

- Reader thread that controls how the DTM extracts data from the source.
- Transformation thread that controls how the DTM processes data in the pipeline.
- Writer thread that controls how the DTM loads data to the target.

Because the pipeline contains three stages, the DTM can process three sets of rows concurrently and optimize mapping performance. For example, while the reader thread processes the third row set, the transformation thread processes the second row set, and the writer thread processes the first row set.

If you have the partitioning option, the Data Integration Service can maximize parallelism for mappings and profiles. When you maximize parallelism, the DTM separates a mapping into pipeline stages and uses multiple threads to process each stage.

Output Files

The DTM generates output files when it runs mappings, mappings included in a workflow, profiles, SQL queries to an SQL data service, or web service operation requests. Based on transformation cache settings and target types, the DTM can create cache, reject, target, and temporary files.

By default, the DTM stores output files in the directories defined by execution options for the Data Integration Service.

Data objects and transformations in the Developer tool use system parameters to access the values of these Data Integration Service directories. By default, the system parameters are assigned to flat file directory, cache file directory, and temporary file directory fields.

For example, when a developer creates an Aggregator transformation in the Developer tool, the CacheDir system parameter is the default value assigned to the cache directory field. The value of the CacheDir system parameter is defined in the **Cache Directory** property for the Data Integration Service. Developers can remove

the default system parameter and enter a different value for the cache directory. However, jobs fail to run if the Data Integration Service cannot access the directory.

In the Developer tool, developers can change the default system parameters to define different directories for each transformation or data object.

Cache Files

The DTM creates at least one cache file for each Aggregator, Joiner, Lookup, Rank, and Sorter transformation included in a mapping, profile, SQL data service, or web service operation mapping.

If the DTM cannot process a transformation in memory, it writes the overflow values to cache files. When the job completes, the DTM releases cache memory and usually deletes the cache files.

By default, the DTM stores cache files for Aggregator, Joiner, Lookup, and Rank transformations in the list of directories defined by the Cache Directory property for the Data Integration Service. The DTM creates index and data cache files. It names the index file `PM*.idx`, and the data file `PM*.dat`.

The DTM stores the cache files for Sorter transformations in the list of directories defined by the Temporary Directories property for the Data Integration Service. The DTM creates one sorter cache file.

Reject Files

The DTM creates a reject file for each target instance in a mapping or web service operation mapping. If the DTM cannot write a row to the target, the DTM writes the rejected row to the reject file. If the reject file does not contain any rejected rows, the DTM deletes the reject file when the job completes.

By default, the DTM stores reject files in the directory defined by the Rejected Files Directory property for the Data Integration Service. The DTM names reject files based on the name of the target data object. The default name for reject files is `<file_name>.bad`.

Target Files

If a mapping or web service operation mapping writes to a flat file target, the DTM creates the target file based on the configuration of the flat file data object.

By default, the DTM stores target files in the list of directories defined by the Target Directory property for the Data Integration Service. The DTM names target files based on the name of the target data object. The default name for target files is `<file_name>.out`.

Temporary Files

The DTM can create temporary files when it runs mappings, profiles, SQL queries, or web service operation mappings. When the jobs complete, the temporary files are usually deleted.

By default, the DTM stores temporary files in the list of directories defined by the Temporary Directories property for the Data Integration Service. The DTM also stores the cache files for Sorter transformations in the list of directories defined by the Temporary Directories property.

Process Where DTM Instances Run

Based on how you configure the Data Integration Service, DTM instances can run in the Data Integration Service process, in a separate DTM process on the local node, or in a separate DTM process on a remote node.

A DTM process is an operating system process that the Data Integration Service starts to run DTM instances. Multiple DTM instances can run within the Data Integration Service process or within the same DTM process.

The **Launch Job Options** property on the Data Integration Service determines where the service starts DTM instances. Configure the property based on whether the Data Integration Service runs on a single node or a grid and based on the types of jobs that the service runs.

The following table lists each process where DTM instances can run:

Process Where DTM Instances Run	Data Integration Service Configuration	Types of Jobs
In the Data Integration Service process	Single node or grid	<p>Any job that runs on a single node or on a grid where each node has both the service and compute roles.</p> <p>Jobs that run in the service process achieve better performance, but stability decreases.</p> <p>Recommendation: Run SQL data service and web service jobs in the service process to increase performance.</p>
In separate DTM processes on the local node	Single node or grid	<p>Any job that runs on a single node or on a grid where each node has both the service and compute roles.</p> <p>Run jobs in separate local processes to increase stability. Stability increases because an unexpected interruption to one job does not affect all other jobs.</p> <p>Recommendation: Run mapping, profile, and workflow jobs in separate processes. You can run SQL data service and web service jobs in separate local processes, but performance might decrease.</p>
In separate DTM processes on remote nodes	Grid	<p>Mapping, profile, and workflow jobs on a grid where nodes have a different combination of roles.</p> <p>Run jobs in separate remote processes to increase stability. Stability increases because an unexpected interruption to one job does not affect all other jobs. In addition, you can better use the resources available on each node in the grid. When a node has the compute role only, the node does not have to run the service process. The machine uses all available processing power to run mappings.</p> <p>Note: You cannot run SQL data service and web service jobs in separate DTM processes on remote nodes.</p>

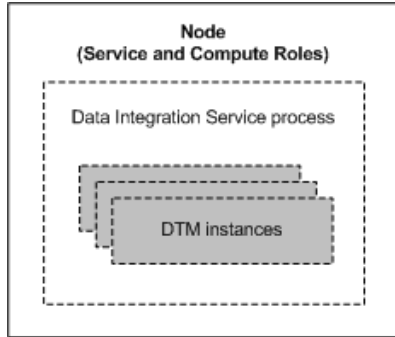
Note: Ad hoc jobs, with the exception of profiles, can run in the Data Integration Service process or in separate DTM processes on the local node. Ad hoc jobs include mappings run from the Developer tool or previews, scorecards, or drill downs on profile results run from the Developer tool or Analyst tool. If you configure a Data Integration Service grid to run jobs in separate remote processes, the service runs ad hoc jobs in separate local processes.

In the Data Integration Service Process

To run DTM instances in the Data Integration Service process, configure the Data Integration Service to launch jobs in the service process. Configure DTM instances to run in the Data Integration Service process when the service runs SQL data service and web service jobs on a single node or on a grid.

SQL data service and web service jobs typically achieve better performance when the Data Integration Service runs jobs in the service process.

The following image shows a Data Integration Service that runs DTM instances in the Data Integration Service process:

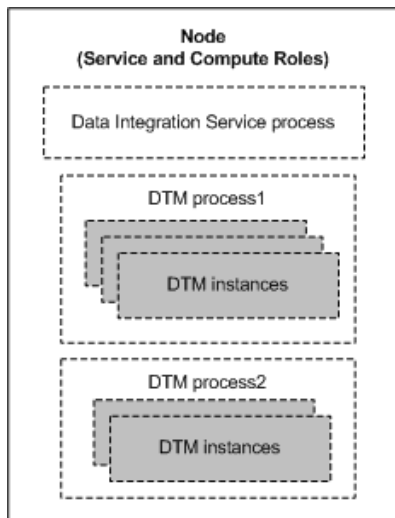


In Separate DTM Processes on the Local Node

To run DTM instances in separate DTM processes on the local node, configure the Data Integration Service to launch jobs in separate local processes. Configure DTM instances to run in separate DTM processes on the local node when the Data Integration Service runs mapping, profile, and workflow jobs on a single node or on a grid where each node has both the service and compute roles.

When the Data Integration Service runs jobs in separate local processes, stability increases because an unexpected interruption to one job does not affect all other jobs.

The following image shows a Data Integration Service that runs DTM instances in separate DTM processes on the local node:

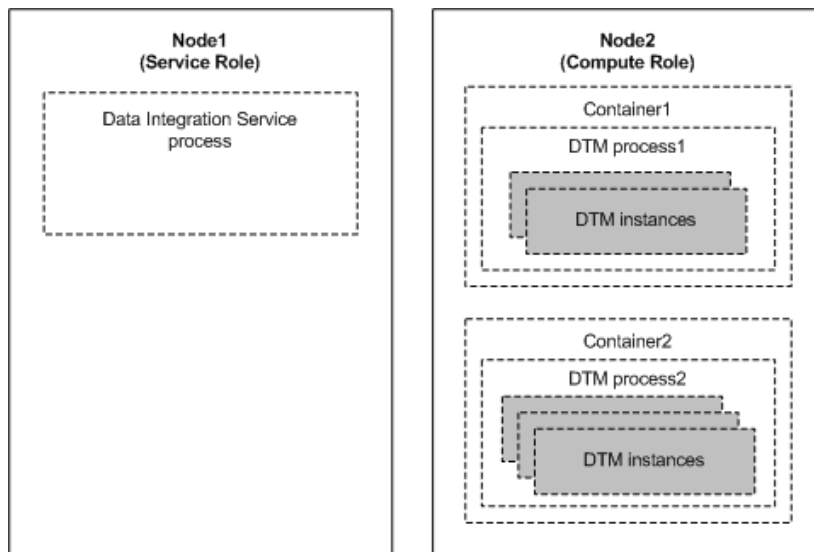


In Separate DTM Processes on Remote Nodes

To run DTM instances in separate DTM processes on remote nodes, configure the Data Integration Service to launch jobs in separate remote processes. Configure DTM instances to run in separate DTM processes on remote nodes when the Data Integration Service runs mapping, profile, and workflow jobs on a grid where nodes can have a different combination of roles.

When the Data Integration Service runs jobs in separate remote processes, stability increases because an unexpected interruption to one job does not affect all other jobs. In addition, you can better use the resources available on each node in the grid. When a node has the compute role only, the node does not have to run the service process. The machine uses all available processing power to run mappings.

The following image shows two of many nodes in a Data Integration Service grid. Node1 has the service role, and Node2 has the compute role. The Data Integration Service process on Node1 manages application deployments, logging, job requests, and job optimizations. The Service Manager on Node2 runs DTM instances in separate DTM processes started within containers.



Single Node

When the Data Integration Service runs on a single node, the service and compute components of the Data Integration Service run on the same node. The node must have both the service and compute roles.

A Data Integration Service that runs on a single node can run DTM instances in the Data Integration Service process or in separate DTM processes. Configure the service based on the types of jobs that the service runs.

If you run the Data Integration Service on a single node and you have the high availability option, you can configure back-up nodes in case the primary node becomes unavailable. High availability enables the Service Manager and the Data Integration Service to react to network failures and failures of the Data Integration Service. If a Data Integration Service becomes unavailable, the Service Manager can restart the service on the same node or on a back-up node.

Grid

If your license includes grid, you can configure the Data Integration Service to run on a grid. A grid is an alias assigned to a group of nodes that run jobs.

When the Data Integration Service runs on a grid, you improve scalability and performance by distributing jobs to processes running on multiple nodes in the grid. The Data Integration Service is also more resilient when it runs on a grid. If a service process shuts down unexpectedly, the Data Integration Service remains available as long as another service process runs on another node.

When the Data Integration Service runs on a grid, the service and compute components of the Data Integration Service can run on the same node or on different nodes, based on how you configure the grid and the node roles. Nodes in a Data Integration Service grid can have a combination of the service only role, the compute only role, and both the service and compute roles.

A Data Integration Service that runs on a grid can run DTM instances in the Data Integration Service process, in separate DTM processes on the same node, or in separate DTM processes on remote nodes. Configure the service based on the types of jobs that the service runs.

Logs

The Data Integration Service generates log events about service configuration and processing and about the jobs that the DTM runs.

The Data Integration Service generates the following types of log events:

Service log events

The Data Integration Service process generates log events about service configuration, processing, and failures. These log events are collected by the Log Manager in the domain. You can view the logs for the Data Integration Service on the Logs tab of the Administrator tool.

Job log events

The DTM generates log events about the jobs that it runs. The DTM generates log events for the following jobs:

- Previews, profiles, scorecards, or mappings run from the Analyst tool or the Developer tool
- Deployed mappings
- Logical data objects
- SQL data service queries
- Web service operation mappings
- Workflows

You can view the logs for these jobs on the Monitor tab of the Administrator tool.

When the DTM runs, it generates log events for the job that it is running. The DTM bypasses the Log Manager and sends the log events to log files. The DTM stores the log files in the Log Directory property specified for the Data Integration Service process. Log files have a `.log` file name extension.

If you created a custom location for logs before upgrading to the current version of Informatica, the Data Integration Service continues to write logs to that location after you upgrade. When you create a new Data Integration Service, the Data Integration Service writes logs to the default location unless you specify a different location.

When the Workflow Orchestration Service Module runs a workflow, it generates log events for the workflow. The Workflow Orchestration Service Module bypasses the Log Manager and sends the log events to log files. The Workflow Orchestration Service Module stores the log files in a folder named `workflow` in the log directory that you specify for the Data Integration Service process.

When a Mapping task in a workflow starts a DTM instance to run a mapping, the DTM generates log events for the mapping. The DTM stores the log files in a folder named `mappingtask` in the log directory that you specify for the Data Integration Service process.

CHAPTER 5

Data Integration Service Management

This chapter includes the following topics:

- [Data Integration Service Management Overview, 92](#)
- [Enable and Disable Data Integration Services and Processes, 93](#)
- [Directories for Data Integration Service Files, 95](#)
- [Run Jobs in Separate Processes, 98](#)
- [Maintain Connection Pools, 100](#)
- [PowerExchange Connection Pools, 103](#)
- [Maximize Parallelism for Mappings and Profiles, 106](#)
- [Result Set Caching, 111](#)
- [Data Object Caching, 111](#)
- [Persisting Virtual Data in Temporary Tables, 116](#)
- [Content Management for the Profiling Warehouse, 119](#)
- [Web Service Security Management, 124](#)
- [Pass-through Security, 125](#)

Data Integration Service Management Overview

After you create the Data Integration Service, use the Administrator tool to manage the service. When you change a service property, you must recycle the service or disable and then enable the service for the changes to take affect.

You can configure directories for the source, output, and log files that the Data Integration Service accesses when it runs jobs. When a Data Integration Service runs on multiple nodes, you might need to configure some of the directory properties to use a single shared directory.

You can optimize Data Integration Service performance by configuring the following features:

Run jobs in separate processes

You can configure the Data Integration Service to run jobs in separate DTM processes or in the Data Integration Service process. Running jobs in separate processes optimizes stability because an unexpected interruption to one job does not affect all other jobs.

Maintain connection pools

You can configure whether the Data Integration Service maintains connection pools for database connections when the service processes jobs. When you configure connection pooling, the Data Integration Service maintains and reuses a pool of database connections. Reusing connections optimizes performance because it minimizes the amount of time and resources used to open and close multiple database connections.

Maximize parallelism

If your license includes partitioning, you can enable the Data Integration Service to maximize parallelism when it runs mappings and profiles. When you maximize parallelism, the Data Integration Service dynamically divides the underlying data into partitions and processes all of the partitions concurrently. When the Data Integration Service adds partitions, it increases the number of processing threads, which can optimize mapping and profiling performance.

Cache result sets and data objects

You can configure the Data Integration Service to cache results for SQL data service queries and web service requests. You can also configure the service to use data object caching to access pre-built logical data objects and virtual tables. When the Data Integration Service caches result sets and data objects, subsequent jobs can take less time to run.

Persist virtual data in temporary tables

You can configure the Data Integration Service to persist virtual data in temporary tables. When business intelligence tools can retrieve data from the temporary table instead of the SQL data service, you can optimize SQL data service performance.

You can also manage content for the databases that the service accesses and configure security for SQL data service and web service requests to the Data Integration Service.

Enable and Disable Data Integration Services and Processes

You can enable and disable the entire Data Integration Service or a single Data Integration Service process on a particular node.

If you run the Data Integration Service on a grid or with the high availability option, you have one Data Integration Service process configured for each node. For a grid, the Data Integration Service runs all enabled Data Integration Service processes. For high availability, the Data Integration Service runs the Data Integration Service process on the primary node.

Enable, Disable, or Recycle the Data Integration Service

You can enable, disable, or recycle the Data Integration Service. You might disable the Data Integration Service if you need to perform maintenance or you need to temporarily restrict users from using the service. You might recycle the service if you changed a service property or if you updated the role for a node assigned to the service or to the grid on which the service runs.

The number of service processes that start when you enable the Data Integration Service depends on the following components which the service can run on:

Single node

When you enable a Data Integration Service that runs on a single node, a service process starts on the node.

Grid

When you enable a Data Integration Service that runs on a grid, a service process starts on each node in the grid that has the service role.

Primary and back-up nodes

When you enable a Data Integration Service configured to run on primary and back-up nodes, a service process is available to run on each node, but only the service process on the primary node starts. For example, you have the high availability option and you configure a Data Integration Service to run on a primary node and two back-up nodes. You enable the Data Integration Service, which enables a service process on each of the three nodes. A single process runs on the primary node, and the other processes on the back-up nodes maintain standby status.

Note: The associated Model Repository Service must be started before you can enable the Data Integration Service.

When you disable the Data Integration Service, you shut down the Data Integration Service and disable all service processes. If you are running the Data Integration Service on a grid, you disable all service processes on the grid.

When you disable the Data Integration Service, you must choose the mode to disable it in. You can choose one of the following options:

- **Complete.** Stops all applications and cancels all jobs within each application. Waits for all jobs to cancel before disabling the service.
- **Abort.** Stops all applications and tries to cancel all jobs before aborting them and disabling the service.

When you recycle the Data Integration Service, the Service Manager restarts the service. When the Service Manager restarts the Data Integration Service, it also restores the state of each application associated with the Data Integration Service.

Enabling, Disabling, or Recycling the Service

You can enable, disable, or recycle the service from the Administrator tool.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the service.
3. On the **Manage** tab **Actions** menu, click one of the following options:
 - **Enable Service** to enable the service.
 - **Disable Service** to disable the service.

Choose the mode to disable the service in. Optionally, you can choose to specify whether the action was planned or unplanned, and enter comments about the action. If you complete these options, the information appears in the **Events** and **Command History** panels in the **Domain** view on the **Manage** tab.

- **Recycle Service** to recycle the service.

Enable or Disable a Data Integration Service Process

You can enable or disable a Data Integration Service process on a particular node.

The impact on the Data Integration Service after you disable a service process depends on the following components which the service can run on:

Single node

When the Data Integration Service runs on a single node, disabling the service process disables the service.

Grid

When the Data Integration Service runs on a grid, disabling a service process does not disable the service. The service continues to run on other nodes that are designated to run the service, as long as the nodes are available.

Primary and back-up nodes

When you have the high availability option and you configure the Data Integration Service to run on primary and back-up nodes, disabling a service process does not disable the service. Disabling a service process that is running causes the service to fail over to another node.

Enabling or Disabling a Service Process

You can enable or disable a service process from the Administrator tool.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the service.
3. In the contents panel, click the **Processes** view.
4. On the **Manage** tab **Actions** menu, click one of the following options:
 - **Enable Process** to enable the service process.
 - **Disable Process** to disable the service process. Choose the mode to disable the service process in.

Directories for Data Integration Service Files

The Data Integration Service accesses file directories when it reads source files, reads control files, writes output files, and writes log files.

When the Data Integration Service runs on multiple nodes, you might need to configure some of the directory properties to use a single shared directory to ensure that the processes running on each node can access all files.

When the Data Integration Service uses operating system profiles, the operating system user specified in the profile must have access to the directories that the Data Integration Service accesses at run time.

Source and Output File Directories

Configure the directories for source and output files in the Execution Options on the **Properties** view for the Data Integration Service.

The Data Integration Service accesses source files when it runs a mapping or web service operation mapping that reads from a flat file source. The service generates output files when it runs mappings, mappings

included in a workflow, profiles, SQL queries to an SQL data service, or web service operation requests. Based on transformation cache settings and target types, the Data Integration Service can generate cache, reject, target, and temporary files.

When you configure directories for the source and output files, you configure the paths for the home directory and its subdirectories. The default value of the **Home Directory** property is `<Informatica installation directory>/tomcat/bin`. If you change the default value, verify that the directory exists.

By default, the following directories have values relative to the home directory:

- Temporary directories
- Cache directory
- Source directory
- Target directory
- Rejected files directory

You can define a different directory relative to the home directory. Or, you can define an absolute directory outside the home directory.

If you define a different absolute directory, use the correct syntax for the operating system:

- On Windows, enter an absolute path beginning with a drive letter, colon, and backslash. For example:

```
C:\<Informatica installation directory>\tomcat\bin\MyHomeDir
```

- On UNIX, enter an absolute path beginning with a slash. For example:

```
/<Informatica installation directory>/tomcat/bin/MyHomeDir
```

Data objects and transformations in the Developer tool use system parameters to access the values of these Data Integration Service directories. By default, the system parameters are assigned to flat file directory, cache file directory, and temporary file directory fields.

For example, when a developer creates an Aggregator transformation in the Developer tool, the CacheDir system parameter is the default value assigned to the cache directory field. The value of the CacheDir system parameter is defined in the **Cache Directory** property for the Data Integration Service. Developers can remove the default system parameter and enter a different value for the cache directory. However, jobs fail to run if the Data Integration Service cannot access the directory.

Configure Source and Output File Directories for Multiple Nodes

When the Data Integration Service runs on primary and back-up nodes or on a grid, DTM instances can run jobs on each node with the compute role. Each DTM instance must be able to access the source and output file directories. To run mappings that manage metadata changes in flat file sources, each Data Integration Service process must be able to access the source file directories.

When you configure the source and output file directories for a Data Integration Service that runs on multiple nodes, consider the following guidelines:

- You can configure the **Source Directory** property to use a shared directory to create one directory for source files.

If you run mappings that manage metadata changes in flat file sources and if the Data Integration Service grid is configured to run jobs in separate remote processes, you must configure the **Source Directory** property to use a shared directory.

If you run other types of mappings or if you run mappings that manage metadata changes in flat file sources on any other Data Integration Service grid configuration, you can configure different source directories for each node with the compute role. Replicate all source files in all of the source directories.

- If you run mappings that use a persistent lookup cache, you must configure the **Cache Directory** property to use a shared directory. If no mappings use a persistent lookup cache, you can configure the cache directory to have a different directory for each node with the compute role.
- You can configure the **Target Directory**, **Temporary Directories**, and **Reject File Directory** properties to have different directories for each node with the compute role.

To configure a shared directory, configure the directory in the Execution Options on the **Properties** view. You can configure a shared directory for the home directory so that all source and output file directories use the same shared home directory. Or, you can configure a shared directory for a specific source or output file directory. Remove any overridden values for the same execution option on the **Compute** view.

To configure different directories for each node with the compute role, configure the directory in the Execution Options on the **Compute** view.

Control File Directories

The Data Integration Service accesses control files when it runs mappings that generate columns for flat file sources based on control files. When the Data Integration Service runs the mapping, it fetches metadata from the control file of the flat file source.

Use the Developer tool to configure the control file directory for each flat file data object that is configured to generate run-time column names from a control file. You cannot use the Administrator tool to configure a single control file directory used by the Data Integration Service.

Configure Control File Directories for Multiple Nodes

When the Data Integration Service runs on primary and back-up nodes or on a grid, Data Integration Service processes can run on each node with the service role. Each Data Integration Service process must be able to access the control file directories.

Use the Developer tool to configure the **Control File Directory** property for each flat file data object that is configured to generate run-time column names from a control file. Configure the **Control File Directory** property in the **Advanced** properties for the flat file data object. Find the property in the **Runtime: Read** section.

When the Data Integration Service runs on multiple nodes, use one of the following methods to ensure that each Data Integration Service process can access the directories:

- Configure the **Control File Directory** property for each flat file data object to use a shared directory to create one directory for control files.
- Configure the **Control File Directory** property for each flat file data object to use an identical directory path that is local to each node with the service role. Replicate all control files in the identical directory on each node with the service role.

Log Directory

Configure the directory for log files on the **Processes** view for the Data Integration Service. Data Integration Service log files include files that contain service log events and files that contain job log events.

By default, the log directory for each Data Integration Service process is within the Informatica installation directory on the node.

Configure the Log Directory for Multiple Nodes

When the Data Integration Service runs on primary and back-up nodes or on a grid, a Data Integration Service process can run on each node with the service role. Configure each service process to use the same shared directory for log files.

When you configure a shared log directory, you ensure that if the master service process fails over to another node, the new master service process can access previous log files.

Configure each service process with identical absolute paths to the shared directories. If you use a mapped or mounted drive, the absolute path to the shared location must also be identical.

For example, a newly elected master service process cannot access previous log files when nodes use the following drives for the log directory:

- **Mapped drive on node1:** `F:\shared\<Informatica installation directory>\logs\<node_name>\services\DataIntegrationService\disLogs`
- **Mapped drive on node2:** `G:\shared\<Informatica installation directory>\logs\<node_name>\services\DataIntegrationService\disLogs`

A newly elected master service process also cannot access previous log files when nodes use the following drives for the log directory:

- **Mounted drive on node1:** `/mnt/shared/<Informatica installation directory>/logs/<node_name>/services/DataIntegrationService/disLogs`
- **Mounted drive on node2:** `/mnt/shared_filesystem/<Informatica installation directory>/logs/<node_name>/services/DataIntegrationService/disLogs`

Output and Log File Permissions

When a Data Integration Service process generates output or log files, it sets file permissions based on the operating system.

When a Data Integration Service process on UNIX generates an output or log file, it sets the file permissions according to the umask of the shell that starts the Data Integration Service process. For example, when the umask of the shell that starts the Data Integration Service process is 022, the Data Integration Service process creates files with `rw-r--r--` permissions. To change the file permissions, you must change the umask of the shell that starts the Data Integration Service process and then restart it.

A Data Integration Service process on Windows generates output and log files with read and write permissions.

Run Jobs in Separate Processes

The Data Integration Service can run jobs in the Data Integration Service process or in separate DTM processes on local or remote nodes. You optimize service performance when you configure the recommended option based on the job types that the service runs.

When the Data Integration Service receives a request to run a job, the service creates a DTM instance to run the job. A DTM instance is a specific, logical representation of the execution Data Transformation Manager. You can configure the Data Integration Service to run DTM instances in the Data Integration Service process, in a separate DTM process on the local node, or in a separate DTM process on a remote node.

A DTM process is an operating system process started to run DTM instances. Multiple DTM instances can run within the Data Integration Service process or within the same DTM process.

The **Launch Job Options** property on the Data Integration Service determines where the service starts DTM instances. Configure the property based on whether the Data Integration Service runs on a single node or a grid and based on the types of jobs that the service runs.

Choose one of the following options for the **Launch Job Options** property:

In the service process

Configure when you run jobs on a single node or on a grid where each node has both the service and compute roles.

SQL data service and web service jobs typically achieve better performance when the Data Integration Service runs jobs in the service process.

In separate local processes

Configure when you run jobs on a single node or on a grid where each node has both the service and compute roles. You can run SQL data service and web service jobs in separate local processes, but performance might decrease.

Configure when the Data Integration Service uses operating system profiles.

When the Data Integration Service runs jobs in separate local processes, stability increases because an unexpected interruption to one job does not affect all other jobs.

In separate remote processes

Configure when you run mapping, profile, and workflow jobs on a grid where nodes have a different combination of roles. If you choose this option when the Data Integration Service runs on a single node, then the service runs jobs in separate local processes. You cannot run SQL data service or web service jobs in separate remote processes.

When the Data Integration Service runs jobs in separate remote processes, stability increases because an unexpected interruption to one job does not affect all other jobs. In addition, you can better use the resources available on each node in the grid. When a node has the compute role only, the node does not have to run the service process. The machine uses all available processing power to run mappings.

Note: If you run multiple job types, create multiple Data Integration Services. Configure one Data Integration Service to run SQL data service and web service jobs in the Data Integration Service process. Configure the other Data Integration Service to run mappings, profiles, and workflows in separate local processes or in separate remote processes.

RELATED TOPICS:

- [“Process Where DTM Instances Run” on page 87](#)

DTM Process Pool Management

When the Data Integration Service runs jobs in separate local or remote processes, the Data Integration Service maintains a pool of reusable DTM processes.

The DTM process pool includes DTM processes that are running jobs and DTM processes that are idle. Each running DTM process in the pool is reserved for use by one of the following groups of related jobs:

- Jobs from the same deployed application
- Preview jobs
- Profiling jobs
- Mapping jobs run from the Developer tool

For example, if you run two jobs from the same deployed application, two DTM instances are created in the same DTM process. If you run a preview job, the DTM instance is created in a different DTM process.

When a DTM process finishes running a job, the process closes the DTM instance. When the DTM process finishes running all jobs, the DTM process is released to the pool as an idle DTM process. An idle DTM process is available to run any type of job.

Rules and Guidelines when Jobs Run in Separate Processes

Consider the following rules and guidelines when you configure the Data Integration Service to run jobs in separate local or remote processes:

- You cannot use the **Maximum Memory Size** property for the Data Integration Service to limit the amount of memory that the service allocates to run jobs. If you set the maximum memory size, the Data Integration Service ignores it.
- If the Data Integration Service runs on UNIX, the host file on each node with the compute role and on each node with both the service and compute roles must contain a localhost entry. If the host file does not contain a localhost entry, jobs that run in separate processes fail. Windows does not require a localhost entry in the host file.
- If you configure connection pooling, each DTM process maintains its own connection pool library. All DTM instances running in the DTM process can use the connection pool library. The number of connection pool libraries depends on the number of running DTM processes.

Maintain Connection Pools

Connection pooling is a framework to cache database connection information that is used by the Data Integration Service. Connection pools increase performance through the reuse of cached connection information.

A connection pool is a group of connection instances for one connection object. A connection instance is a representation of a physical connection to a data source. A connection pool library can contain multiple connection pools. The number of connection pools depends on the number of unique connections that the DTM instances use while running jobs.

You configure the Data Integration Service to run DTM instances in the Data Integration Service process or in separate DTM processes that run on local or remote nodes. Each Data Integration Service process or DTM process maintains its own connection pool library that all DTM instances running in the process can use. The number of connection pool libraries depends on the number of running Data Integration Service processes or DTM processes.

A connection instance can be active or idle. An active connection instance is a connection instance that a DTM instance is using to connect to a database. A DTM process or the Data Integration Service process can create an unlimited number of active connection instances.

An idle connection instance is a connection instance in a connection pool that is not in use. A connection pool retains idle connection instances based on the pooling properties that you configure for a database connection. You configure the minimum connections, the maximum connections, and the maximum idle connection time.

Connection Pool Management

When a DTM process or the Data Integration Service process runs a job, it requests a connection instance from the pool. If an idle connection instance exists, the connection pool releases it to the DTM process or the Data Integration Service process. If the connection pool does not have an idle connection instance, the DTM process or the Data Integration Service process creates an active connection instance.

When the DTM process or the Data Integration Service process completes the job, it releases the active connection instance to the pool as an idle connection instance. If the connection pool contains the maximum number of idle connection instances, the process drops the active connection instance instead of releasing it to the pool.

The DTM process or the Data Integration Service process drops an idle connection instance from the pool when the following conditions are true:

- A connection instance reaches the maximum idle time.
- The connection pool exceeds the minimum number of idle connections.

When you update the user name, password, or connection string for a database connection that has connection pooling enabled, the updates take effect immediately. Subsequent connection requests use the updated information. Also, the connection pool library drops all idle connections and restarts the connection pool. It does not return any connection instances that are active at the time of the restart to the connection pool when complete.

If you update any other database connection property, you must restart the Data Integration Service to apply the updates.

Pooling Properties in Connection Objects

You can edit connection pooling properties in the **Pooling** view for a database connection.

The number of connection pool libraries depends on the number of running Data Integration Service processes or DTM processes. Each Data Integration Service process or DTM process maintains its own connection pool library. The values of the pooling properties are for each connection pool library.

For example, if you set maximum connections to 15, then each connection pool library can have a maximum of 15 idle connections in the pool. If the Data Integration Service runs jobs in separate local processes and three DTM processes are running, then you can have a maximum of 45 idle connection instances.

To decrease the total number of idle connection instances, set the minimum number of connections to 0 and decrease the maximum idle time for each database connection.

The following list describes database connection pooling properties that you can edit in the **Pooling** view for a database connection:

Enable Connection Pooling

Enables connection pooling. When you enable connection pooling, each connection pool retains idle connection instances in memory. To delete the pools of idle connections, you must restart the Data Integration Service.

If connection pooling is disabled, the DTM process or the Data Integration Service process stops all pooling activity. The DTM process or the Data Integration Service process creates a connection instance each time it processes a job. It drops the instance when it finishes processing the job.

Default is enabled for DB2 for i5/OS, DB2 for z/OS, IBM DB2, Microsoft SQL Server, Oracle, and ODBC connections. Default is disabled for Adabas, IMS, Sequential, and VSAM connections.

Minimum # of Connections

The minimum number of idle connection instances that a pool maintains for a database connection after the maximum idle time is met. Set this value to be equal to or less than the maximum number of idle connection instances. Default is 0.

Maximum # of Connections

The maximum number of idle connection instances that a pool maintains for a database connection before the maximum idle time is met. Set this value to be more than the minimum number of idle connection instances. Default is 15.

Maximum Idle Time

The number of seconds that a connection instance that exceeds the minimum number of connection instances can remain idle before the connection pool drops it. The connection pool ignores the idle time when the connection instance does not exceed the minimum number of idle connection instances. Default is 120.

Example of a Connection Pool

You want to use connection pools to optimize connection performance. You have configured the Data Integration Service to run jobs in separate local processes.

You configure the following pooling properties for a connection:

- Connection Pooling: Enabled
- Minimum Connections: 2
- Maximum Connections: 4
- Maximum Idle Time: 120 seconds

When a DTM process runs five jobs, it uses the following process to maintain the connection pool:

1. The DTM process receives a request to process five jobs at 11:00 a.m., and it creates five connection instances.
2. The DTM process completes processing at 11:30 a.m., and it releases four connections to the connection pool as idle connections.
3. It drops one connection because it exceeds the connection pool size.
4. At 11:32 a.m., the maximum idle time is met for the idle connections, and the DTM process drops two idle connections.
5. The DTM process maintains two idle connections because the minimum connection pool size is two.

Optimize Connection Performance

To optimize connection performance, configure connection pooling for the database connections. Each DTM process or the Data Integration Service process caches database connections for jobs and maintains a pool of connections that it can reuse.

The DTM process or the Data Integration Service process caches and releases the connections based on how you configure connection pooling properties for the connection. Reusing connections optimizes performance. It minimizes the amount of time and resources that the DTM process or the Data Integration Service process uses when it opens and closes multiple database connections.

To optimize connection performance, enable the **Connection Pooling** property in the database connection properties. Optionally, configure additional connection pooling properties.

PowerExchange Connection Pools

A PowerExchange® connection pool is a set of network connections to a PowerExchange Listener. The Data Integration Service connects to a PowerExchange data source through the PowerExchange Listener.

PowerExchange uses connection pools for the following types of database connection objects:

- Adabas
- DB2 for i5/OS
- DB2 for z/OS
- IMS
- Sequential
- VSAM

To define a connection to a PowerExchange Listener, include a NODE statement in the DBMOVER file on the Data Integration Service machine. Then define a database connection and associate the connection with the Listener. The **Location** property specifies the Listener node name. Define database connection pooling properties in the **Pooling** view for a database connection.

PowerExchange Connection Pool Management

The Data Integration Service connects to a PowerExchange data source through the PowerExchange Listener. A PowerExchange connection pool is a set of connections to a PowerExchange Listener.

When a DTM process or the Data Integration Service process runs a data transformation job, it requests a connection instance from a connection pool. If the DTM process or the Data Integration Service process requires a PowerExchange connection instance, it requests the connection instance from PowerExchange.

When PowerExchange receives a request for a connection to a Listener, it uses a connection in the pool that has matching characteristics, including user ID and password. If the pool does not contain a connection with matching characteristics, PowerExchange modifies and reuses a pooled connection to the Listener, if possible. For example, if PowerExchange receives a request for a connection for USER1 on NODE1 and finds only a pooled connection for USER2 on NODE1, PowerExchange reuses the connection, signs off USER2, and signs on USER1.

When PowerExchange returns a Listener connection to the pool, it closes any files or databases that the Listener had opened.

If you associate multiple database connection objects with the same Listener node name, PowerExchange combines the connections into a single pool. For example, if you associate multiple database connections to NODE1, a connection pool is used for all PowerExchange connections to NODE1. To determine the maximum size of the connection pool for the Listener, PowerExchange adds the **Maximum # of Connections** values that you specify for each database connection that uses the Listener.

If you want each database connection object to use a separate connection pool, define multiple NODE statements for the same PowerExchange Listener and associate each database connection object with a different Listener node name.

Note: PowerExchange connection pooling does not reuse netport connections unless the user name and password match.

Connection Pooling for PowerExchange Netport Jobs

Netport jobs that use connection pooling might result in constraint issues.

Depending on the data source, the netport JCL might reference a data set or other resource exclusively. Because a pooled netport connection can persist for some time after the data processing has finished, you might encounter concurrency issues. If you cannot change the netport JCL to reference resources nonexclusively, consider disabling connection pooling.

In particular, IMS netport jobs that use connection pooling might result in constraint issues. Because the program specification block (PSB) is scheduled for a longer period of time when netport connections are pooled, resource constraints can occur in the following cases:

- A netport job on another port might try to read a separate database in the same PSB, but the scheduling limit is reached.
- The netport runs as a DL/1 job, and you attempt to restart the database within the IMS/DC environment after the mapping finishes running. The database restart fails, because the database is still allocated to the netport DL/1 region.
- Processing in a second mapping or a z/OS job flow relies on the database being available when the first mapping has finished running. If pooling is enabled, there is no guarantee that the database is available.
- You might need to build a PSB that includes multiple IMS databases that the Data Integration Service accesses. In this case, resource constraint issues are more severe as netport jobs are pooled that tie up multiple IMS databases for long periods.

This requirement might apply because you can include up to ten NETPORT statements in a DBMOVER file. Also, PowerExchange data maps cannot include program communication block (PCB) and PSB values that PowerExchange can use dynamically.

PowerExchange Connection Pooling Configuration

To configure PowerExchange connection pooling, include statements in the DBMOVER configuration files on each machine that hosts the PowerExchange Listener or the Data Integration Service. Also, define connection pooling properties in the **Pooling** view of the connection.

DBMOVER Configuration Statements for PowerExchange Connection Pooling

To configure PowerExchange connection pooling, define DBMOVER configuration statements on each machine that hosts the PowerExchange Listener or the Data Integration Service.

Define the following statements:

LISTENER

Defines the TCP/IP port on which a named PowerExchange Listener process listens for work requests. Include the LISTENER statement in the DBMOVER configuration file on the PowerExchange Listener machine.

MAXTASKS

Defines the maximum number of tasks that can run concurrently in a PowerExchange Listener. Include the MAXTASKS statement in the DBMOVER configuration file on the PowerExchange Listener machine.

Ensure that MAXTASKS is large enough to accommodate twice the maximum size of the connection pool for the Listener. The maximum size of the connection pool is equal to the sum of the values that you enter for the **Maximum # of Connections** pooling property for each database connection that is associated with the Listener.

Default is 30.

NODE

Defines the TCP/IP host name and port that PowerExchange uses to contact a PowerExchange Listener. Include the NODE statement in the DBMOVER file on the Data Integration Service machine.

TCPIP_SHOW_POOLING

Writes diagnostic information to the PowerExchange log file. Include the TCPIP_SHOW_POOLING statement in the DBMOVER file on the Data Integration Service machine.

If TCPIP_SHOW_POOLING=Y, PowerExchange writes message PWX-33805 to the PowerExchange log file each time a connection is returned to a PowerExchange connection pool.

Message PWX-33805 provides the following information:

- Size. Total size of PowerExchange connection pools.
- Hits. Number of times that PowerExchange found a connection in a PowerExchange connection pool that it could reuse.
- Partial hits. Number of times that PowerExchange found a connection in a PowerExchange connection pool that it could modify and reuse.
- Misses. Number of times that PowerExchange could not find a connection in a PowerExchange connection pool that it could reuse.
- Expired. Number of connections that were discarded from a PowerExchange connection pool because the maximum idle time was exceeded.
- Discarded pool full. Number of connections that were discarded from a PowerExchange connection pool because the pool was full.
- Discarded error. Number of connections that were discarded from a PowerExchange connection pool due to an error condition.

Pooling Properties in PowerExchange Connection Objects

Configure connection pooling properties in the **Pooling** view for a PowerExchange database connection.

Enable Connection Pooling

Enables connection pooling. When you enable connection pooling, each connection pool retains idle PowerExchange Listener connection instances in memory. When you disable connection pooling, the DTM process or the Data Integration Service process stops all pooling activity. To delete the pool of idle connections, you must restart the Data Integration Service.

Default is enabled for DB2 for i5/OS and DB2 for z/OS connections. Default is disabled for Adabas, IMS, Sequential, and VSAM connections.

Minimum # of Connections

The minimum number of idle connection instances that a pool maintains for a database connection after the maximum idle time is met. If multiple database connections are associated with a PowerExchange Listener, PowerExchange determines the minimum number of connections to the PowerExchange Listener by adding the values for each database connection.

Maximum # of Connections

The maximum number of idle connection instances that a pool maintains for a database connection before the maximum idle time is met. If multiple database connections are associated with a PowerExchange Listener, PowerExchange determines the maximum number of connections to the PowerExchange Listener node by adding the values for each database connection.

Verify that the value of MAXTASKS in the DBMOVER configuration file is large enough to accommodate twice the maximum number of connections to the PowerExchange Listener node.

Enter 0 to specify an unlimited connection pool size.

Default is 15.

Maximum Idle Time

The number of seconds that a connection instance that exceeds the minimum number of connection instances can remain idle before the connection pool drops it. The connection pool ignores the idle time when the connection instance does not exceed the minimum number of idle connection instances.

If multiple database connections are associated with a PowerExchange Listener, PowerExchange calculates the arithmetic mean of the non-zero values for each database connection to determine the maximum idle time for connections to the same Listener.

Default is 120.

Tip: Assign the same maximum idle time to each database connection.

Maximize Parallelism for Mappings and Profiles

If you have the partitioning option, you can enable the Data Integration Service to maximize parallelism when it runs mappings, runs column profiles, or performs data domain discovery. When you maximize parallelism, the Data Integration Service dynamically divides the underlying data into partitions and processes all of the partitions concurrently.

Note: When you run a profile job, the Data Integration Service converts the profile job into one or more mappings, and then can run those mappings in multiple partitions.

If mappings process large data sets or contain transformations that perform complicated calculations, the mappings can take a long time to process and can cause low data throughput. When you enable partitioning for these mappings, the Data Integration Service uses additional threads to process the mapping. Increasing the number of processing threads increases the load on the node where the mapping runs. If the node contains sufficient CPU bandwidth, concurrently processing rows of data in a mapping can optimize mapping performance.

By default, the **Maximum Parallelism** property is set to 1 for the Data Integration Service. When the Data Integration Service runs a mapping, it separates the mapping into pipeline stages and uses one thread to process each stage. These threads are allocated to reading, transforming, and writing tasks, and they run in parallel.

When you increase the maximum parallelism value, you enable partitioning. The Data Integration Service uses multiple threads to process each pipeline stage.

The Data Integration Service can create partitions for mappings that have physical data as input and output. The Data Integration Service can use multiple partitions to complete the following actions during a mapping run:

- Read from flat file, IBM DB2 for LUW, or Oracle sources.
- Run transformations.
- Write to flat file, IBM DB2 for LUW, or Oracle targets.

One Thread for Each Pipeline Stage

When maximum parallelism is set to 1, partitioning is disabled. The Data Integration Service separates a mapping into pipeline stages and uses one reader thread, one transformation thread, and one writer thread to process each stage.

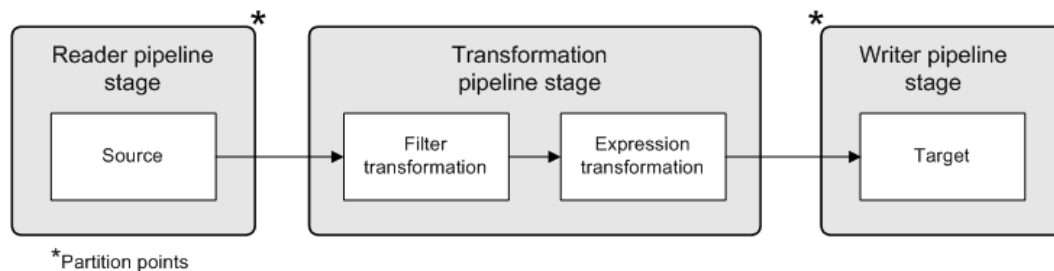
Each mapping contains one or more pipelines. A pipeline consists of a Read transformation and all the transformations that receive data from that Read transformation. The Data Integration Service separates a mapping pipeline into pipeline stages and then performs the extract, transformation, and load for each pipeline stage in parallel.

Partition points mark the boundaries in a pipeline and divide the pipeline into stages. For every mapping pipeline, the Data Integration Service adds a partition point after the Read transformation and before the Write transformation to create multiple pipeline stages.

Each pipeline stage runs in one of the following threads:

- Reader thread that controls how the Data Integration Service extracts data from the source.
- Transformation thread that controls how the Data Integration Service processes data in the pipeline.
- Writer thread that controls how the Data Integration Service loads data to the target.

The following figure shows a mapping separated into a reader pipeline stage, a transformation pipeline stage, and a writer pipeline stage:



Because the pipeline contains three stages, the Data Integration Service can process three sets of rows concurrently and optimize mapping performance. For example, while the reader thread processes the third row set, the transformation thread processes the second row set, and the writer thread processes the first row set.

The following table shows how multiple threads can concurrently process three sets of rows:

Reader Thread	Transformation Thread	Writer Thread
Row Set 1	-	-
Row Set 2	Row Set 1	-
Row Set 3	Row Set 2	Row Set 1
Row Set 4	Row Set 3	Row Set 2
Row Set n	Row Set (n-1)	Row Set (n-2)

If the mapping pipeline contains transformations that perform complicated calculations, processing the transformation pipeline stage can take a long time. To optimize performance, the Data Integration Service adds partition points before some transformations to create an additional transformation pipeline stage.

Multiple Threads for Each Pipeline Stage

When maximum parallelism is set to a value greater than 1, partitioning is enabled. The Data Integration Service separates a mapping into pipeline stages and uses multiple threads to process each stage.

When you maximize parallelism, the Data Integration Service dynamically performs the following tasks at run time:

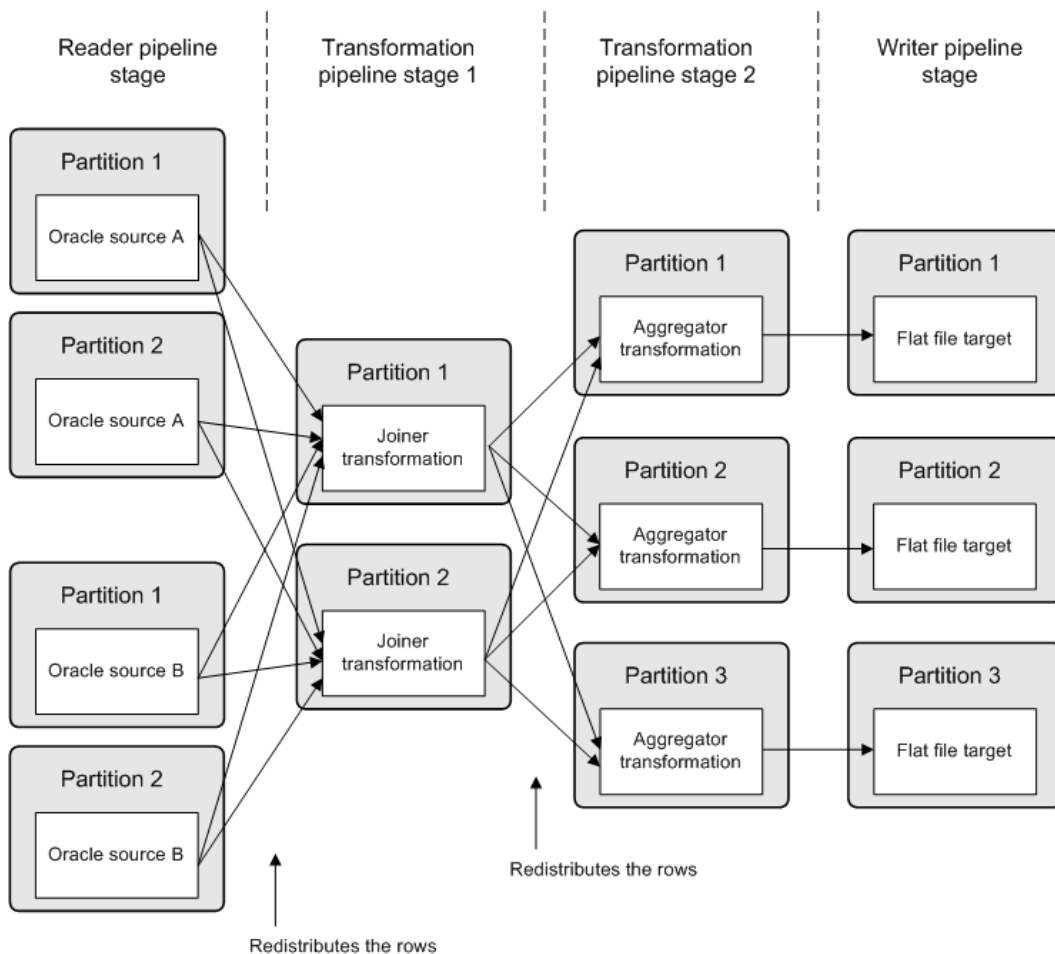
Divides the data into partitions.

The Data Integration Service dynamically divides the underlying data into partitions and runs the partitions concurrently. The Data Integration Service determines the optimal number of threads for each pipeline stage. The number of threads used for a single pipeline stage cannot exceed the maximum parallelism value. The Data Integration Service can use a different number of threads for each pipeline stage.

Redistributes data across partition points.

The Data Integration Service dynamically determines the best way to redistribute data across a partition point based on the transformation requirements.

The following image shows an example mapping that distributes data across multiple partitions for each pipeline stage:



In the preceding image, maximum parallelism for the Data Integration Service is three. Maximum parallelism for the mapping is Auto. The Data Integration Service separates the mapping into four pipeline stages and

uses a total of 12 threads to run the mapping. The Data Integration Service performs the following tasks at each of the pipeline stages:

- At the reader pipeline stage, the Data Integration Service queries the Oracle database system to discover that both source tables, source A and source B, have two database partitions. The Data Integration Service uses one reader thread for each database partition.
- At the first transformation pipeline stage, the Data Integration Service redistributes the data to group rows for the join condition across two threads.
- At the second transformation pipeline stage, the Data Integration Service determines that three threads are optimal for the Aggregator transformation. The service redistributes the data to group rows for the aggregate expression across three threads.
- At the writer pipeline stage, the Data Integration Service does not need to redistribute the rows across the target partition point. All rows in a single partition stay in that partition after crossing the target partition point.

Maximum Parallelism Guidelines

Maximum parallelism determines the maximum number of parallel threads that can process a single pipeline stage. Configure the **Maximum Parallelism** property for the Data Integration Service based on the available hardware resources. When you increase the maximum parallelism value, you might decrease the amount of processing time.

Consider the following guidelines when you configure maximum parallelism:

Increase the value based on the number of available CPUs.

Increase the maximum parallelism value based on the number of CPUs available on the nodes where mappings run. When you increase the maximum parallelism value, the Data Integration Service uses more threads to run the mapping and leverages more CPUs. A simple mapping runs faster in two partitions, but typically requires twice the amount of CPU than when the mapping runs in a single partition.

Consider the total number of processing threads.

Consider the total number of processing threads when setting the maximum parallelism value. If a complex mapping results in multiple additional partition points, the Data Integration Service might use more processing threads than the CPU can handle.

The total number of processing threads is equal to the maximum parallelism value.

Consider the other jobs that the Data Integration Service must run.

If you configure maximum parallelism such that each mapping uses a large number of threads, fewer threads are available for the Data Integration Service to run additional jobs.

Optionally change the value for a mapping.

By default, the maximum parallelism for each mapping is set to Auto. Each mapping uses the maximum parallelism value defined for the Data Integration Service.

In the Developer tool, developers can change the maximum parallelism value in the mapping run-time properties to define a maximum value for a particular mapping. When maximum parallelism is set to different integer values for the Data Integration Service and the mapping, the Data Integration Service uses the minimum value of the two.

Note: You cannot use the Developer tool to change the maximum parallelism value for profiles. When the Data Integration Service converts a profile job into one or more mappings, the mappings always use Auto for the mapping maximum parallelism value.

Enabling Partitioning for Mappings and Profiles

To enable partitioning for mappings, column profiles, and data domain discovery, set maximum parallelism for the Data Integration Service to a value greater than 1.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the Data Integration Service.
3. In the contents panel, click the **Properties** view.
4. In the **Execution Options** section, click **Edit**.
5. Enter a value greater than 1 for the **Maximum Parallelism** property.
6. Click **OK**.
7. Recycle the Data Integration Service to apply the changes.

Optimize Cache and Target Directories for Partitioning

For optimal performance during cache partitioning for Aggregator, Joiner, Rank, and Sorter transformations, configure multiple cache directories for the Data Integration Service. For optimal performance when multiple threads write to a file target, configure multiple target directories for the Data Integration Service.

When multiple threads write to a single directory, the mapping might encounter a bottleneck due to input/output (I/O) contention. An I/O contention can occur when threads write data to the file system at the same time.

When you configure multiple directories, the Data Integration Service determines the output directory for each thread in a round-robin fashion. For example, you configure a flat file data object to use directoryA and directoryB as target directories. If the Data Integration Service uses four threads to write to the file target, the first and third writer threads write target files to directoryA. The second and fourth writer threads write target files to directoryB.

If the Data Integration Service does not use cache partitioning for transformations or does not use multiple threads to write to the target, the service writes the files to the first listed directory.

In the Administrator tool, you configure multiple cache and target directories by entering multiple directories separated by semicolons for the Data Integration Service execution properties. Configure the directories in the following execution properties:

Cache Directory

Defines the cache directories for Aggregator, Joiner, and Rank transformations. By default, the transformations use the CacheDir system parameter to access the cache directory value defined for the Data Integration Service.

Temporary Directories

Defines the cache directories for Sorter transformations. By default, the Sorter transformation uses the TempDir system parameter to access the temporary directory value defined for the Data Integration Service.

Target Directory

Defines the target directories for flat file targets. By default, flat file targets use the TargetDir system parameter to access the target directory value defined for the Data Integration Service.

Instead of using the default system parameters, developers can configure multiple directories specific to the transformation or flat file data object in the Developer tool.

Note: A Lookup transformation can only use a single cache directory.

Result Set Caching

Result set caching enables the Data Integration Service to use cached results for SQL data service queries and web service requests. Users that run identical queries in a short period of time may want to use result set caching to decrease the runtime of identical queries.

When you configure result set caching, the Data Integration Service caches the results of the DTM process associated with each SQL data service query and web service request. The Data Integration Service caches the results for the expiration period that you configure. When an external client makes the same query or request before the cache expires, the Data Integration Service returns the cached results.

The Result Set Cache Manager creates in-memory caches to temporarily store the results of the DTM process. If the Result Set Cache Manager requires more space than allocated in the result set cache properties, it stores the data in encrypted cache files. The files are saved at `<Domain_install_dir>/tomcat/bin/disTemp/<Service_Name>/<Node_Name>/. Do not rename or move the cache files.`

Complete the following steps to configure result set caching for SQL data services and web service operations:

1. Configure the result set cache properties in the Data Integration Service process properties.
2. Configure the cache expiration period in the SQL data service properties.
3. Configure the cache expiration period in the web service operation properties. If you want the Data Integration Service to cache the results by user, enable WS-Security in the web service properties.

The Data Integration Service purges result set caches in the following situations:

- When the result set cache expires, the Data Integration Service purges the cache.
- When you restart an application or run the `infacmd dis purgeResultSetCache` command, the Data Integration Service purges the result set cache for objects in the application.
- When you restart a Data Integration Service, the Data Integration Service purges the result set cache for objects in applications that run on the Data Integration Service.
- When you change the permissions for a user, the Data Integration Service purges the result set cache associated with that user.

Data Object Caching

The Data Integration Service uses data object caching to access pre-built logical data objects and virtual tables. Enable data object caching to increase performance for mappings, SQL data service queries, and web service requests that include logical data objects and virtual tables.

By default, the Data Integration Service extracts source data and builds required data objects when it runs a mapping, SQL data service query, or a web service request. When you enable data object caching, the Data Integration Service can use cached logical data objects and virtual tables.

Perform the following steps to configure data object caching for logical data objects and virtual tables in an application:

1. Configure the data object cache database connection in the cache properties for the Data Integration Service.
2. Enable caching in the properties of logical data objects or virtual tables in an application.

By default, the Data Object Cache Manager component of the Data Integration Service manages the cache tables for logical data objects and virtual tables in the data object cache database. When the Data Object Cache Manager manages the cache, it inserts all data into the cache tables with each refresh. If you want to incrementally update the cache tables, you can choose to manage the cache tables yourself using a database client or other external tool. After enabling data object caching, you can configure a logical data object or virtual table to use a user-managed cache table.

To use the Timestamp with Time Zone data type and to enable data object caching for IBM DB2 or for Microsoft SQL Server, set the date time format of the deployed mapping to the "YYYY-MM-DD HH24:MI:SS" format. The Data Integration Service writes the data up to seconds.

Cache Tables

The Data Object Cache Manager is the component of the Data Integration Service that creates and manages cache tables in a relational database.

You can use the following database types to store data object cache tables:

- IBM DB2
- Microsoft SQL Server
- Oracle

After the database administrator sets up the data object cache database, use the Administrator tool to create a connection to the database. Then, you configure the Data Integration Service to use the cache database connection.

When data object caching is enabled, the Data Object Cache Manager creates a cache table when you start the application that contains the logical data object or virtual table. It creates one table in the cache database for each cached logical data object or virtual table in an application. The Data Object Cache Manager uses a prefix of *CACHE* to name each table.

Objects within an application share cache tables, but objects in different applications do not. If one logical data object or virtual table is used in multiple applications, the Data Object Cache Manager creates a separate cache table for each instance of the object.

Data Object Caching Configuration

To configure data object caching, configure the cache database connection for the Data Integration Service. Then, enable caching for each logical data object or virtual table that end users access frequently.

Perform the following steps to configure data object caching:

1. Configure the cache database connection in the cache properties for the Data Integration Service.

The Data Object Cache Manager creates the cache tables in this database.

2. Enable caching in the properties of logical data objects or virtual tables in an application.

When you enable caching, you can also configure the Data Integration Service to generate indexes on the cache tables based on a column. Indexes can increase the performance of queries on the cache database.

Step 1. Configure the Cache Database Connection

The Data Integration Service stores cached logical data objects and virtual tables in the data object cache database. You configure the connection that the Data Integration Service uses to access the database.

Verify that the database administrator has set up the data object cache database and that you have created the connection to the database.

To configure the connection for the Data Integration Service, click the **Properties** view for the service in the Administrator tool. Click **Edit** in the **Logical Data Object/Virtual Table Cache** area, and then select the database connection name for the **Cache Connection** property. Restart the service for the property to take effect.

Step 2. Enable Data Object Caching for an Object

To enable caching for an object, stop the application that contains the logical data object or virtual table, edit the object properties, and restart the application.

1. In the Administrator tool, click the Manage tab > Services and Nodes view.
2. In the Domain Navigator, select the Data Integration Service.
3. Click the **Applications** view.
4. Select the application that contains the logical data object or virtual table for which you want to enable caching.
5. Stop the application.
6. Expand the application, and select the logical data object or virtual table.
7. In the **Logical Data Object Properties** or **Virtual Table Properties** area, click **Edit**.
The **Edit Properties** dialog box appears.
8. Select **Enable Caching**.
9. In the **Cache Refresh Period** property, enter the amount of time in minutes that the Data Object Cache Manager waits before refreshing the cache.

For example, if you enter 720, the Data Object Cache Manager refreshes the cache every 12 hours. If you leave the default value of zero, the Data Object Cache Manager does not refresh the cache according to a schedule. You must manually refresh the cache using the `infacmd dis RefreshDataObjectCache` command.

10. Leave the **Cache Table Name** property blank.
When you enter a table name, the Data Object Cache Manager does not manage the cache for the object. Enter a table name only when you want to use a user-managed cache table. A user-managed cache table is a table in the data object cache database that you create, populate, and manually refresh when needed.
11. Click **OK**.
12. To generate indexes on the cache table based on a column, expand the logical data object or virtual table.
 - a. Select a column, and then click **Edit** in the **Logical Data Object Column Properties** or **Virtual Table Column Properties** area.
The **Edit Column Properties** dialog box appears.
 - b. Select **Create Index** and then click **OK**.
13. Restart the application.
The Data Object Cache Manager creates and populates the cache table.

Data Object Cache Management

By default, the Data Object Cache Manager manages the cache tables in the data object cache database. You can use the Administrator tool or `infacmd` to configure when and how the Data Object Cache Manager

refreshes the cache. Or, you can choose to manage the cache tables yourself using a database client or other external tool.

When the Data Object Cache Manager manages the cache, it inserts all data into the cache table with each refresh. You can choose to manage the cache tables yourself so that you can incrementally update the cache.

Cache Tables Managed by the Data Object Cache Manager

By default, the Data Object Cache Manager manages the cache tables in the data object cache database.

When the Data Object Cache Manager manages the cache tables, you can perform the following operations on the data object cache:

Refresh the cache

You can refresh the cache for a logical data object or virtual table according to a schedule or manually. To refresh data according to a schedule, set the cache refresh period for the logical data object or virtual table in the Administrator tool.

To refresh the cache manually, use the `infacmd dis RefreshDataObjectCache` command. When the Data Object Cache Manager refreshes the cache, it creates a new cache. If an end user runs a mapping or queries an SQL data service during a cache refresh, the Data Integration Service returns information from the existing cache.

Abort a refresh

To abort a cache refresh, use the `infacmd dis CancelDataObjectCacheRefresh` command. If you abort a cache refresh, the Data Object Cache Manager restores the existing cache.

Purge the cache

To purge the cache, use the `infacmd dis PurgeDataObjectCache` command. You must disable the application before you purge the cache.

User-Managed Cache Tables

A user-managed cache table is a table in the data object cache database that you create, populate, and manually refresh when needed.

Configure a logical data object or virtual table to use a user-managed cache table when you want to incrementally update the cache. When the Data Object Cache Manager manages the cache, it inserts all data into the cache table with each refresh. If the source contains a large data set, the refresh can take a long time to process. Instead, you can configure the object to use a user-managed cache table and then use an external tool to insert only the changed data into the cache table. For example, you can use a PowerCenter CDC mapping to extract changed data for the objects and incrementally update the cache.

When you configure an object to use a user-managed cache table, you must use a database client or other tool to create, populate, purge, and refresh the cache table. You create the user-managed cache table in the data object cache database that the Data Integration Service accesses with the cache database connection.

You cannot use the Administrator tool or command line tools to manage a user-managed cache table. The Data Integration Service uses the cache stored in the user-managed cache table when it runs a mapping, an SQL data service query, or a web service request that includes the object. However, the Data Object Cache Manager does not manage the cache table. When you use the **Monitor** tab to monitor an object that uses a user-managed cache table, the object has a cache state of Skipped.

Note: If the user-managed cache table is stored in a Microsoft SQL Server database and the database user name is not the same as the schema name, you must specify a schema name in the database connection object. Otherwise, mappings, SQL data service queries, and web service requests that access the cache fail.

Configure User-Managed Cache Tables

To configure a logical data object or virtual table to use a user-managed cache table, you must create a table in the data object cache database. Populate the table with the initial cache, and then enter the table name in the data object properties.

Note: Before you configure an object to use a user-managed cache table, you must configure the cache database connection for the Data Integration Service. You also must enable data object caching for the object so that the Data Object Cache Manager creates the default cache table.

Step 1. Find the Name of the Default Cache Table

On the **Monitor** tab of the Administrator tool, find the name of the default cache table that the Data Object Cache Manager created after you enabled data object caching for the object.

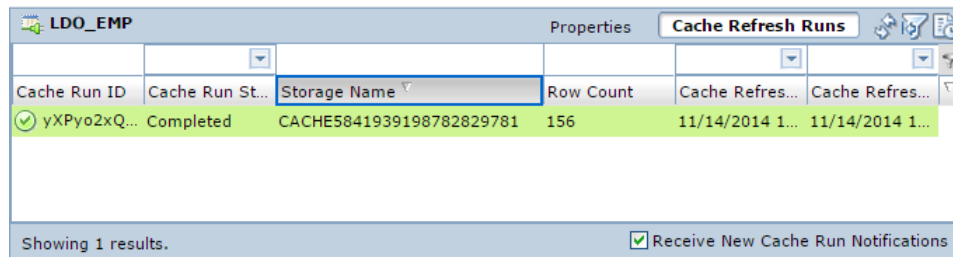
1. In the Administrator tool, click the **Monitor** tab.
2. Click the **Execution Statistics** view.
3. In the Navigator, expand a Data Integration Service.
4. In the Navigator, expand an application and select **Logical Data Objects** or **SQL Data Services**.
5. In the contents panel, perform one of the following steps:
 - Select a logical data object.
 - Select an SQL data service, click the **Virtual Tables** view, and then select a table row.

Details about the selected object appear in the details panel.

6. In the details panel, select the **Cache Refresh Runs** view.

The Storage Name column lists the name of the default cache table that the Data Object Cache Manager created.

For example, the following image displays a cache table named *CACHE5841939198782829781*:



Cache Run ID	Cache Run St...	Storage Name	Row Count	Cache Refres...	Cache Refres...
✓ yXPyo2xQ...	Completed	CACHE5841939198782829781	156	11/14/2014 1...	11/14/2014 1...

Showing 1 results. ☒ Receive New Cache Run Notifications

Step 2. Create the User-Managed Cache Table

Ask the database administrator to create a table in the data object cache database using the same table structure as the default cache table.

Use a database client to find the default cache table in the data object cache database. Use the SQL DDL from the default cache table to create the user-managed cache table with a different name. The name of the user-managed cache table cannot have the prefix *CACHE*. The prefix *CACHE* is reserved for names of cache tables that are managed by the Data Object Cache Manager.

After creating the user-managed cache table, populate the table by copying the initial cache data from the default cache table.

Step 3. Configure the Object to Use the User-Managed Cache Table

To configure a logical data object or virtual table to use a user-managed cache table, stop the application that contains the object, edit the object properties, and restart the application.

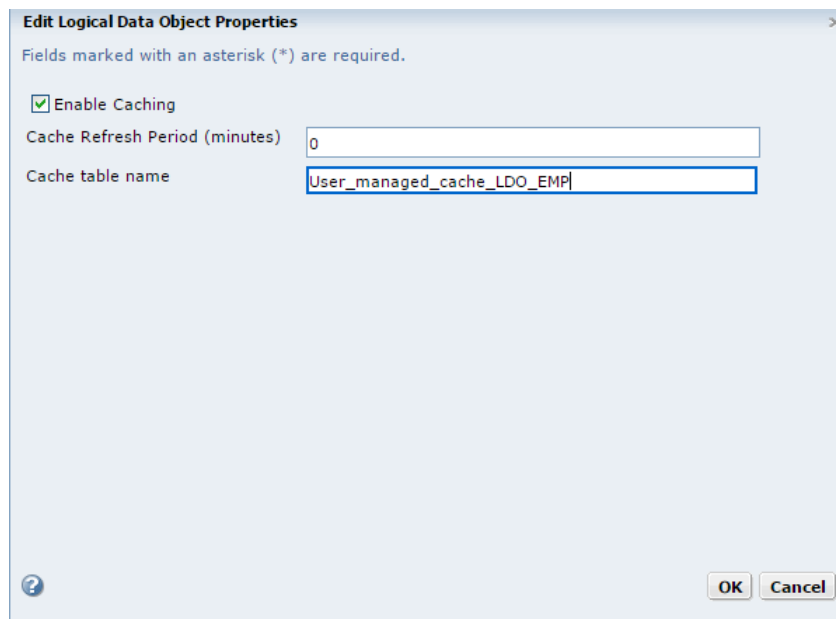
1. In the Administrator tool, select the Data Integration Service.
2. Click the **Applications** view.
3. Select the application that contains the logical data object or virtual table for which you want to use a user-managed cache table.
4. Stop the application.
5. Expand the application, and select the logical data object or virtual table.
6. In the **Logical Data Object Properties** or **Virtual Table Properties** area, click **Edit**.

The **Edit Properties** dialog box appears.

7. Enter the name of the user-managed cache table that you created in the data object cache database.

When you enter a cache table name, the Data Object Cache Manager does not generate the cache for the object and ignores the cache refresh period.

The following figure shows a logical data object configured to use a user-managed cache table:



8. Click **OK**.
9. Restart the application.

Persisting Virtual Data in Temporary Tables

A temporary table is a table in a relational database that stores intermediate, temporary data. Complex queries commonly require storage for large amounts of intermediate data, such as information from joins. When you implement temporary tables, business intelligence tools can retrieve this data from the temporary table instead of the SQL data service. This results in an increase in performance.

Temporary tables also provide increased security in two ways. First, only the user of the active session can access the tables. Also, the tables persist while a session is active, and the database drops the tables when the connection closes.

You must configure the Table Storage Connection property of the Data Integration Service before the database administrator creates a temporary table.

Temporary tables for all SQL data services in a Data Integration Service use the same relational database connection. When the connection to the SQL data service is active, you can connect to the SQL data service through a JDBC or ODBC client. The relational database drops temporary tables when the session ends. If the Data Integration Service unexpectedly shuts down, the relational database drops temporary tables on the next Data Integration Service startup.

Temporary Table Implementation

You can store intermediate query result set data in temporary tables when complex queries produce large amounts of intermediate data. For example, temporary tables can store frequently used join results. Business intelligence tools can query the temporary table instead of the SQL data service, resulting in increased performance.

To implement temporary tables, the Informatica administrator and the business intelligence tool user perform the following separate tasks:

Step 1. The Informatica administrator creates a connection for the data integration service.

In the Administrator tool, create a connection to the SQL data service. Edit the **SQL Properties** of the Data Integration Service and select a relational database connection for the **Table Storage Connection** property. Recycle the Data Information Service.

Step 2. The business intelligence tool user creates a connection for the SQL data service.

In a business intelligence tool, create a connection to the SQL data service. The connection uses the Informatica ODBC or JDBC driver.

Step 3. Queries from the business intelligence tool create and use temporary tables.

While the connection is active, the business intelligence tool issues queries to the SQL data service. These queries create and use temporary tables to store large amounts of data that the complex query produces. When the connection ends, the database drops the temporary table.

Temporary Table Operations

After you create the SQL data service connection, you can use SQL operations to create, populate, select from, or drop a temporary table. You can issue these commands in a regular or stored SQL statement.

You can perform the following operations:

Create a temporary table.

To create a temporary table on the relational database, use the following syntax:

```
CREATE TABLE emp (empID INTEGER PRIMARY KEY,eName char(50) NOT NULL,)
```

You can specify the table name in the SQL data service.

Note: Use `CREATE TABLE`, not `CREATE TEMPORARY TABLE`. The use of `CREATE TEMPORARY TABLE` is not supported.

Create a temporary table from a source table.

You can create a temporary table with or without data from a source table.

The following syntax is supported in Informatica Data Services version 9.5.1:

```
CREATE TABLE emp.backup as select * from emp
```

Where `emp` is an existing schema in the SQL data service that you connected to.

The following syntax is supported in Informatica Data Services version 9.6.0 and 9.6.1:

```
CREATE TABLE emp.backup as select * from emp [ [LIMIT n] ]
```

Where `emp` is an existing schema in the SQL data service that you connected to.

When you create a temporary table with data, the Data Integration Service populates the table with the data. The `CREATE AS` operator copies columns from a database table into the temporary table.

You cannot maintain foreign key or primary key constraints when you use `CREATE AS`.

You can cancel a request before the Data Integration Service copies all the data.

Note: The Informatica administrator must create a connection, and then configure it in **SQL Properties** as the **Table Storage Connection**, before you create the temporary table.

Insert data into a temporary table.

To insert data into a temporary table, use the `INSERT INTO <temp_table>` statement. You can insert literal data and query data into a temporary table.

The following table shows examples of SQL statements that you can use to insert literal data and query data into a temporary table:

Type	Description
Literal data	<p>Literals describe a user or system-supplied string or value that is not an identifier or keyword. Use strings, numbers, dates, or boolean values when you insert literal data into a temporary table. Use the following statement format to insert literal data into a temporary table:</p> <pre>INSERT INTO <TABLENAME> <OPTIONAL COLUMN LIST> VALUES (<VALUE LIST>), (<VALUE LIST>)</pre> <p>For example, <code>INSERT INTO temp_dept (dept_id, dept_name, location) VALUES (2, 'Marketing', 'Los Angeles')</code>.</p>
Query data	<p>You can query an SQL data service and insert data from the query into a temporary table. Use the following statement format to insert query data into a temporary table:</p> <pre>INSERT INTO <TABLENAME> <OPTIONAL COLUMN LIST> <SELECT QUERY></pre> <p>For example, <code>INSERT INTO temp_dept(dept_id, dept_name, location) SELECT dept_id, dept_name, location from dept where dept_id = 99.</code></p> <p>You can use a set operator, such as <code>UNION</code>, in the SQL statement when you insert query data into a temporary table. Use the following statement format when you use a set operator:</p> <pre>INSERT INTO <TABLENAME> <OPTIONAL COLUMN LIST> (<SELECT QUERY> <SET OPERATOR> <SELECT QUERY>)</pre> <p>For example, <code>INSERT INTO temp_dept select * from north_america_dept UNION select * from asia_dept.</code></p>

Select from a temporary table.

You can query the temporary table with the `SELECT ... from <table>` statement.

Drop a temporary table.

To drop a temporary table from the relational database, use the following syntax:

```
DROP TABLE <tableName>
```

If the table is not dropped on the physical database, the SQL data service drops the table the next time the Data Integration Service starts, if the table still exists.

Rules and Guidelines for Temporary Tables

Consider the following rules and guidelines for creation and use of temporary tables:

- You can specify schema and default schema for a temporary table.
- You can place the primary key, NULL, NOT NULL, and DEFAULT constraints on a temporary table.
- You cannot place a foreign key or CHECK and UNIQUE constraints on a temporary table.
- You cannot issue a query that contains a common table expression or a correlated subquery against a temporary table.
- `CREATE AS` statements cannot contain a correlated subquery.

Content Management for the Profiling Warehouse

To create and run profiles and scorecards, you must associate the Data Integration Service with a profiling warehouse. You can specify the profiling warehouse when you create the Data Integration Service or when you edit the Data Integration Service properties.

The profiling warehouse stores profiling data and metadata. If you specify a new profiling warehouse database, you must create the profiling content. If you specify an existing profiling warehouse, you can use the existing content or delete and create new content.

You can create or delete content for a profiling warehouse at any time. You may choose to delete the content of a profiling warehouse to delete corrupted data or to increase disk or database space.

Creating and Deleting Profiling Warehouse Content

The Data Integration Service must be running when you create or delete profiling warehouse content.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select a Data Integration Service that has an associated profiling warehouse.
3. To create profiling warehouse content, click the Actions menu on the **Manage** tab and select **Profiling Warehouse Database Contents > Create**.
4. To delete profiling warehouse content, click the Actions menu on the **Manage** tab and select **Profiling Warehouse Database Contents > Delete**.

Database Management

You need to periodically review and manage the profiling warehouse database growth. You can remove profile information that you no longer need and monitor or maintain the profiling warehouse tables.

The need for maintenance depends on different scenarios, such as short-term projects or when you no longer need the profile results. You can remove unused profile results and recover disk space used by the results so that you can reuse the database space for other purposes.

Purge

Purges profile and scorecard results from the profiling warehouse. The `infacmd ps Purge` command purges all the profile and scorecard results except for the results from the latest profile or scorecard run.

The `infacmd ps Purge` command uses the following syntax:

```
Purge
<-DomainName|-dn> domain_name
[<-Gateway|-hp> gateway_name]
[<-NodeName|-nn>] node_name
<-UserName|-un> user_name
<-Password|-pd> Password
[<-SecurityDomain|-sdn> security_domain]
<-MrsServiceName|-msn> MRS_name
<-DsServiceName|-dsn> data_integration_service_name
<-ObjectType|-ot> object_type
<-ObjectPathAndName|-opn> MRS_object_path
[<-RetainDays|-rd> results_retain_days]
[<-ProjectFolderPath|-pf> project_folder_path]
[<-ProfileName|-pt> profile_task_name]
[<-Recursive|-r> recursive]
[<-PurgeAllResults|-pa> purge_all_results]
```

The following table describes infacmd ps Purge options and arguments:

Option	Argument	Description
-DomainName -dn	domain_name	Required. The name of the Informatica domain. You can set the domain name with the -dn option or the environment variable INFA_DEFAULT_DOMAIN. If you set a domain name with both methods, the -dn option takes precedence.
-Gateway -hp	gateway_name	Optional if you run the command from the Informatica installation \bin directory. Required if you run the command from another location. The gateway node name. Use the following syntax: [Domain_Host]:[HTTP_Port]
-NodeName -nn	node_name	Required. The name of the node where the Data Integration Service runs.
-UserName -un	user_name	Required if the domain uses Native or LDAP authentication. User name to connect to the domain. You can set the user name with the -un option or the environment variable INFA_DEFAULT_DOMAIN_USER. If you set a user name with both methods, the -un option takes precedence. Optional if the domain uses Kerberos authentication. To run the command with single sign-on, do not set the user name. If you set the user name, the command runs without single sign-on.
-Password -pd	Password	Required if you specify the user name. Password for the user name. The password is case sensitive. You can set a password with the -pd option or the environment variable INFA_DEFAULT_DOMAIN_PASSWORD. If you set a password with both methods, the password set with the -pd option takes precedence.
-SecurityDomain -sdn	security_domain	Required if the domain uses LDAP authentication. Optional if the domain uses native authentication or Kerberos authentication. Name of the security domain to which the domain user belongs. You can set a security domain with the -sdn option or the environment variable INFA_DEFAULT_SECURITY_DOMAIN. If you set a security domain name with both methods, the -sdn option takes precedence. The security domain name is case sensitive. If the domain uses native or LDAP authentication, the default is Native. If the domain uses Kerberos authentication, the default is the LDAP security domain created during installation. The name of the security domain is the same as the user realm specified during installation.
-MrsServiceName -msn	MRS_name	Required. The Model Repository Service name.
-DsServiceName -dsn	data_integration_service_name	Required. The Data Integration Service name

Option	Argument	Description
-ObjectType -ot	-	Required. Enter profile or scorecard.
-ObjectPathAndName -opn *	MRS_object_path	Optional. Do not use with ProjectFolderPath or Recursive. The path to the profile or scorecard in the Model repository. Use the following syntax: ProjectName/FolderName/.../{SubFolder_Name/ObjectName ProjectName/ObjectName}
-RetainDays -rd	results_retain_days	Optional. Specifies the time range for the profile and scorecard results to be eligible for retention in the profiling warehouse. The Data Integration Service purges the rest of the profile and scorecard results. For example, if you enter -rd 10, then the results from the current day and past nine days are retained and the rest of the results are purged from the profiling warehouse.
-ProjectFolderPath -pf *	project_folder_path	Optional. Do not use with ObjectPathAndName or ProfileTaskName. The names of the project and folder where the profile or scorecard is stored. Use the following syntax: ProjectName/FolderName
-ProfileName -pt *	profile_task_name	Optional. The name of the profile task that you want to purge. If a folder has only one profile, then you can use only the ProjectFolderPath option because the ProjectFolderPath includes the name of the profile that contains the profile task. If a folder has multiple profiles in a folder, then use the ProfileName option along with the ProjectFolderPath option to specify the profile name.
-Recursive -r	recursive	Optional. Do not use with ObjectPathAndName. Applies the command to objects in the folder that you specify and its subfolders.
-PurgeAllResults -pa	purge_all_results	Optional. Set this option to purge all results for the profile or scorecard object. Use with the -recursive option to apply the command to profile and scorecard results in the folder that you specify and its subfolders.
* To run the command, you need to specify ObjectPathAndName or ProjectFolderPath or ProfileTaskName.		

Tablespace Recovery

As part of the regular profile operations, the Data Integration Service writes profile results to the profiling warehouse and deletes results from the profiling warehouse. The indexes and base tables can become fragmented over a period of time. You need to reclaim the unused disk space, especially for Index Organized Tables in Oracle database.

Most of the profiling warehouse tables contain relatively small amount of data and you do not need to recover the tablespace and index space.

The following tables store large amounts of profile data and deleting the tables can leave the tables fragmented:

Name	Description
IDP_FIELD_VERBOSE_SMRY_DATA	Stores the value frequencies
IDP_VERBOSE_FIELD_DTL_RES	Stores the staged data

When you perform the tablespace recovery, ensure that no user runs a profile task. After you recover the data, update the database statistics to reflect the changed structure.

IBM DB2

The recommendation is to shut down the Data Integration Service when you reorganize the tables and indexes.

To recover the database for a table, run the following command:

```
REORG TABLE <TABLE NAME>
```

```
REORG INDEXES ALL FOR TABLE <TABLE NAME> ALLOW WRITE ACCESS CLEANUP ONLY ALL
```

Oracle

You can rebuild Index Organized Tables in Oracle. This action reclaims unused fragments inside the index and applies to the IDP_FIELD_VERBOSE_SMRY_DATA and IDP_FIELD_VERBOSE_SMRY_DATA profiling warehouse tables.

To recover the database for a table, run the following command:

```
ALTER TABLE <Table Name> MOVE ONLINE
```

Microsoft SQL Server

Microsoft SQL Server reclaims unused space back into the tablespace and compacts indexes when rows are deleted. You do not need to maintain the database.

Database Statistics

Update the database statistics to allow the database to quickly run the queries on the profiling warehouse.

Database Statistics on IBM DB2

IBM DB2 recommends that you run the RUNSTATS command to update the statistics after a lot of updates have been made to a table or after a reorganization of the table.

To update the statistics, run the following command:

```
RUNSTATS ON TABLE <TABLE NAME> WITH DISTRIBUTION AND DETAILED INDEXES ALL
```

Database Statistics on Oracle

By default, Oracle gathers database statistics and therefore, you do not need to perform any action. For more information, refer the documentation on Oracle `DBMS_STATS` command.

Database Statistics on Microsoft SQL Server

By default, Microsoft SQL Server gathers statistics and therefore, no action is required. To update the statistics more frequently than the default recommended option, refer the documentation on SQL Server `UPDATE STATISTICS` command.

Web Service Security Management

An HTTP client filter, transport layer security, and message layer security can provide secure data transfer and authorized data access for a web service. When you configure message layer security, the Data Integration Service can pass credentials to connections.

You can configure the following security option for a REST web service:

Is Authentication Required

Enables basic authentication for the REST web service. Basic authentication requires that each web service request includes a user name and a password to the domain. Enable the property from the Data Integration Service in the Administrator tool. Click **Applications > ApplicationName REST Web Service > isAuthenticationRequired**. When authentication is required, each GET request requires a user name and password before the REST web service returns a response. Default is disabled.

You can configure the following security options for a SOAP web service:

HTTP Client Filter

If you want the Data Integration Service to accept requests based on the host name or IP address of the web service client, use the Administrator tool to configure an HTTP client filter. By default, a web service client running on any machine can send requests.

Message Layer Security

If you want the Data Integration Service to authenticate user credentials in SOAP requests, use the Administrator tool to enable WS-Security and configure web service permissions. The Data Integration Service can validate user credentials that are provided as a user name token in the SOAP request. If the user name token is not valid, the Data Integration Service rejects the request and sends a system-defined fault to the web service client. If a user does not have permission to execute the web service operation, the Data Integration Service rejects the request and sends a system-defined fault to the web service client.

Transport Layer Security (TLS)

If you want the web service and web service client to communicate using an HTTPS URL, use the Administrator tool to enable transport layer security (TLS) for a web service. The Data Integration Service that the web service runs on must also use the HTTPS protocol. An HTTPS URL uses SSL to provide a secure connection for data transfer between a web service and a web service client.

Pass-Through Security

If an operation mapping requires connection credentials, the Data Integration Service can pass credentials from the user name token in the SOAP request to the connection. To configure the Data

Integration Service to pass credentials to a connection, use the Administrator tool to configure the Data Integration Service to use pass-through security for the connection and enable WS-Security for the web service.

Note: You cannot use pass-through security when the user name token includes a hashed or digested password.

HTTP Client Filter

An HTTP client filter specifies web services client machine that can send requests to the Data Integration Service. By default, a web service client running on any machine can send requests.

To specify machines that can send web service request to a Data Integration Service, configure the HTTP client filter properties in the Data Integration Service properties. When you configure these properties, the Data Integration Service compares the IP address or host name of machines that submit web service requests against these properties. The Data Integration Service either allows the request to continue or refuses to process the request.

You can use constants or Java regular expressions as values for these properties. You can include a period (.) as a wildcard character in a value.

Note: You can allow or deny requests from a web service client that runs on the same machine as the Data Integration Service. Enter the host name of the Data Integration Service machine in the allowed or denied host names property.

Example

The Finance department wants to configure a web service to accept web service requests from a range of IP addresses. To configure the Data Integration Service to accept web service requests from machines in a local network, enter the following expression as an allowed IP Address:

```
"192\.\168\.\1\.[0-9]*"
```

The Data Integration Service accepts requests from machines with IP addresses that match this pattern. The Data Integration Service refuses to process requests from machines with IP addresses that do not match this pattern.

Pass-through Security

Pass-through security is the capability to connect to an SQL data service or an external source with the client user credentials instead of the credentials from a connection object.

Users might have access to different sets of data based on the job in the organization. Client systems restrict access to databases by the user name and the password. When you create an SQL data service, you might combine data from different systems to create one view of the data. However, when you define the connection to the SQL data service, the connection has one user name and password.

If you configure pass-through security, you can restrict users from some of the data in an SQL data service based on their user name. When a user connects to the SQL data service, the Data Integration Service ignores the user name and the password in the connection object. The user connects with the client user name or the LDAP user name.

A web service operation mapping might need to use a connection object to access data. If you configure pass-through security and the web service uses WS-Security, the web service operation mapping connects to a source using the user name and password provided in the web service SOAP request.

Configure pass-through security for a connection in the connection properties of the Administrator tool or with `infacmd dis UpdateServiceOptions`. You can set pass-through security for connections to deployed applications. You cannot set pass-through security in the Developer tool. Only SQL data services and web services recognize the pass-through security configuration.

For more information about configuring security for SQL data services, see the Informatica How-To Library article "How to Configure Security for SQL Data Services":
https://kb.informatica.com/h2l/HowTo%20Library/1/0266_ConfiguringSecurityForSQLDataServices.pdf.

Example

An organization combines employee data from multiple databases to present a single view of employee data in an SQL data service. The SQL data service contains data from the Employee and Compensation databases. The Employee database contains name, address, and department information. The Compensation database contains salary and stock option information.

A user might have access to the Employee database but not the Compensation database. When the user runs a query against the SQL data service, the Data Integration Service replaces the credentials in each database connection with the user name and the user password. The query fails if the user includes salary information from the Compensation database.

Pass-Through Security with Data Object Caching

To use data object caching with pass-through security, you must enable caching in the pass-through security properties for the Data Integration Service.

When you deploy an SQL data service or a web service, you can choose to cache the logical data objects in a database. You must specify the database in which to store the data object cache. The Data Integration Service validates the user credentials for access to the cache database. If a user can connect to the cache database, the user has access to all tables in the cache. The Data Integration Service does not validate user credentials against the source databases when caching is enabled.

For example, you configure caching for the EmployeeSQLDS SQL data service and enable pass-through security for connections. The Data Integration Service caches tables from the Compensation and the Employee databases. A user might not have access to the Compensation database. However, if the user has access to the cache database, the user can select compensation data in an SQL query.

When you configure pass-through security, the default is to disallow data object caching for data objects that depend on pass-through connections. When you enable data object caching with pass-through security, verify that you do not allow unauthorized users access to some of the data in the cache. When you enable caching for pass-through security connections, you enable data object caching for all pass-through security connections.

Adding Pass-Through Security

Enable pass-through security for a connection in the connection properties. Enable data object caching for pass-through security connections in the pass-through security properties of the Data Integration Service.

1. Select a connection.
2. Click the **Properties** view.
3. Edit the connection properties.
The **Edit Connection Properties** dialog box appears.
4. To choose pass-through security for the connection, select the **Pass-through Security Enabled** option.
5. Optionally, select the Data Integration Service for which you want to enable object caching for pass-through security.

6. Click the **Properties** view.
7. Edit the pass-through security options.

The **Edit Pass-through Security Properties** dialog box appears.

8. Select **Allow Caching** to allow data object caching for the SQL data service or web service. This applies to all connections.
9. Click **OK**.

You must recycle the Data Integration Service to enable caching for the connections.

CHAPTER 6

Data Integration Service Grid

This chapter includes the following topics:

- [Data Integration Service Grid Overview, 128](#)
- [Before You Configure a Data Integration Service Grid, 130](#)
- [Grid for Jobs that Run in the Service Process, 130](#)
- [Grid for Jobs that Run in Local Mode, 135](#)
- [Grid for Jobs that Run in Remote Mode, 141](#)
- [Grid and Content Management Service, 150](#)
- [Maximum Number of Concurrent Jobs on a Grid, 151](#)
- [Editing a Grid, 152](#)
- [Deleting a Grid, 153](#)
- [Troubleshooting a Grid, 153](#)

Data Integration Service Grid Overview

If your license includes grid, you can configure the Data Integration Service to run on a grid. A grid is an alias assigned to a group of nodes. When you run jobs on a Data Integration Service grid, you improve scalability and performance by distributing jobs to processes running on multiple nodes in the grid.

To configure a Data Integration Service to run on a grid, you create a grid object and assign nodes to the grid. Then, you assign the Data Integration Service to run on the grid.

When you enable a Data Integration Service assigned to a grid, a Data Integration Service process runs on each node in the grid that has the service role. If a service process shuts down unexpectedly, the Data Integration Service remains available as long as another service process runs on another node. Jobs can run on each node in the grid that has the compute role. The Data Integration Service balances the workload among the nodes based on the type of job and based on how the grid is configured.

When the Data Integration Service runs on a grid, the service and compute components of the Data Integration Service can run on the same node or on different nodes, based on how you configure the grid and the node roles. Nodes in a Data Integration Service grid can have a combination of the service only role, the compute only role, and both the service and compute roles.

Grid Configuration by Job Type

A Data Integration Service that runs on a grid can run DTM instances in the Data Integration Service process, in separate DTM processes on the local node, or in separate DTM processes on remote nodes. Configure the service based on the types of jobs that the service runs.

Configure a Data Integration Service grid based on the following types of jobs that the service runs:

SQL data services and web services

When a Data Integration Service grid runs SQL queries and web service requests, you can configure the service to run jobs in the Data Integration Service process. You can also configure SQL data service and web service jobs to run in separate DTM processes on the local node. All nodes in the grid must have both the service and compute roles. The Data Integration Service dispatches jobs to available nodes in a round-robin fashion.

SQL data service and web service jobs typically achieve better performance when the Data Integration Service runs jobs in the service process.

Mappings, profiles, and workflows that run in local mode

When a Data Integration Service grid runs mappings, profiles, and workflows, you can configure the service to run jobs in separate DTM processes on the local node. All nodes in the grid must have both the service and compute roles. The Data Integration Service dispatches jobs to available nodes in a round-robin fashion.

When the Data Integration Service runs jobs in separate local processes, stability increases because an unexpected interruption to one job does not affect all other jobs.

Mappings, profiles, and workflows that run in remote mode

When a Data Integration Service grid runs mappings, profiles, and workflows, you can configure the service to run jobs in separate DTM processes on remote nodes. The nodes in the grid can have a different combination of roles. The Data Integration Service designates one node with the compute role as the master compute node. The Service Manager on the master compute node communicates with the Resource Manager Service to dispatch jobs to an available worker compute node. The Resource Manager Service matches job requirements with resource availability to identify the best compute node to run the job.

When the Data Integration Service runs jobs in separate remote processes, stability increases because an unexpected interruption to one job does not affect all other jobs. In addition, you can better use the resources available on each node in the grid. When a node has the compute role only, the node does not have to run the service process. The machine uses all available processing power to run mappings.

Note: Ad hoc jobs, with the exception of profiles, can run in the Data Integration Service process or in separate DTM processes on the local node. Ad hoc jobs include mappings run from the Developer tool or previews, scorecards, or drill downs on profile results run from the Developer tool or Analyst tool. If you configure a Data Integration Service grid to run jobs in separate remote processes, the service runs ad hoc jobs in separate local processes.

By default, each Data Integration Service is configured to run jobs in separate local processes, and each node has both the service and compute roles.

If you run SQL queries or web service requests, and you run other job types in which stability and scalability is important, create multiple Data Integration Services. Configure one Data Integration Service grid to run SQL queries and web service requests in the Data Integration Service process. Configure the other Data Integration Service grid to run mappings, profiles, and workflows in separate local processes or in separate remote processes.

Before You Configure a Data Integration Service Grid

Before you configure a Data Integration Service to run on a grid, complete the prerequisite tasks for a grid.

Define and Add Multiple Nodes to the Domain

Run the Informatica installer on each machine that you want to define as a node in the Data Integration Service grid. The installer adds the node to the domain with both the service and compute roles enabled. When you log in to the Administrator tool, the node appears in the Navigator.

Verify that All Grid Nodes are Homogeneous

All machines represented by nodes in a Data Integration Service grid must have homogeneous environments. Verify that each machine meets the following requirements:

- All machines must use the same operating system.
- All machines must use the same locale settings.
- All machines that represent nodes with the compute role or nodes with both the service and compute roles must have installations of the native database client software associated with the databases that the Data Integration Service accesses. For example, you run mappings that read from and write to an Oracle database. You must install and configure the same version of the Oracle client on all nodes in the grid that have the compute role and all nodes in the grid that have both the service and compute roles.

For more information about establishing native connectivity between the Data Integration Service and a database, see [“Configure Native Connectivity on Service Machines” on page 401](#).

Obtain an External HTTP Load Balancer for Web Service Requests

To run web service requests on a Data Integration Service grid, you must obtain and use an external HTTP load balancer. If you do not use an external HTTP load balancer, web service requests are not distributed across the nodes in the grid. Each web service request runs on the node that receives the request from the web service client.

Grid for Jobs that Run in the Service Process

You can configure the Data Integration Service to run jobs in the service process. Configure this option when the service runs SQL data service and web service jobs on a single node or on a grid. All nodes in the grid must have both the service and compute roles.

When you enable a Data Integration Service that runs on a grid, one service process starts on each node with the service role in the grid. The Data Integration Service designates one service process as the master service process, and designates the remaining service processes as worker service processes. When a worker service process starts, it registers itself with the master service process so that the master is aware of the worker.

The master service process manages application deployments and logging. The worker service processes run the SQL data service, web service, and preview jobs. The master service process also acts as a worker service process and completes jobs.

The Data Integration Service balances the workload across the nodes in the grid based on the following job types:

SQL data services

When you connect to an SQL data service from a third-party client tool to run queries against the service, the Data Integration Service dispatches the connection directly to a worker service process. To ensure faster throughput, the Data Integration Service bypasses the master service process. When you establish multiple connections to SQL data services, the Data Integration Service uses round robin to dispatch each connection to a worker service process. When you run multiple queries against the SQL data service using the same connection, each query runs on the same worker service process.

Web services

When you submit a web service request, the Data Integration Service uses an external HTTP load balancer to distribute the request to a worker service process. When you submit multiple requests against web services, the Data Integration Service uses round robin to dispatch each query to a worker service process.

To run web service requests on a grid, you must configure the external HTTP load balancer. Specify the logical URL for the load balancer in the web service properties of the Data Integration Service. When you configure the external load balancer, enter the URLs for all nodes in the grid that have both the service and compute roles. If you do not configure an external HTTP load balancer, web service requests are not distributed across the nodes in the grid. Each web service request runs on the node that receives the request from the web service client.

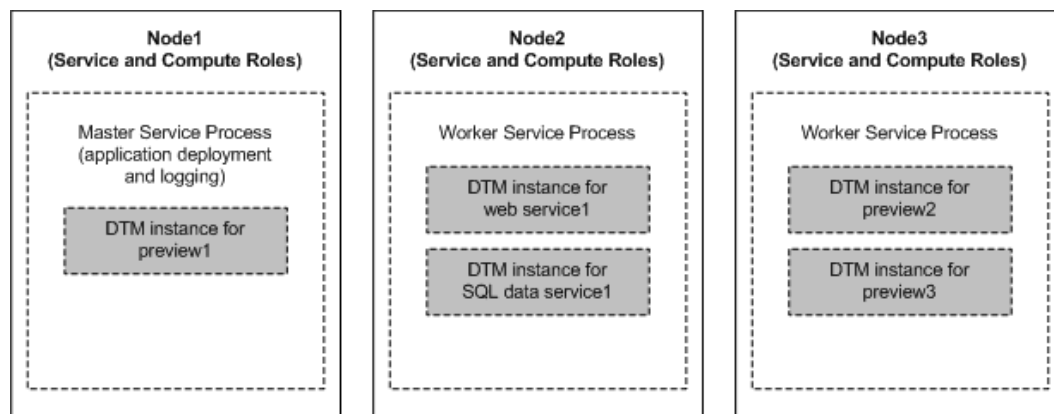
Previews

When you preview a stored procedure output or virtual table data, the Data Integration Service uses round robin to dispatch the first preview query directly to a worker service process. To ensure faster throughput, the Data Integration Service bypasses the master service process. When you preview additional objects from the same login, the Data Integration Service dispatches the preview queries to the same worker service process.

Example Grid that Runs Jobs in the Service Process

In this example, the grid contains three nodes. All nodes have both the service and compute roles. The Data Integration Service is configured to run jobs in the service process.

The following image shows an example Data Integration Service grid configured to run SQL data service, web service, and preview jobs in the Data Integration Service process:



The Data Integration Service manages requests and runs jobs on the following nodes in the grid:

- On Node1, the master service process manages application deployment and logging. The master service process also acts as a worker service process and completes jobs. The Data Integration Service dispatches a preview request directly to the service process on Node1. The service process creates a DTM instance to run the preview job. SQL data service and web service jobs can also run on Node1.
- On Node2, the Data Integration Service dispatches SQL queries and web service requests directly to the worker service process. The worker service process creates a separate DTM instance to run each job and complete the request. Preview jobs can also run on Node2.
- On Node3, the Data Integration Service dispatches two preview requests from a different user login than the preview1 request directly to the worker service process. The worker service process creates a separate DTM instance to run each preview job. SQL data service and web service jobs can also run on Node3.

Rules and Guidelines for Grids that Run Jobs in the Service Process

Consider the following rules and guidelines when you configure a Data Integration Service grid to run SQL data service, web service, and preview jobs in the Data Integration Service process:

- If the grid contains nodes with the compute role only, the Data Integration Service cannot start.
- If the grid contains nodes with the service role only, jobs that are dispatched to the service process on the node fail to run.
- Configure environment variables for the Data Integration Service processes on the **Processes** view for the service. The Data Integration Service ignores any environment variables configured on the **Compute** view.

Configuring a Grid that Runs Jobs in the Service Process

Configure the Data Integration Service to run jobs in the service process to increase performance. This configuration provides better performance, but decreases stability. This configuration is recommended for running SQL queries and web service requests.

To configure a Data Integration Service grid to run in the service process, perform the following tasks:

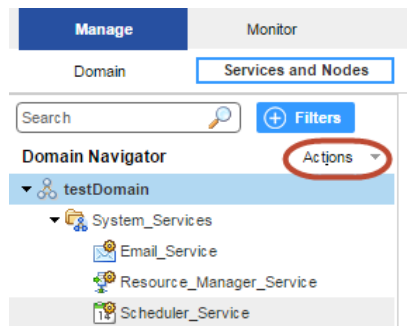
1. Create a grid for the desired jobs.
2. Assign the Data Integration Service to the grid.
3. Configure the Data Integration Service to run jobs in the service process.
4. Configure load balancing for web services.
5. Configure a shared log directory.
6. Optionally, configure properties for each Data Integration Service process that runs on a node in the grid.
7. Optionally, configure compute properties for each DTM instance that can run on a node in the grid.
8. Recycle the Data Integration Service.

Step 1. Create a Grid

To create a grid, create the grid object and assign nodes to the grid. You can assign a node to more than one grid when the Data Integration Service is configured to run jobs in the service process or in separate local processes.

When a Data Integration Service grid runs SQL queries or web service requests, all nodes in the grid must have both the service and compute roles. When you assign nodes to the grid, select nodes that have both roles.

1. In the Administrator tool, click the **Manage** tab.
2. Click the **Services and Nodes** view.
3. In the Domain Navigator, select the domain.



4. On the Navigator Actions menu, click **New > Grid**.
The **Create Grid** dialog box appears.
5. Enter the following properties:

Property	Description
Name	Name of the grid. The name is not case sensitive and must be unique within the domain. It cannot exceed 128 characters or begin with @. It also cannot contain spaces or the following special characters: ` ~ % ^ * + = { } \ ; : ' " / ? . , < > ! () []
Description	Description of the grid. The description cannot exceed 765 characters.
Nodes	Select nodes to assign to the grid.
Path	Location in the Navigator, such as: DomainName/ProductionGrids

6. Click **OK**.

Step 2. Assign the Data Integration Service to the Grid

Assign the Data Integration Service to run on the grid.

1. On the **Services and Nodes** view, select the Data Integration Service in the Domain Navigator.
2. Select the **Properties** tab.
3. In the **General Properties** section, click **Edit**.

The **Edit General Properties** dialog box appears.

4. Next to **Assign**, select **Grid**.
5. Select the grid to assign to the Data Integration Service.
6. Click **OK**.

Step 3. Run Jobs in the Service Process

Configure the Data Integration Service to run jobs in the service process.

1. On the **Services and Nodes** view, select the Data Integration Service in the Domain Navigator.
2. Select the **Properties** tab.
3. In the **Execution Options** section, click **Edit**.
The **Edit Execution Options** dialog box appears.
4. For the **Launch Job Options** property, select **In the service process**.
5. Click **OK**.

Step 4. Configure Load Balancing for Web Services

To run web service requests on a grid, you must configure an external HTTP load balancer. If you do not configure an external HTTP load balancer, the Data Integration Service runs the web service on the node that receives the request.

To configure load balancing, specify the logical URL for the load balancer in the Data Integration Service properties. Then, configure the external load balancer to distribute web service requests to all nodes in the grid that have both the service and compute roles.

1. Complete the following steps in the Administrator tool to configure the Data Integration Service to communicate with the external HTTP load balancer:
 - a. On the **Services and Nodes** view, select the Data Integration Service in the Domain Navigator.
 - b. Select the **Properties** tab.
 - c. In the **Web Service Properties** section, click **Edit**.
The **Edit Web Service Properties** window appears.
 - d. Enter the logical URL for the external HTTP load balancer, and then click **OK**.
2. Configure the external load balancer to distribute requests to all nodes in the grid that have both the service and compute roles.

Step 5. Configure a Shared Log Directory

When the Data Integration Service runs on a grid, a Data Integration Service process can run on each node with the service role. Configure each service process to use the same shared directory for log files. When you configure a shared log directory, you ensure that if the master service process fails over to another node, the new master service process can access previous log files.

1. On the **Services and Nodes** view, select the Data Integration Service in the Domain Navigator.
2. Select the **Processes** tab.
3. Select a node to configure the shared log directory for that node.
4. In the **Logging Options** section, click **Edit**.
The **Edit Logging Options** dialog box appears.
5. Enter the location to the shared log directory.

6. Click **OK**.
7. Repeat the steps for each node listed in the **Processes** tab to configure each service process with identical absolute paths to the shared directories.

RELATED TOPICS:

- [“Log Directory” on page 97](#)

Step 6. Optionally Configure Process Properties

Optionally, configure the Data Integration Service process properties for each node with the service role in the grid. You can configure the service process properties differently for each node.

To configure properties for the Data Integration Service processes, click the **Processes** view. Select a node with the service role to configure properties specific to that node.

RELATED TOPICS:

- [“Data Integration Service Process Properties” on page 67](#)

Step 7. Optionally Configure Compute Properties

You can configure the compute properties that the execution Data Transformation Manager (DTM) uses when it runs jobs. When the Data Integration Service runs on a grid, DTM processes run jobs on each node with the compute role. You can configure the compute properties differently for each node.

To configure compute properties for the DTM, click the **Compute** view. Select a node with the compute role to configure properties specific to DTM instances that run on the node. For example, you can configure a different temporary directory for each node.

When a Data Integration Service grid runs jobs in the Data Integration Service process, you can configure the execution options on the **Compute** view. If you configure environment variables on the **Compute** view, they are ignored.

RELATED TOPICS:

- [“Data Integration Service Compute Properties” on page 70](#)

Step 8. Recycle the Data Integration Service

After you change Data Integration Service properties, you must recycle the service for the changed properties to take effect.

To recycle the service, select the service in the Domain Navigator and click **Recycle the Service**.

Grid for Jobs that Run in Local Mode

Configure the Data Integration Service to run jobs in separate DTM processes on the local node to increase stability. Use this configuration when the Data Integration Service grid runs mappings, profiles, and workflows. All nodes in the grid must have both the service and compute roles.

When you enable a Data Integration Service that runs on a grid, one service process starts on each node with the service role in the grid. The Data Integration Service designates one service process as the master service process, and designates the remaining service processes as worker service processes. When a

worker service process starts, it registers itself with the master service process so that the master is aware of the worker.

The master service process manages application deployments, logging, job requests, and the dispatch of mappings to worker service processes. The worker service processes optimize and compile mapping and preview jobs. The worker service processes create separate DTM processes to run jobs. The master service process also acts as a worker service process and runs jobs.

The Data Integration Service balances the workload across the nodes in the grid based on the following job types:

Workflows

When you run a workflow instance, the master service process runs the workflow instance and non-mapping tasks. The master service process uses round robin to dispatch each mapping within a Mapping task to a worker service process. The worker service process optimizes and compiles the mapping. The worker service process then creates a DTM instance within a separate DTM process to run the mapping.

Deployed mappings

When you run a deployed mapping, the master service process uses round robin to dispatch each mapping to a worker service process. The worker service process optimizes and compiles the mapping. The worker service process then creates a DTM instance within a separate DTM process to run the mapping.

Profiles

When you run a profile, the master service process converts the profiling job into multiple mapping jobs based on the advanced profiling properties of the Data Integration Service. The master service process then uses round robin to dispatch the mappings across the worker service processes. The worker service process optimizes and compiles the mapping. The worker service process then creates a DTM instance within a separate DTM process to run the mapping.

Ad hoc jobs, with the exception of profiles

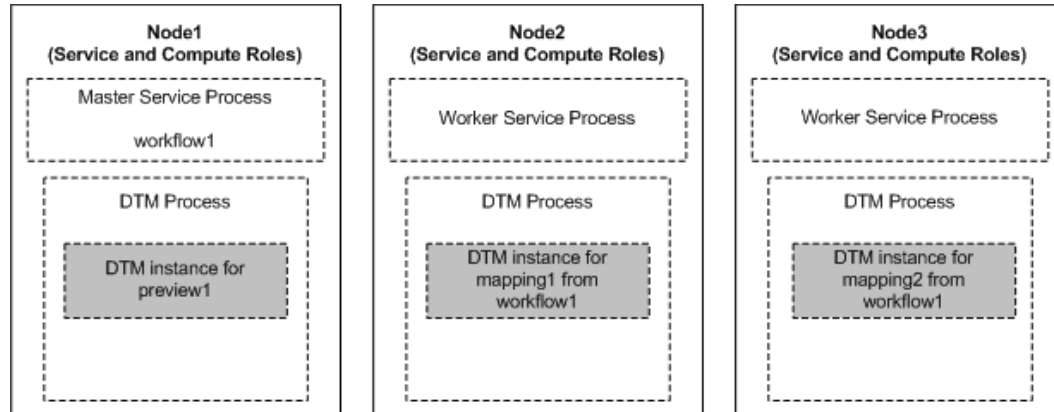
When you run ad hoc jobs, with the exception of profiles, the Data Integration Service uses round robin to dispatch the first request directly to a worker service process. Ad hoc jobs include mappings run from the Developer tool or previews, scorecards, or drill downs on profile results run from the Developer tool or Analyst tool. To ensure faster throughput, the Data Integration Service bypasses the master service process. The worker service process creates a DTM instance within a separate DTM process to run the job. When you run additional ad hoc jobs from the same login, the Data Integration Service dispatches the requests to the same worker service process.

Note: Informatica does not recommend running SQL queries or web service requests on a Data Integration Service grid that is configured to run jobs in separate local processes. SQL data service and web service jobs typically achieve better performance when the Data Integration Service runs jobs in the service process. For web service requests, you must configure the external HTTP load balancer to distribute requests to nodes that have both the service and compute roles.

Example Grid that Runs Jobs in Local Mode

In this example, the grid contains three nodes. All nodes have both the service and compute roles. The Data Integration Service is configured to run jobs in separate local processes.

The following image shows an example Data Integration Service grid configured to run mapping, profile, workflow, and ad hoc jobs in separate local processes:



The Data Integration Service manages requests and runs jobs on the following nodes in the grid:

- On Node1, the master service process runs the workflow instance and non-mapping tasks. The master service process dispatches mappings included in Mapping tasks from workflow1 to the worker service processes on Node2 and Node3. The master service process also acts as a worker service process and completes jobs. The Data Integration Service dispatches a preview request directly to the service process on Node1. The service process creates a DTM instance within a separate DTM process to run the preview job. Mapping and profile jobs can also run on Node1.
- On Node2, the worker service process creates a DTM instance within a separate DTM process to run mapping1 from workflow1. Ad hoc jobs can also run on Node2.
- On Node3, the worker service process creates a DTM instance within a separate DTM process to run mapping2 from workflow1. Ad hoc jobs can also run on Node3.

Rules and Guidelines for Grids that Run Jobs in Local Mode

Consider the following rules and guidelines when you configure a Data Integration Service grid to run jobs in separate local processes:

- If the grid contains nodes with the compute role only, the Data Integration Service cannot start.
- If the grid contains nodes with the service role only, jobs that are dispatched to the service process on the node fail to run.
- Configure environment variables for the Data Integration Service processes on the **Processes** view for the service. The Data Integration Service ignores any environment variables configured on the **Compute** view.

Configuring a Grid that Runs Jobs in Local Mode

When a Data Integration Service grid runs mappings, profiles, and workflows, you can configure the Data Integration Service to run jobs in separate DTM processes on local nodes.

To configure a Data Integration Service grid to run mappings, profiles, and workflows in separate local processes, perform the following tasks:

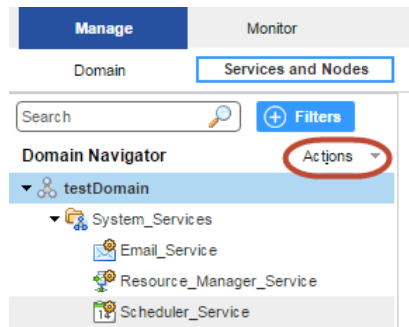
1. Create a grid for mappings, profiles, and workflows that run in separate local processes.
2. Assign the Data Integration Service to the grid.
3. Configure the Data Integration Service to run jobs in separate local processes.
4. Configure a shared log directory.
5. Optionally, configure properties for each Data Integration Service process that runs on a node in the grid.
6. Optionally, configure compute properties for each DTM instance that can run on a node in the grid.
7. Recycle the Data Integration Service.

Step 1. Create a Grid

To create a grid, create the grid object and assign nodes to the grid. You can assign a node to more than one grid when the Data Integration Service is configured to run jobs in the service process or in separate local processes.

When a Data Integration Service grid runs mappings, profiles, and workflows in separate local processes, all nodes in the grid must have both the service and compute roles. When you assign nodes to the grid, select nodes that have both roles.

1. In the Administrator tool, click the **Manage** tab.
2. Click the **Services and Nodes** view.
3. In the Domain Navigator, select the domain.



4. On the Navigator Actions menu, click **New > Grid**.
The **Create Grid** dialog box appears.

5. Enter the following properties:

Property	Description
Name	Name of the grid. The name is not case sensitive and must be unique within the domain. It cannot exceed 128 characters or begin with @. It also cannot contain spaces or the following special characters: ` ~ % ^ * + = { } \ ; : ' " / ? . , < > ! () []
Description	Description of the grid. The description cannot exceed 765 characters.
Nodes	Select nodes to assign to the grid.
Path	Location in the Navigator, such as: DomainName/ProductionGrids

6. Click **OK**.

Step 2. Assign the Data Integration Service to the Grid

Assign the Data Integration Service to run on the grid.

1. On the **Services and Nodes** view, select the Data Integration Service in the Domain Navigator.
2. Select the **Properties** tab.
3. In the **General Properties** section, click **Edit**.
The **Edit General Properties** dialog box appears.
4. Next to **Assign**, select **Grid**.
5. Select the grid to assign to the Data Integration Service.
6. Click **OK**.

Step 3. Run Jobs in Separate Local Processes

Configure the Data Integration Service to run jobs in separate local processes.

1. On the **Services and Nodes** view, select the Data Integration Service in the Domain Navigator.
2. Select the **Properties** tab.
3. In the **Execution Options** section, click **Edit**.
The **Edit Execution Options** dialog box appears.
4. For the **Launch Job Options** property, select **In separate local processes**.
5. Click **OK**.

Step 4. Configure a Shared Log Directory

When the Data Integration Service runs on a grid, a Data Integration Service process can run on each node with the service role. Configure each service process to use the same shared directory for log files. When you configure a shared log directory, you ensure that if the master service process fails over to another node, the new master service process can access previous log files.

1. On the **Services and Nodes** view, select the Data Integration Service in the Domain Navigator.

2. Select the **Processes** tab.
3. Select a node to configure the shared log directory for that node.
4. In the **Logging Options** section, click **Edit**.
The **Edit Logging Options** dialog box appears.
5. Enter the location to the shared log directory.
6. Click **OK**.
7. Repeat the steps for each node listed in the **Processes** tab to configure each service process with identical absolute paths to the shared directories.

RELATED TOPICS:

- [“Log Directory” on page 97](#)

Step 5. Optionally Configure Process Properties

Optionally, configure the Data Integration Service process properties for each node with the service role in the grid. You can configure the service process properties differently for each node.

To configure properties for the Data Integration Service processes, click the **Processes** view. Select a node with the service role to configure properties specific to that node.

RELATED TOPICS:

- [“Data Integration Service Process Properties” on page 67](#)

Step 6. Optionally Configure Compute Properties

You can configure the compute properties that the execution Data Transformation Manager (DTM) uses when it runs jobs. When the Data Integration Service runs on a grid, DTM processes run jobs on each node with the compute role. You can configure the compute properties differently for each node.

To configure compute properties for the DTM, click the **Compute** view. Select a node with the compute role to configure properties specific to DTM instances that run on the node. For example, you can configure a different temporary directory for each node.

When a Data Integration Service grid runs jobs in separate local processes, you can configure the execution options on the **Compute** view. If you configure environment variables on the **Compute** view, they are ignored.

RELATED TOPICS:

- [“Data Integration Service Compute Properties” on page 70](#)

Step 7. Recycle the Data Integration Service

After you change Data Integration Service properties, you must recycle the service for the changed properties to take effect.

To recycle the service, select the service in the Domain Navigator and click **Recycle the Service**.

Grid for Jobs that Run in Remote Mode

When a Data Integration Service grid runs mappings, profiles, and workflows, you can configure the service to run jobs in separate DTM processes on remote nodes. The nodes in the grid can have a different combination of roles.

A Data Integration Service grid uses the following components to run jobs in separate remote processes:

Master service process

When you enable a Data Integration Service that runs on a grid, one service process starts on each node with the service role in the grid. The Data Integration Service designates one service process as the master service process. The master service process manages application deployments, logging, job requests, and the dispatch of mappings to worker service processes for optimization and compilation. The master service process also acts as a worker service process and can optimize and compile mappings.

Worker service processes

The Data Integration Service designates the remaining service processes as worker service processes. When a worker service process starts, it registers itself with the master service process so that the master is aware of the worker. A worker service process optimizes and compiles mappings, and then generates a grid task. A grid task is a job request sent by the worker service process to the Service Manager on the master compute node.

Service Manager on the master compute node

When you enable a Data Integration Service that runs on a grid, the Data Integration Service designates one node with the compute role as the master compute node.

The Service Manager on the master compute node performs the following functions to determine the optimal worker compute node to run the mapping:

- Communicates with the Resource Manager Service to manage the grid of available compute nodes. When the Service Manager on a node with the compute role starts, the Service Manager registers the node with the Resource Manager Service.
- Orchestrates worker service process requests and dispatches mappings to worker compute nodes.

The master compute node also acts as a worker compute node and can run mappings.

DTM processes on worker compute nodes

The Data Integration Service designates the remaining nodes with the compute role as worker compute nodes. The Service Manager on a worker compute node runs mappings in separate DTM processes started within containers.

Supported Node Roles

When a Data Integration Service grid runs jobs in separate remote processes, the nodes in the grid can contain the service role only, the compute role only, or both the service and compute roles.

A Data Integration Service grid that runs jobs in separate remote processes can contain nodes with the following roles:

Service role

A Data Integration Service process runs on each node with the service role. Service components within the Data Integration Service process run workflows and profiles, and perform mapping optimization and compilation.

Compute role

DTM processes run on each node with the compute role. The DTM processes run deployed mappings, mappings run by Mapping tasks within a workflow, and mappings converted from a profile.

Both service and compute roles

A Data Integration Service process and DTM processes run on each node with both the service and compute roles. At least one node with both service and compute roles is required to run ad hoc jobs, with the exception of profiles. Ad hoc jobs include mappings run from the Developer tool or previews, scorecards, or drill downs on profile results run from the Developer tool or Analyst tool. The Data Integration Service runs these job types in separate DTM processes on the local node.

In addition, nodes with both roles can complete all of the tasks that a node with the service role only or a node with the compute role only can complete. For example, a workflow can run on a node with the service role only or on a node with both the service and compute roles. A deployed mapping can run on a node with the compute role only or on a node with both the service and compute roles.

The following table lists the job types that run on nodes based on the node role:

Job Type	Service Role	Compute Role	Service and Compute Roles
Perform mapping optimization and compilation.	Yes	-	Yes
Run deployed mappings.	-	Yes	Yes
Run workflows.	Yes	-	Yes
Run mappings included in workflow Mapping tasks.	-	Yes	Yes
Run profiles.	Yes	-	Yes
Run mappings converted from profiles.	-	Yes	Yes
Run ad hoc jobs, with the exception of profiles, from the Analyst tool or the Developer tool.	-	-	Yes

Note: If you associate a Content Management Service with the Data Integration Service to run mappings that read reference data, each node in the grid must have both the service and compute roles.

Job Types

When a Data Integration Service grid runs jobs in separate remote processes, how the Data Integration Service runs each job depends on the job type.

The Data Integration Service balances the workload across the nodes in the grid based on the following job types:

Workflows

When you run a workflow instance, the master service process runs the workflow instance and non-mapping tasks. The master service process uses round robin to dispatch each mapping within a Mapping task to a worker service process. The LDTM component of the worker service process optimizes and compiles the mapping. The worker service process then communicates with the master compute node to dispatch the compiled mapping to a separate DTM process running on a worker compute node.

Deployed mappings

When you run a deployed mapping, the master service process uses round robin to dispatch each mapping to a worker service process. The LDTM component of the worker service process optimizes and compiles the mapping. The worker service process then communicates with the master compute node to dispatch the compiled mapping to a separate DTM process running on a worker compute node.

Profiles

When you run a profile, the master service process converts the profiling job into multiple mapping jobs based on the advanced profiling properties of the Data Integration Service. The master service process then distributes the mappings across the worker service processes. The LDTM component of the worker service process optimizes and compiles the mapping. The worker service process then communicates with the master compute node to dispatch the compiled mapping to a separate DTM process running on a worker compute node.

Ad hoc jobs, with the exception of profiles

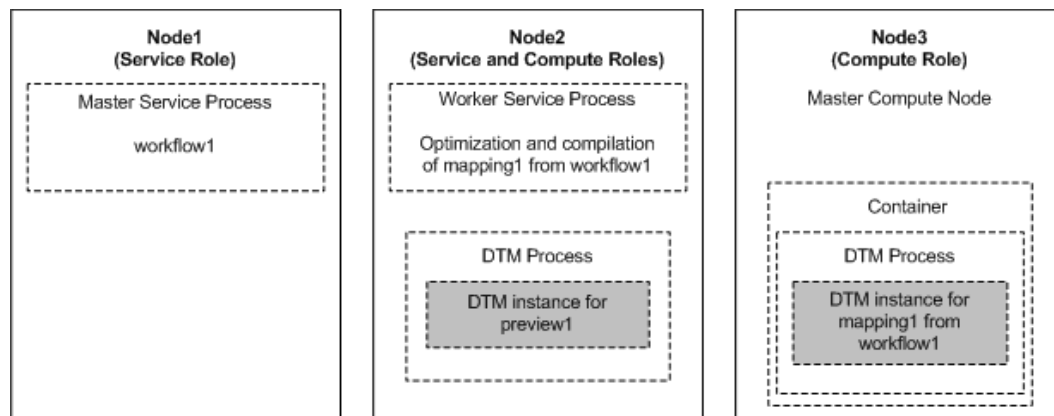
When you run an ad hoc job, with the exception of profiles, the Data Integration Service uses round robin to dispatch the first request directly to a worker service process that runs on a node with both the service and compute roles. The worker service process runs the job in a separate DTM process on the local node. To ensure faster throughput, the Data Integration Service bypasses the master service process. When you run additional ad hoc jobs from the same login, the Data Integration Service dispatches the requests to the same worker service process.

Note: You cannot run SQL queries or web service requests on a Data Integration Service grid that is configured to run jobs in separate remote processes.

Example Grid that Runs Jobs in Remote Mode

In this example, the grid contains three nodes. Node1 has the service role only. Node2 has both the service and compute roles. Node3 has the compute role only. The Data Integration Service is configured to run jobs in separate remote processes.

The following image shows an example Data Integration Service grid configured to run mapping, profile, workflow, and ad hoc jobs in separate remote processes:



The Data Integration Service manages requests and runs jobs on the following nodes in the grid:

- On Node1, the master service process runs the workflow instance and non-mapping tasks. The master service process dispatches a mapping included in a Mapping task from workflow1 to the worker service process on Node2. The master service process also acts as a worker service process and can optimize and compile mappings. Profile jobs can also run on Node1.

- On Node2, the worker service process optimizes and compiles the mapping. The worker service process then communicates with the master compute node on Node3 to dispatch the compiled mapping to a worker compute node. The Data Integration Service dispatches a preview request directly to the worker service process on Node2. The service process creates a DTM instance within a separate DTM process on Node2 to run the preview job. Node2 also serves as a worker compute node and can run compiled mappings.
- On Node3, the Service Manager on the master compute node orchestrates requests to run mappings. The master compute node also acts as a worker compute node and runs the mapping from workflow1 in a separate DTM process started within a container.

Rules and Guidelines for Grids that Run Jobs in Remote Mode

Consider the following rules and guidelines when you configure a Data Integration Service grid to run jobs in separate remote processes:

- The grid must contain at least one node with both the service and compute roles to run an ad hoc job, with the exception of profiles. The Data Integration Service runs these job types in a separate DTM process on the local node. Add additional nodes with both the service and compute roles so that these job types can be distributed to service processes running on other nodes in the grid.
- To support failover for the Data Integration Service, the grid must contain at least two nodes that have the service role.
- If you associate a Content Management Service with the Data Integration Service to run mappings that read reference data, each node in the grid must have both the service and compute roles.
- The grid cannot include two nodes that are defined on the same host machine.
- Informatica does not recommend assigning multiple Data Integration Services to the same grid nor assigning one node to multiple Data Integration Service grids.

If a worker compute node is shared across multiple grids, mappings dispatched to the node might fail due to an over allocation of the node's resources. If a master compute node is shared across multiple grids, the log events for the master compute node are also shared and might become difficult to troubleshoot.

Recycle the Service When Jobs Run in Remote Mode

You must recycle the Data Integration Service if you change a service property or if you update the role for a node assigned to the service or to the grid on which the service runs. You must recycle the service for additional reasons when the service is on a grid and is configured to run jobs in separate remote processes.

When a Data Integration Service grid runs jobs in separate remote processes, recycle the Data Integration Service after you complete the following actions:

- Override compute node attributes for a node assigned to the grid.
- Add or remove a node from the grid.
- Shut down or restart a node assigned to the grid.

To recycle the Data Integration Service, select the service in the Domain Navigator and click **Recycle the Service**.

Configuring a Grid that Runs Jobs in Remote Mode

When a Data Integration Service grid runs mappings, profiles, and workflows, you can configure the Data Integration Service to run jobs in separate DTM processes on remote nodes.

To configure a Data Integration Service grid to run mappings, profiles, and workflows in separate remote processes, perform the following tasks:

1. Update the roles for the nodes in the grid.
2. Create a grid for mappings, profiles, and workflows that run in separate remote processes.
3. Assign the Data Integration Service to the grid.
4. Configure the Data Integration Service to run jobs in separate remote processes.
5. Enable the Resource Manager Service.
6. Configure a shared log directory.
7. Optionally, configure properties for each Data Integration Service process that runs on a node with the service role.
8. Optionally, configure compute properties for each DTM instance that can run on a node with the compute role.
9. Recycle the Data Integration Service.

Step 1. Update Node Roles

By default, each node has both the service and compute roles. You can update the roles of each node that you plan to add to the grid. Enable only the service role to dedicate a node to running the Data Integration Service process. Enable only the compute role to dedicate a node to running mappings.

At least one node in the grid must have both the service and compute roles to run ad hoc jobs, with the exception of profiles.

Note: Before you can disable the service role on a node, you must shut down all application service processes running on the node and remove the node as a primary or back-up node for any application service. You cannot disable the service role on a gateway node.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select a node that you plan to add to the grid.
3. In the **Properties** view, click **Edit** for the general properties.
The **Edit General Properties** dialog box appears.
4. Select or clear the service and compute roles to update the node role.
5. Click **OK**.
6. If you disabled the compute role, the **Disable Compute Role** dialog box appears. Perform the following steps:
 - a. Select one of the following modes to disable the compute role:
 - **Complete.** Allows jobs to run to completion before disabling the role.
 - **Stop.** Stops all jobs and then disables the role.
 - **Abort.** Tries to stop all jobs before aborting them and disabling the role.
 - b. Click **OK**.
7. Repeat the steps to update the node role for each node that you plan to add to the grid.

Step 2. Create a Grid

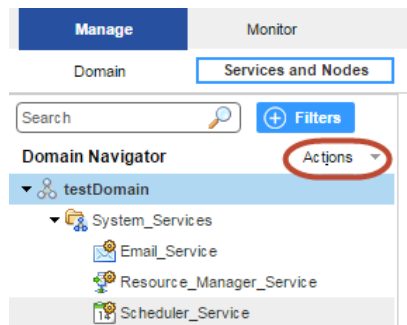
To create a grid, create the grid object and assign nodes to the grid. You can assign a node to one grid when the Data Integration Service is configured to run jobs in separate remote processes.

When a Data Integration Service grid runs mappings, profiles, and workflows in separate remote processes, the grid can include the following nodes:

- Any number of nodes with the service role only.
- Any number of nodes with the compute role only.
- At least one node with both the service and compute roles to run previews and to run ad hoc jobs, with the exception of profiles.

If you associate a Content Management Service with the Data Integration Service to run mappings that read reference data, each node in the grid must have both the service and compute roles.

1. In the Administrator tool, click the **Manage** tab.
2. Click the **Services and Nodes** view.
3. In the Domain Navigator, select the domain.



4. On the Navigator Actions menu, click **New > Grid**.
The **Create Grid** dialog box appears.
5. Enter the following properties:

Property	Description
Name	Name of the grid. The name is not case sensitive and must be unique within the domain. It cannot exceed 128 characters or begin with @. It also cannot contain spaces or the following special characters: ` ~ % ^ * + = { } \ ; : ' " / ? . , < > ! () []
Description	Description of the grid. The description cannot exceed 765 characters.
Nodes	Select nodes to assign to the grid.
Path	Location in the Navigator, such as: DomainName/ProductionGrids

6. Click **OK**.

Step 3. Assign the Data Integration Service to the Grid

Assign the Data Integration Service to run on the grid.

1. On the **Services and Nodes** view, select the Data Integration Service in the Domain Navigator.
2. Select the **Properties** tab.
3. In the **General Properties** section, click **Edit**.
The **Edit General Properties** dialog box appears.
4. Next to **Assign**, select **Grid**.
5. Select the grid to assign to the Data Integration Service.
6. Click **OK**.

Step 4. Run Jobs in Separate Remote Processes

Configure the Data Integration Service to run jobs in separate remote processes.

1. On the **Services and Nodes** view, select the Data Integration Service in the Domain Navigator.
2. Select the **Properties** tab.
3. In the **Execution Options** section, click **Edit**.
The **Edit Execution Options** dialog box appears.
4. For the **Launch Job Options** property, select **In separate remote processes**.
5. Click **OK**.

Step 5. Enable the Resource Manager Service

By default, the Resource Manager Service is disabled. You must enable the Resource Manager Service so that the Data Integration Service grid can run jobs in separate remote processes.

1. On the **Services and Nodes** view, expand the **System_Services** folder.
2. Select the Resource Manager Service in the Domain Navigator, and click **Recycle the Service**.

Step 6. Configure a Shared Log Directory

When the Data Integration Service runs on a grid, a Data Integration Service process can run on each node with the service role. Configure each service process to use the same shared directory for log files. When you configure a shared log directory, you ensure that if the master service process fails over to another node, the new master service process can access previous log files.

1. On the **Services and Nodes** view, select the Data Integration Service in the Domain Navigator.
2. Select the **Processes** tab.
3. Select a node to configure the shared log directory for that node.
4. In the **Logging Options** section, click **Edit**.
The **Edit Logging Options** dialog box appears.
5. Enter the location to the shared log directory.
6. Click **OK**.
7. Repeat the steps for each node listed in the **Processes** tab to configure each service process with identical absolute paths to the shared directories.

RELATED TOPICS:

- [“Log Directory” on page 97](#)

Step 7. Optionally Configure Process Properties

Optionally, configure the Data Integration Service process properties for each node with the service role in the grid. You can configure the service process properties differently for each node.

To configure properties for the Data Integration Service processes, click the **Processes** view. Select a node with the service role to configure properties specific to that node.

RELATED TOPICS:

- [“Data Integration Service Process Properties” on page 67](#)

Step 8. Optionally Configure Compute Properties

You can configure the compute properties that the execution Data Transformation Manager (DTM) uses when it runs jobs. When the Data Integration Service runs on a grid, DTM processes run jobs on each node with the compute role. You can configure the compute properties differently for each node.

To configure compute properties for the DTM, click the **Compute** view. Select a node with the compute role to configure properties specific to DTM processes that run on the node. For example, you can configure a different temporary directory or different environment variable values for each node.

RELATED TOPICS:

- [“Data Integration Service Compute Properties” on page 70](#)

Step 9. Recycle the Data Integration Service

After you change Data Integration Service properties, you must recycle the service for the changed properties to take effect.

To recycle the service, select the service in the Domain Navigator and click **Recycle the Service**.

Logs for Jobs that Run in Remote Mode

When a Data Integration Service grid runs a mapping in a separate remote process, the worker service process that optimizes and compiles the mapping writes log events to one log file. The DTM process that runs the mapping writes log events to another log file. When you access the mapping log, the Data Integration Service consolidates the two files into a single log file.

The worker service process writes to a log file in the shared log directory configured for each Data Integration Service process. The DTM process writes to a temporary log file in the log directory configured for the worker compute node. When the DTM process finishes running the mapping, it sends the log file to the master Data Integration Service process. The master service process writes the DTM log file to the shared log directory configured for the Data Integration Service processes. The DTM process then removes the temporary DTM log file from the worker compute node.

When you access the mapping log using the Administrator tool or the `infacmd ms getRequestLog` command, the Data Integration Service consolidates the two files into a single log file.

The consolidated log file contains the following types of messages:

LDTM messages written by the worker service process on the service node

The first section of the mapping log contains LDTM messages about mapping optimization and compilation and about generating the grid task written by the worker service process on the service node.

The grid task messages include the following message that indicates the location of the log file written by the DTM process on the worker compute node:

```
INFO: [GCL 5] The grid task [gtid-1443479776986-1-79777626-99] cluster logs can be found at [./1443479776986/taskletlogs/gtid-1443479776986-1-79777626-99].
```

The listed directory is a subdirectory of the following default log directory configured for the worker compute node:

```
<Informatica installation directory>/logs/<node name>/dtmLogs/
```

DTM messages written by the DTM process on the compute node

The second section of the mapping log contains messages about mapping execution written by the DTM process on the worker compute node.

The DTM section of the log begins with the following lines which indicate the name of the worker compute node that ran the mapping:

```
###  
### <MyWorkerComputeNodeName>  
###  
  
### Start Grid Task [gtid-1443479776986-1-79777626-99] Segment [s0] Tasklet [t-0]  
Attempt [1]
```

The DTM section of the log concludes with the following line:

```
### End Grid Task [gtid-1443479776986-1-79777626-99] Segment [s0] Tasklet [t-0]  
Attempt [1]
```

Override Compute Node Attributes to Increase Concurrent Jobs

You can override compute node attributes to increase the number of concurrent jobs that run on the node. You can override the maximum number of cores and the maximum amount of memory that the Resource Manager Service can allocate for jobs that run on the compute node. The default values are the actual number of cores and memory available on the machine.

When the Data Integration Service runs jobs in separate remote processes, by default a machine that represents a compute node requires at least five cores and 2.5 GB of memory to initialize a container to start a DTM process. If any compute node assigned to the grid has fewer than five cores, then that number is used as the minimum number of cores required to initialize a container. For example, if a compute node assigned to the grid has three cores, then each compute node in that grid requires at least three cores and 2.5 GB of memory to initialize a container.

You might want to override compute node attributes to increase the number of concurrent jobs when the following conditions are true:

- You run long-running jobs on the grid.
- The Data Integration Service cannot reuse DTM processes because you run jobs from different deployed applications.
- Job concurrency is more important than the job execution time.

For example, you have configured a Data Integration Service grid that contains a single compute node. You want to concurrently run two mappings from different applications. Because the mappings are in different applications, the Data Integration Service runs the mappings in separate DTM processes, which requires two containers. The machine that represents the compute node has four cores. Only one container can be

initialized, and so the two mappings cannot run concurrently. You can override the compute node attributes to specify that the Resource Manager Service can allocate eight cores for jobs that run on the compute node. Then, two DTM processes can run at the same time and the two mappings can run concurrently.

Use caution when you override compute node attributes. Specify values that are close to the actual resources available on the machine so that you do not overload the machine. Configure the values such that the memory requirements for the total number of concurrent mappings do not exceed the actual resources. A mapping that runs in one thread requires one core. A mapping can use the amount of memory configured in the **Maximum Memory Per Request** property for the Data Integration Service modules.

To override compute node attributes, run the `infacmd rms SetComputeNodeAttributes` command for a specified node.

You can override the following options:

Option	Argument	Description
-MaxCores -mc	max_number_of_cores_to_allocate	Optional. Maximum number of cores that the Resource Manager Service can allocate for jobs that run on the compute node. A compute node requires at least five available cores to initialize a container to start a DTM process. If any compute node assigned to the grid has fewer than five cores, then that number is used as the minimum number of cores required to initialize a container. By default, the maximum number of cores is the actual number of cores available on the machine.
-MaxMem -mm	max_memory_in_mb_to_allocate	Optional. Maximum amount of memory in megabytes that the Resource Manager Service can allocate for jobs that run on the compute node. A compute node requires at least 2.5 GB of memory to initialize a container to start a DTM process. By default, the maximum memory is the actual memory available on the machine.

After you override compute node attributes, you must recycle the Data Integration Service for the changes to take effect. To reset an option to its default value, specify `-1` as the value.

Grid and Content Management Service

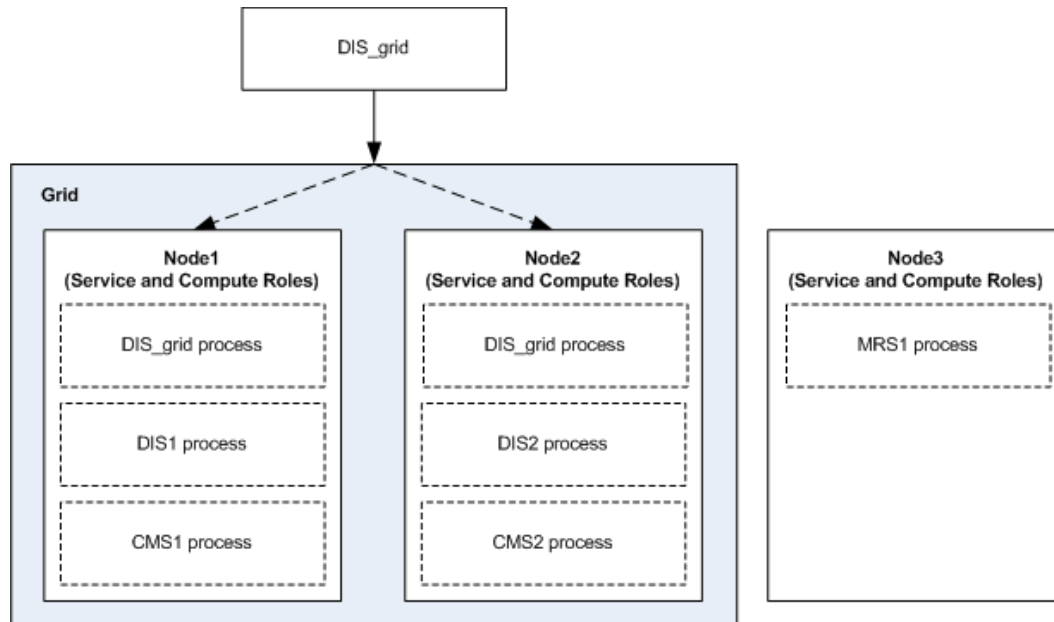
You must associate a Content Management Service with a Data Integration Service to run mappings that read reference data. To associate a Content Management Service with a Data Integration Service that runs on a grid, you must create and configure multiple Content Management Services and multiple Data Integration Services.

To associate a Content Management Service with a Data Integration Service that runs on a grid, perform the following tasks:

1. Create a grid where each node in the grid includes both the service and compute roles.
2. Create a Data Integration Service and assign the service to run on the grid. Configure the Data Integration Service to run jobs in separate local or remote processes.
3. Create a Content Management Service and a new Data Integration Service to run on each node in the grid.
4. Associate each Content Management Service with the Data Integration Service that runs on the same node.

5. Associate each Content Management Service and Data Integration Service with the same Model Repository Service that the Data Integration Service on grid is associated with.
The Content Management Service provides reference data information to all Data Integration Service processes that run on the same node and that are associated with the same Model Repository Service.

The following image shows an example domain that contains three nodes. A total of three Data Integration Services, two Content Management Services, and one Model Repository Service exist in the domain:



The following services run in the domain:

- A Data Integration Service named DIS_grid. DIS_grid is assigned to run on the grid. A DIS_grid process runs on each node in the grid. When you run a job on the grid, the DIS_grid processes run the job.
- A Data Integration Service named DIS1 and a Content Management Service named CMS1 assigned to run on Node1. CMS1 is associated with DIS1.
- A Data Integration Service named DIS2 and a Content Management Service named CMS2 assigned to run on Node2. CMS2 is associated with DIS2.
- A Model Repository Service named MRS1 assigned to run on Node3. Each Data Integration Service and Content Management Service in the domain is associated with MRS1. In this example, the Model Repository Service runs on a node outside of the Data Integration Service grid. However, the Model Repository Service can run on any node in the domain.

Maximum Number of Concurrent Jobs on a Grid

You can set the number of deployed and on-demand jobs that the Data Integration Service process can run concurrently.

You can configure the following Data Integration Service properties:

Maximum on-demand execution pool size

Determine the maximum number of on-demand jobs that you can run concurrently. On-demand jobs include data previews, profiling jobs, SQL queries, and web service requests. The Data Integration

Service immediately runs on-demand jobs if enough resources are available. Otherwise, the Data Integration Service rejects the job. The default value is 10.

Maximum native batch execution pool size

Determines the maximum number of jobs that you can run concurrently in the native environment. The Data Integration Service moves deployed native jobs from the queue to the native batch pool when enough resources are available. The default value is 10.

Maximum Hadoop batch execution pool size

Determines the maximum number of deployed jobs that you can run concurrently in the Hadoop environment. The Data Integration Service moves deployed Hadoop jobs from the queue to the Hadoop batch pool when enough resources are available. The default value is 100.

When the Data Integration Service runs on a grid, the maximum number of deployed and on-demand jobs that can run concurrently across the grid are calculated as follows:

Maximum on-demand pool size * Number of running service processes

Maximum native batch pool size * Number of running service processes

Maximum Hadoop batch pool size * Number of running service processes

For example, a Data Integration Service grid includes three running service processes. If you set the Hadoop batch pool size to 10, each Data Integration Service process can run up to 10 deployed Hadoop jobs concurrently. A total of 30 deployed Hadoop jobs can run concurrently on the grid. If you try to run more than 30 Hadoop jobs, the Data Integration Service queues the jobs until there is space in the pool.

When you increase the pool size values, the Data Integration Service uses more hardware resources such as CPU, memory, and system I/O. Set these values based on the resources available on the nodes in the grid. For example, consider the number of CPUs on the machines where Data Integration Service processes run and the amount of memory that is available to the Data Integration Service.

Note: If the Data Integration Service grid runs jobs in separate remote processes, additional concurrent jobs might not run on compute nodes after you increase the value of these properties. You might need to override compute node attributes to increase the number of concurrent jobs on each compute node. For more information, see [“Override Compute Node Attributes to Increase Concurrent Jobs” on page 149](#).

Editing a Grid

You can edit a grid to change the description, add nodes to the grid, or remove nodes from the grid.

Before you remove a node from the grid, disable the Data Integration Service process running on the node.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. Select the grid in the Domain Navigator.
3. To edit the grid, click **Edit** in the **Grid Details** section.

You can change the grid description, add nodes to the grid, or remove nodes from the grid.

4. Click **OK**.
5. If you added or removed a node from a Data Integration Service grid configured to run jobs in separate remote processes, recycle the Data Integration Service for the changes to take effect.

Deleting a Grid

You can delete a grid from the domain if the grid is no longer required.

Before you delete a grid, disable the Data Integration Service running on the grid.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. Select the grid in the Domain Navigator.
3. Select **Actions** > **Delete**.

Troubleshooting a Grid

I enabled a Data Integration Service that runs on a grid, but one of the service processes failed to start.

When you enable a Data Integration Service that runs on a grid, a service process starts on each node in the grid that has the service role. A service process might fail to start for the following reasons:

- The node does not have the service role.
Enable the service role on the node, and then enable the service process running on that node.
- Another process running on the machine is using the HTTP port number assigned to the service process.
On the **Processes** view for the Data Integration Service, enter a unique HTTP port number for the service process. Then, enable the service process running on that node.

A job failed to run on a Data Integration Service grid. Which logs do I review?

If the Data Integration Service grid is configured to run jobs in the service process or in separate local processes, review the following logs in this order:

1. Job log accessible from the **Monitor** tab.
Includes log events about how the DTM instance runs the job.
2. Data Integration Service log accessible from the **Service** view of the **Logs** tab.
Includes log events about service configuration, processing, and failures.

If the Data Integration Service grid is configured to run jobs in separate remote processes, additional components write log files. Review the following logs in this order:

1. Job log accessible from the **Monitor** tab.
Includes log events about how the DTM instance runs the job.
2. Data Integration Service log accessible from the **Service** view of the **Logs** tab.
Includes log events about service configuration, processing, and failures. The Data Integration Service log includes the following message which indicates the host name and port number of the master compute node:

```
INFO: [GRIDCAL_0204] The Integration Service [<MyDISName>] elected a new master  
compute node [<HostName>:<PortNumber>].
```
3. Master compute node log accessible in the `cadi_services_0.log` file located in the log directory configured for the master compute node.

Includes log events written by the Service Manager on the master compute node about managing the grid of compute nodes and orchestrating worker service process requests. The master compute node logs are not accessible from the Administrator tool.

4. Resource Manager Service log accessible from the **Service** view of the **Logs** tab.

Includes log events about service configuration and processing and about nodes with the compute role that register with the service.

5. Container management log accessible from the **Domain** view of the **Logs** tab. Select **Container Management** for the category.

Includes log events about how the Service Manager manages containers on nodes with the compute role.

A mapping that ran in a separate remote process has an incomplete log file.

When a mapping runs on a Data Integration Service grid configured to run jobs in separate remote processes, the Data Integration Service writes two files for the mapping log. The worker service process that optimizes and compiles the mapping on the service node writes log events to one log file. The DTM process that runs the mapping on the compute node writes log events to another log file. When you access the mapping log, the Data Integration Service consolidates the two files into a single log file.

A mapping log might be incomplete for the following reasons:

- The mapping is still running.

When a DTM process finishes running a mapping, it sends the log file to the master Data Integration Service process. No DTM messages appear in the mapping log until the entire mapping is complete. To resolve the issue, you can wait until the mapping completes before accessing the log. Or, you can find the log file that the DTM process temporarily writes on the worker compute node.

- The mapping has completed, but the DTM process failed to send the complete log file to the master Data Integration Service process.

The DTM process might fail to send the complete DTM log because of a network error or because the worker compute node unexpectedly shut down. The DTM process sends the log file to the Data Integration Service process in multiple sections. The DTM section of the log begins and ends with the following lines:

```
###
### <MyWorkerComputeNodeName>
###

### Start Grid Task [gtid-1443479776986-1-79777626-99] Segment [s0] Tasklet [t-0]
Attempt [1]

....

### End Grid Task [gtid-1443479776986-1-79777626-99] Segment [s0] Tasklet [t-0]
Attempt [1]
```

If these lines are not included in the mapping log or if the beginning line is included but not the ending line, then the DTM process failed to send the complete log file. To resolve the issue, you can find the DTM log files written to the following directory on the node where the master Data Integration Service process runs:

```
<Informatica installation directory>/logs/<node name>/services/DataIntegrationService/
disLogs/logConsolidation/<mappingName>_<jobID>_<timestamp>
```

If the job ID folder is empty, find the log file that the DTM process temporarily writes on the worker compute node.

To find the temporary DTM log file on the worker compute node, find the following message in the first section of the mapping log:

```
INFO: [GCL_5] The grid task [gtid-1443479776986-1-79777626-99] cluster logs can be found
at [./1443479776986/taskletlogs/gtid-1443479776986-1-79777626-99].
```

The listed directory is a subdirectory of the following default log directory configured for the worker compute node:

```
<Informatica installation directory>/logs/<node name>/dtmLogs/
```

CHAPTER 7

Data Integration Service Applications

This chapter includes the following topics:

- [Data Integration Service Applications Overview, 156](#)
- [Applications, 157](#)
- [Logical Data Objects, 161](#)
- [Physical Data Objects, 162](#)
- [Mappings, 162](#)
- [SQL Data Services, 164](#)
- [Web Services, 168](#)
- [Workflows, 171](#)

Data Integration Service Applications Overview

A developer can create a logical data object, physical data object, mapping, SQL data service, web service, or workflow and add it to an application in the Developer tool. To run the application, the developer must deploy it. A developer can deploy an application to an application archive file or deploy the application directly to the Data Integration Service.

As an administrator, you can deploy an application archive file to a Data Integration Service. You can enable the application to run and start the application.

When you deploy an application archive file to a Data Integration Service, the Deployment Manager validates the logical data objects, physical data objects, mappings, SQL data services, web services, and workflows in the application. The deployment fails if errors occur. The connections that are defined in the application must be valid in the domain that you deploy the application to.

The Data Integration Service stores the application in the Model repository associated with the Data Integration Service.

You can configure the default deployment mode for a Data Integration Service. The default deployment mode determines the state of each application after deployment. An application is disabled, stopped, or running after deployment.

Applications View

To manage deployed applications, select a Data Integration Service in the Navigator and then click the Applications view.

The Applications view displays the applications that have been deployed to a Data Integration Service. You can view the objects in the application and the properties. You can start and stop an application, an SQL data service, and a web service in the application. You can also back up and restore an application.

The Applications view shows the applications in alphabetic order. The Applications view does not show empty folders. Expand the application name in the top panel to view the objects in the application.

When you select an application or object in the top panel of the Applications view, the bottom panel displays read-only general properties and configurable properties for the selected object. The properties change based on the type of object you select.

When you select physical data objects, you can click a column heading in the lower panel to sort the list of objects. You can use the filter bar to filter the list of objects.

Refresh the Applications view to see the latest applications and their states.

Applications

The Applications view displays the applications that users deployed to a Data Integration Service. You can view the objects in the application and application properties. You can deploy, enable, rename, start, back up, and restore an application.

Application State

The Applications view shows the state for each application deployed to the Data Integration Service.

An application can have one of the following states:

- Running. The application is running.
- Stopped. The application is enabled to run but it is not running.
- Disabled. The application is disabled from running. If you recycle the Data Integration Service, the application will not start.
- Failed. The administrator started the application, but it failed to start.

Application Properties

Application properties include read-only general properties and a property to configure whether the application starts when the Data Integration Service starts.

The following table describes the read-only general properties for applications:

Property	Description
Name	Name of the application.
Description	Short description of the application.

Property	Description
Type	Type of the object. Valid value is application.
Location	The location of the application. This includes the domain and Data Integration Service name.
Last Modification Date	Date that the application was last modified.
Deployment Date	Date that the application was deployed.
Created By	User who created the application.
Unique Identifier	ID that identifies the application in the Model repository.
Creation Project Path	Path in the project that contains the application.
Creation Date	Date that the application was created.
Last Modified By	User who modified the application last.
Creation Domain	Domain in which the application was created.
Deployed By	User who deployed the application.

The following table describes the configurable application property:

Property	Description
Startup Type	Determines whether an application starts when the Data Integration Service starts. When you enable the application, the application starts by default when you start or recycle the Data Integration Service. Choose Disabled to prevent the application from starting. You cannot manually start an application if it is disabled.

Deploying an Application

Deploy an object to an application archive file if you want to check the application into version control or if your organization requires that administrators deploy objects to Data Integration Services.

1. Click the **Domain** tab.
2. Select a Data Integration Service, and then click the **Applications** view.
3. In Domain Actions, click **Deploy Full Application**.
The **Deploy Application** dialog box appears.
4. Click **Upload Files**.
The **Add Files** dialog box appears.
5. Click **Browse** to search for an application archive file.
Application archive files have a file extension of .iar.
6. Click **Add More Files** if you want to deploy multiple application files.
You can add up to 10 files.
7. Click **Deploy** to finish the selection.

The application file names appear in the **Uploaded Applications Archive Files** panel. The destination Data Integration Service appears as selected in the **Data Integration Services** panel.

8. To select additional Data Integration Services, select them in the **Data Integration Services** panel. To choose all Data Integration Services, select the box at the top of the list.

9. Click **OK** to start the deployment.

If no errors are reported, the deployment succeeds and the application starts.

10. To continue working while deployment is in progress, you can click **Deploy in Background**.

The deployment process continues in the background.

11. If a name conflict occurs, choose one of the following options to resolve the conflict:

- **Keep the existing application and discard the new application.**
- **Replace the existing application with the new application.**
- **Update the existing application with the new application.**
- **Rename the new application.** Enter the new application name if you select this option.

If you replace or update the existing application and the existing application is running, select the **Force Stop the Existing Application if it is Running** option to stop the existing application. You cannot update or replace an existing application that is running. When you stop an application, all running objects in the application are aborted.

After you select an option, click **OK**.

12. Click **Close**.

You can also deploy an application file using the `infacmd dis deployApplication` program.

Enabling an Application

An application must be enabled to run before you can start it. When you enable a Data Integration Service, the enabled applications start automatically.

You can configure a default deployment mode for a Data Integration Service. When you deploy an application to a Data Integration Service, the property determines the application state after deployment. An application might be enabled or disabled. If an application is disabled, you can enable it manually. If the application is enabled after deployment, the SQL data services, web services, and workflows are also enabled.

1. Select the Data Integration Service in the Navigator.
2. In the **Applications** view, select the application that you want to enable.
3. In **Application Properties** area, click **Edit**.

The **Edit Application Properties** dialog box appears.

4. In the **Startup Type** field, select **Enabled** and click **OK**.

The application is enabled to run.

You must enable each SQL data service or web service that you want to run.

Renaming an Application

Rename an application to change the name. You can rename an application when the application is not running.

1. Select the Data Integration Service in the Navigator.
2. In the **Application** view, select the application that you want to rename.

3. Click **Actions > Rename Application**.
4. Enter the name and click **OK**.

Starting an Application

You can start an application from the Administrator tool.

An application must be running before you can start or access an object in the application. You can start the application from the Applications Actions menu if the application is enabled to run.

1. Select the Data Integration Service in the Navigator.
2. In the **Applications** view, select the application that you want to start.
3. Click **Actions > Start Application**.

Backing Up an Application

You can back up an application to an XML file. The backup file contains all the property settings for the application. You can restore the application to another Data Integration Service.

You must stop the application before you back it up.

1. In the **Applications** view, select the application to back up.
2. Click **Actions > Backup Application**.
The Administrator tool prompts you to open the XML file or save the XML file.
3. Click **Open** to view the XML file in a browser.
4. Click **Save** to save the XML file.
5. If you click **Save**, enter an XML file name and choose the location to back up the application.
The Administrator tool backs up the application to an XML file in the location you choose.

Restoring an Application

You can restore an application from an XML backup file. The application must be an XML backup file that you create with the Backup option.

1. In the Domain Navigator, select a Data Integration Service that you want to restore the application to.
2. Click the **Applications** view.
3. Click **Actions > Restore Application from File**.
The Administrator tool prompts you for the file to restore.
4. Browse for and select the XML file.
5. Click **OK** to start the restore.
The Administrator tool checks for a duplicate application.
6. If a conflict occurs, choose one of the following options:
 - Keep the existing application and discard the new application. The Administrator tool does not restore the file.
 - Replace the existing application with the new application. The Administrator tool restores the backup application to the Data Integration Service.
 - Rename the new application. Choose a different name for the application you are restoring.

7. Click **OK** to restore the application.

The application starts if the default deployment option is set to Enable and Start for the Data Integration Service.

Refreshing the Applications View

Refresh the Applications view to view newly deployed and restored applications, remove applications that were recently undeployed, and update the state of each application.

1. Select the Data Integration Service in the Navigator.
2. Click the **Applications** view.
3. Select the application in the **Content** panel.
4. Click **Refresh Application View** in the application Actions menu.
The **Application** view refreshes.

Logical Data Objects

The Applications view displays logical data objects included in applications that have been deployed to the Data Integration Service.

Logical data object properties include read-only general properties and properties to configure caching for logical data objects.

The following table describes the read-only general properties for logical data objects:

Property	Description
Name	Name of the logical data object.
Description	Short description of the logical data object.
Type	Type of the object. Valid value is logical data object.
Location	The location of the logical data object. This includes the domain and Data Integration Service name.

The following table describes the configurable logical data object properties:

Property	Description
Enable Caching	Cache the logical data object in the data object cache database.
Cache Refresh Period	Number of minutes between cache refreshes.
Cache Table Name	<p>The name of the user-managed table from which the Data Integration Service accesses the logical data object cache. A user-managed cache table is a table in the data object cache database that you create, populate, and manually refresh when needed.</p> <p>If you specify a cache table name, the Data Object Cache Manager does not manage the cache for the object and ignores the cache refresh period.</p> <p>If you do not specify a cache table name, the Data Object Cache Manager manages the cache for the object.</p>

The following table describes the configurable logical data object column properties:

Property	Description
Create Index	Enables the Data Integration Service to generate indexes for the cache table based on this column. Default is false.

Physical Data Objects

The Applications view displays physical data objects included in applications that are deployed on the Data Integration Service.

The following table describes the read-only general properties for physical data objects:

Property	Description
Name	Name of the physical data object.
Type	Type of the object.

Mappings

The Applications view displays mappings included in applications that have been deployed to the Data Integration Service.

Mapping properties include read-only general properties and properties to configure the settings the Data Integration Services uses when it runs the mappings in the application.

The following table describes the read-only general properties for mappings:

Property	Description
Name	Name of the mapping.
Description	Short description of the mapping.
Type	Type of the object. Valid value is mapping.
Location	The location of the mapping. This includes the domain and Data Integration Service name.

The following table describes the configurable mapping properties:

Property	Description
Date format	Date/time format the Data Integration Services uses when the mapping converts strings to dates. Default is MM/DD/YYYY HH24:MI:SS.
Enable high precision	Runs the mapping with high precision. High precision data values have greater accuracy. Enable high precision if the mapping produces large numeric values, for example, values with precision of more than 15 digits, and you require accurate values. Enabling high precision prevents precision loss in large numeric values. Default is enabled.
Tracing level	Overrides the tracing level for each transformation in the mapping. The tracing level determines the amount of information the Data Integration Service sends to the mapping log files. Choose one of the following tracing levels: <ul style="list-style-type: none"> - None. The Data Integration Service uses the tracing levels set in the mapping. - Terse. The Data Integration Service logs initialization information, error messages, and notification of rejected data. - Normal. The Data Integration Service logs initialization and status information, errors encountered, and skipped rows due to transformation row errors. It summarizes mapping results, but not at the level of individual rows. - Verbose Initialization. In addition to normal tracing, the Data Integration Service logs additional initialization details, names of index and data files used, and detailed transformation statistics. - Verbose Data. In addition to verbose initialization tracing, the Data Integration Service logs each row that passes into the mapping. The Data Integration Service also notes where it truncates string data to fit the precision of a column and provides detailed transformation statistics. The Data Integration Service writes row data for all rows in a block when it processes a transformation. Default is None.

Property	Description
Optimization level	<p>Controls the optimization methods that the Data Integration Service applies to a mapping as follows:</p> <ul style="list-style-type: none"> - None. The Data Integration Service does not optimize the mapping. - Minimal. The Data Integration Service applies the early projection optimization method to the mapping. - Normal. The Data Integration Service applies the early projection, early selection, and predicate optimization methods to the mapping. - Full. The Data Integration Service applies the early projection, early selection, predicate optimization, and semi-join optimization methods to the mapping. <p>Default is Normal.</p>
Sort order	<p>Order in which the Data Integration Service sorts character data in the mapping.</p> <p>Default is Binary.</p>

SQL Data Services

The Applications view displays SQL data services included in applications that have been deployed to a Data Integration Service. You can view objects in the SQL data service and configure properties that the Data Integration Service uses to run the SQL data service. You can enable and rename an SQL data service.

SQL Data Service Properties

SQL data service properties include read-only general properties and properties to configure the settings the Data Integration Service uses when it runs the SQL data service.

When you expand an SQL data service in the top panel of the Applications view, you can access the following objects contained in an SQL data service:

- Virtual tables
- Virtual columns
- Virtual stored procedures

The Applications view displays read-only general properties for SQL data services and the objects contained in the SQL data services. Properties that appear in the view depend on the object type.

The following table describes the read-only general properties for SQL data services, virtual tables, virtual columns, and virtual stored procedures:

Property	Description
Name	Name of the selected object. Appears for all object types.
Description	Short description of the selected object. Appears for all object types.
Type	Type of the selected object. Appears for all object types.
Location	The location of the selected object. This includes the domain and Data Integration Service name. Appears for all object types.

Property	Description
JDBC URL	JDBC connection string used to access the SQL data service. The SQL data service contains virtual tables that you can query. It also contains virtual stored procedures that you can run. Appears for SQL data services.
Column Type	Datatype of the virtual column. Appears for virtual columns.

The following table describes the configurable SQL data service properties:

Property	Description
Startup Type	Determines whether the SQL data service is enabled to run when the application starts or when you start the SQL data service. Enter ENABLED to allow the SQL data service to run. Enter DISABLED to prevent the SQL data service from running.
Trace Level	Level of error written to the log files. Choose one of the following message levels: <ul style="list-style-type: none"> - OFF - SEVERE - WARNING - INFO - FINE - FINEST - ALL Default is INFO.
Connection Timeout	Maximum number of milliseconds to wait for a connection to the SQL data service. Default is 3,600,000.
Request Timeout	Maximum number of milliseconds for an SQL request to wait for an SQL data service response. Default is 3,600,000.
Sort Order	Sort order that the Data Integration Service uses for sorting and comparing data when running in Unicode mode. You can choose the sort order based on your code page. When the Data Integration runs in ASCII mode, it ignores the sort order value and uses a binary sort order. Default is binary.
Maximum Active Connections	Maximum number of active connections to the SQL data service.
Result Set Cache Expiration Period	The number of milliseconds that the result set cache is available for use. If set to -1, the cache never expires. If set to 0, result set caching is disabled. Changes to the expiration period do not apply to existing caches. If you want all caches to use the same expiration period, purge the result set cache after you change the expiration period. Default is 0.

Property	Description
DTM Keep Alive Time	<p>Number of milliseconds that the DTM instance stays open after it completes the last request. Identical SQL queries can reuse the open instance. Use the keep alive time to increase performance when the time required to process the SQL query is small compared to the initialization time for the DTM instance. If the query fails, the DTM instance terminates.</p> <p>Must be an integer. A negative integer value means that the DTM Keep Alive Time for the Data Integration Service is used. 0 means that the Data Integration Service does not keep the DTM instance in memory. Default is -1.</p>
Optimization Level	<p>The optimizer level that the Data Integration Service applies to the object. Enter the numeric value that is associated with the optimizer level that you want to configure. You can enter one of the following numeric values:</p> <ul style="list-style-type: none"> - 0. The Data Integration Service does not apply optimization. - 1. The Data Integration Service applies the early projection optimization method. - 2. The Data Integration Service applies the early projection, early selection, push-into, and predicate optimization methods. - 3. The Data Integration Service applies the cost-based, early projection, early selection, push-into, predicate, and semi-join optimization methods.

Virtual Table Properties

Configure whether to cache virtual tables for an SQL data service and configure how often to refresh the cache. You must disable the SQL data service before configuring virtual table properties.

The following table describes the configurable virtual table properties:

Property	Description
Enable Caching	Cache the virtual table in the data object cache database.
Cache Refresh Period	Number of minutes between cache refreshes.
Cache Table Name	<p>The name of the user-managed table from which the Data Integration Service accesses the virtual table cache. A user-managed cache table is a table in the data object cache database that you create, populate, and manually refresh when needed.</p> <p>If you specify a cache table name, the Data Object Cache Manager does not manage the cache for the object and ignores the cache refresh period.</p> <p>If you do not specify a cache table name, the Data Object Cache Manager manages the cache for the object.</p>

Virtual Column Properties

Configure the properties for the virtual columns included in an SQL data service.

The following table describes the configurable virtual column properties:

Property	Description
Create Index	Enables the Data Integration Service to generate indexes for the cache table based on this column. Default is false.
Deny With	<p>When you use column level security, this property determines whether to substitute the restricted column value or to fail the query. If you substitute the column value, you can choose to substitute the value with NULL or with a constant value.</p> <p>Select one of the following options:</p> <ul style="list-style-type: none">- ERROR. Fails the query and returns an error when an SQL query selects a restricted column.- NULL. Returns a null value for a restricted column in each row.- VALUE. Returns a constant value for a restricted column in each row.
Insufficient Permission Value	The constant that the Data Integration Service returns for a restricted column.

Virtual Stored Procedure Properties

Configure the property for the virtual stored procedures included in an SQL data service.

The following table describes the configurable virtual stored procedure property:

Property	Description
Result Set Cache Expiration Period	The number of milliseconds that the result set cache is available for use. If set to -1, the cache never expires. If set to 0, result set caching is disabled. Changes to the expiration period do not apply to existing caches. If you want all caches to use the same expiration period, purge the result set cache after you change the expiration period. Default is 0.

Enabling an SQL Data Service

Before you can start an SQL data service, the Data Integration Service must be running and the SQL data service must be enabled.

When a deployed application is enabled by default, the SQL data services in the application are also enabled.

When a deployed application is disabled by default, the SQL data services are also disabled. When you enable the application manually, you must also enable each SQL data service in the application.

1. Select the Data Integration Service in the Navigator.
2. In the **Applications** view, select the SQL data service that you want to enable.
3. In **SQL Data Service Properties** area, click **Edit**.
The **Edit Properties** dialog box appears.
4. In the **Startup Type** field, select **Enabled** and click **OK**.

Renaming an SQL Data Service

Rename an SQL data service to change the name of the SQL data service. You can rename an SQL data service when the SQL data service is not running.

1. Select the Data Integration Service in the Navigator.
2. In the **Application** view, select the SQL data service that you want to rename.
3. Click **Actions > Rename SQL Data Service**.
4. Enter the name and click **OK**.

Web Services

The Applications view displays web services included in applications that have been deployed to a Data Integration Service. You can view the operations in the web service and configure properties that the Data Integration Service uses to run a web service. You can enable and rename a web service.

Web Service Properties

REST web service and SOAP web service properties include read-only general properties and properties that the Data Integration Service uses when it runs a web service.

When you expand a web service or a REST web service in the top panel of the Applications view, you can access web service operations or resource in the web service.

The Applications view displays read-only general properties for the web services, web service operations, or web services resources. Properties that appear in the view depend on the object type.

The following table describes the read-only general properties for each type of web service and the web service operations or resources:

Property	Description
Name	Name of the selected object. Appears for all objects.
Description	Short description of the selected object. Appears for all objects.
Type	Type of the selected object. Appears for all object types.
Location	The location of the selected object. This includes the domain and Data Integration Service name. Appears for all objects.
URL	URL used to connect to the web service. Appears for web services.

The following table describes the configurable web service properties for web services:

Property	Description
Startup Type	Determines whether the web service is enabled to run when the application starts or when you start the web service.
Trace Level	<p>Level of error messages written to the run-time web service log. Choose one of the following message levels:</p> <ul style="list-style-type: none"> - OFF. The DTM process does not write messages to the web service run-time logs. - SEVERE. SEVERE messages include errors that might cause the web service to stop running. - WARNING. WARNING messages include recoverable failures or warnings. The DTM process writes WARNING and SEVERE messages to the web service run-time log. - INFO. INFO messages include web service status messages. The DTM process writes INFO, WARNING and SEVERE messages to the web service run-time log. - FINE. FINE messages include data processing errors for the web service request. The DTM process writes FINE, INFO, WARNING and SEVERE messages to the web service run-time log. - FINEST. FINEST message are used for debugging. The DTM process writes FINEST, FINE, INFO, WARNING and SEVERE messages to the web service run-time log. - ALL. The DTM process writes FINEST, FINE, INFO, WARNING and SEVERE messages to the web service run-time log. <p>Default is INFO.</p>
Request Timeout	Maximum number of milliseconds that the Data Integration Service runs an operation mapping before the web service request times out. Default is 3,600,000.
Maximum Concurrent Requests	Maximum number of requests that a web service can process at one time. Default is 10.
Sort Order	Sort order that the Data Integration Service to sort and compare data when running in Unicode mode.
Enable Transport Layer Security	Indicates that the web service must use HTTPS. If the Data Integration Service is not configured to use HTTPS, the web service will not start.

The following table contains properties unique to REST web services:

Property	Description
Is Authentication Required	Enables basic authentication for the REST web service. Basic authentication requires a user name and a password from web service requests. Default is disabled.
Input Precision	Maximum number of characters that the Data Integration Service parses in the request message. The web service request fails when the request message exceeds the input precision. Default is 10,000.
Output Precision	Maximum number of characters that the Data Integration Service generates for the response message. The Data Integration Service truncates the response message when the response message exceeds the output precision. Default is 3,000.

The following table contains properties unique to SOAP web services:

Property	Description
Enable WS-Security	Enables the Data Integration Service to validate the user credentials and verify that the user has permission to run each web service operation. SOAP web services only.
Optimization Level	The optimizer level that the Data Integration Service applies to the object. Enter the numeric value that is associated with the optimizer level that you want to configure. You can enter one of the following numeric values: <ul style="list-style-type: none">- 0. The Data Integration Service does not apply optimization.- 1. The Data Integration Service applies the early projection optimization method.- 2. The Data Integration Service applies the early projection, early selection, push-into, and predicate optimization methods.- 3. The Data Integration Service applies the cost-based, early projection, early selection, push-into, predicate, and semi-join optimization methods.
DTM Keep Alive Time	Number of milliseconds that the DTM instance stays open after it completes the last request. Web service requests that are issued against the same operation can reuse the open instance. Use the keep alive time to increase performance when the time required to process the request is small compared to the initialization time for the DTM instance. If the request fails, the DTM instance terminates. Must be an integer. A negative integer value means that the DTM Keep Alive Time for the Data Integration Service is used. 0 means that the Data Integration Service does not keep the DTM instance in memory. Default is -1.
SOAP Output Precision	Maximum number of characters that the Data Integration Service generates for the response message. The Data Integration Service truncates the response message when the response message exceeds the SOAP output precision. Default is 200,000.
SOAP Input Precision	Maximum number of characters that the Data Integration Service parses in the request message. The web service request fails when the request message exceeds the SOAP input precision. Default is 200,000.

Web Service Operation and Resource Properties

Configure the settings that the Data Integration Service uses when it runs a web service operation or a web service resource.

The following tables describes the configurable property for a SOAP web service operation or a REST web service resource:

Property	Description
Result Set Cache Expiration Period	The number of milliseconds that the result set cache is available for use. If set to -1, the cache never expires. If set to 0, result set caching is disabled. Changes to the expiration period do not apply to existing caches. If you want all caches to use the same expiration period, purge the result set cache after you change the expiration period. Default is 0.

Enabling a Web Service

Enable a web service so that you can start the web service. Before you can start a web service, the Data Integration Service must be running and the web service must be enabled.

1. Select the Data Integration Service in the Navigator.

2. In the **Application** view, select the web service that you want to enable.
3. In **Web Service Properties** section of the **Properties** view, click **Edit**.
The **Edit Properties** dialog box appears.
4. In the **Startup Type** field, select **Enabled** and click **OK**.

Renaming a Web Service

Rename a web service to change the service name of a web service. You can rename a web service when the web service is stopped.

1. Select the Data Integration Service in the Navigator.
2. In the **Application** view, select the web service that you want to rename.
3. Click **Actions > Rename Web Service**.
The **Rename Web Service** dialog box appears.
4. Enter the web service name and click **OK**.

Workflows

The Applications view displays workflows included in applications that have been deployed to a Data Integration Service. You can view workflow properties, enable a workflow, and start a workflow.

Workflow Properties

Workflow properties include read-only general properties.

The following table describes the read-only general properties for workflows:

Property	Description
Name	Name of the workflow.
Description	Short description of the workflow.
Type	Type of the object. Valid value is workflow.
Location	The location of the workflow. This includes the domain and Data Integration Service name.

Enabling a Workflow

Before you can run instances of the workflow, the Data Integration Service must be running and the workflow must be enabled.

Enable a workflow to allow users to run instances of the workflow. Disable a workflow to prevent users from running instances of the workflow. When you disable a workflow, the Data Integration Service aborts any running instances of the workflow.

When a deployed application is enabled by default, the workflows in the application are also enabled.

When a deployed application is disabled by default, the workflows are also disabled. When you enable the application manually, each workflow in the application is also enabled.

1. Select the Data Integration Service in the Navigator.
2. In the **Applications** view, select the workflow that you want to enable.
3. Click **Actions > Enable Workflow**.

Starting a Workflow

After you deploy a workflow, you run an instance of the workflow from the deployed application from the Administrator tool.

1. In the Administrator tool, click the Data Integration Service on which you deployed the workflow.
2. Click the **Applications** tab.
3. Expand the application that contains the workflow you want to start.
4. Select the workflow that you want to run.
5. Click **Actions > Start Workflow**.

The **Start Workflow** dialog box appears.

6. Optionally, browse and select a parameter file for the workflow run.
7. Select Show Workflow Monitoring if you want to view the workflow graph for the workflow run.
8. Click **OK**.

CHAPTER 8

Metadata Manager Service

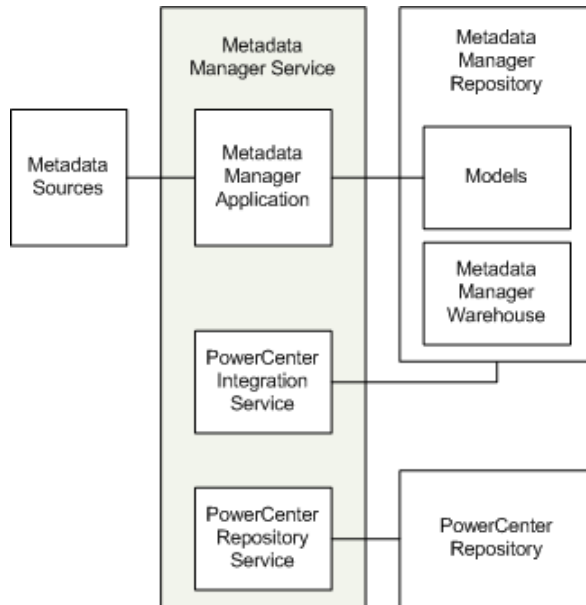
This chapter includes the following topics:

- [Metadata Manager Service Overview, 173](#)
- [Configuring a Metadata Manager Service, 174](#)
- [Creating a Metadata Manager Service, 175](#)
- [Creating and Deleting Repository Content, 180](#)
- [Enabling and Disabling the Metadata Manager Service, 181](#)
- [Metadata Manager Service Properties, 182](#)
- [Configuring the Associated PowerCenter Integration Service, 189](#)

Metadata Manager Service Overview

The Metadata Manager Service is an application service that runs the Metadata Manager application in an Informatica domain. The Metadata Manager application manages access to metadata in the Metadata Manager repository. Create a Metadata Manager Service in the domain to access the Metadata Manager application.

The following figure shows the Metadata Manager components managed by the Metadata Manager Service on a node in an Informatica domain:



The Metadata Manager Service manages the following components:

- Metadata Manager application. The Metadata Manager application is a web-based application. Use Metadata Manager to browse and analyze metadata from disparate source repositories. You can load, browse, and analyze metadata from application, business intelligence, data integration, data modeling, and relational metadata sources.
- PowerCenter repository for Metadata Manager. Contains the metadata objects used by the PowerCenter Integration Service to load metadata into the Metadata Manager warehouse. The metadata objects include sources, targets, sessions, and workflows.
- PowerCenter Repository Service. Manages connections to the PowerCenter repository for Metadata Manager.
- PowerCenter Integration Service. Runs the workflows in the PowerCenter repository to read from metadata sources and load metadata into the Metadata Manager warehouse.
- Metadata Manager repository. Contains the Metadata Manager warehouse and models. The Metadata Manager warehouse is a centralized metadata warehouse that stores the metadata from metadata sources. Models define the metadata that Metadata Manager extracts from metadata sources.
- Metadata sources. The application, business intelligence, data integration, data modeling, and database management sources that Metadata Manager extracts metadata from.

Configuring a Metadata Manager Service

You can create and configure a Metadata Manager Service and the related components in the Administrator tool.

Note: The procedure to configure the Metadata Manager Service varies based on the operating mode of the PowerCenter Repository Service and on whether the PowerCenter repository contents are created or not.

1. Set up the Metadata Manager repository database. Set up a database for the Metadata Manager repository. You supply the database information when you create the Metadata Manager Service.
2. Create a PowerCenter Repository Service and PowerCenter Integration Service (Optional). You can use an existing PowerCenter Repository Service and PowerCenter Integration Service, or you can create them. If want to create the application services to use with Metadata Manager, create the services in the following order:
 - a. PowerCenter Repository Service. Create a PowerCenter Repository Service but do not create contents. Start the PowerCenter Repository Service in exclusive mode.
 - b. PowerCenter Integration Service. Create the PowerCenter Integration Service. The service will not start because the PowerCenter Repository Service does not have content. You enable the PowerCenter Integration Service after you create and configure the Metadata Manager Service.
3. Create the Metadata Manager Service. Use the Administrator tool to create the Metadata Manager Service.
4. Configure the Metadata Manager Service. Configure the properties for the Metadata Manager Service.
5. Create repository contents. The steps to create repository contents differ based on the code page of the Metadata Manager and PowerCenter repositories.

If the code page is Latin-based, then create contents for the Metadata Manager repository and restore the PowerCenter repository. Use the Metadata Manager Service **Actions** menu to create the contents for both repositories.

If the code page is not Latin-based, then create the repository contents in the following order:

- a. Restore the PowerCenter repository. Use the Metadata Manager Service **Actions** menu to restore the PowerCenter repository. When you restore the PowerCenter repository, enable the option to automatically restart the PowerCenter Repository Service in normal mode.
 - b. Create the Metadata Manager repository contents. Use the Metadata Manager Service **Actions** menu to create the contents.
6. Enable the PowerCenter Integration Service. Enable the associated PowerCenter Integration Service for the Metadata Manager Service.
7. Enable the Metadata Manager Service. Enable the Metadata Manager Service in the Informatica domain.
8. Create or assign users. Create users and assign them privileges for the Metadata Manager Service, or assign existing users privileges for the Metadata Manager Service.

Note: You can use a Metadata Manager Service and the associated Metadata Manager repository in one Informatica domain. After you create the Metadata Manager Service and Metadata Manager repository in one domain, you cannot create a second Metadata Manager Service to use the same Metadata Manager repository. You also cannot back up and restore the repository to use with a different Metadata Manager Service in a different domain.

Creating a Metadata Manager Service

Use the Administrator tool to create the Metadata Manager Service. After you create the Metadata Manager Service, create the Metadata Manager repository contents and PowerCenter repository contents to enable the service.

1. In the Administrator tool, click the **Manage** tab.

2. Click the **Services and Nodes** view.
3. Click **Actions > New Metadata Manager Service**.
The **New Metadata Manager Service** dialog box appears.
4. Enter values for the Metadata Manager Service general properties, and click **Next**.
5. Enter values for the Metadata Manager Service database properties, and click **Next**.
6. Enter values for the Metadata Manager Service security properties, and click **Finish**.

Metadata Manager Service Properties

The following table describes the properties that you configure for the Metadata Manager Service:

Property	Description
Name	Name of the Metadata Manager Service. The name is not case sensitive and must be unique within the domain. It cannot exceed 128 characters or begin with @. It also cannot contain spaces or the following special characters: ` ~ % ^ * + = { } \ ; : ' " / ? . , < > ! ()] [
Description	The description cannot exceed 765 characters.
Location	Domain and folder where the service is created. Click Browse to choose a different folder. You can move the Metadata Manager Service after you create it.
License	License object that allows use of the service.
Node	Node in the Informatica domain that the Metadata Manager Service runs on.
Associated Integration Service	PowerCenter Integration Service used by Metadata Manager to load metadata into the Metadata Manager warehouse.
Repository User Name	User account for the PowerCenter repository. Use the repository user account you configured for the PowerCenter Repository Service. For a list of the required privileges for this user, see "Privileges for the Associated PowerCenter Integration Service User" on page 190 .
Repository Password	Password for the PowerCenter repository user.
Security Domain	Name of the security domain to which the PowerCenter repository user belongs.
Database Type	Type of database for the Metadata Manager repository.
Code Page	Metadata Manager repository code page. The Metadata Manager Service and Metadata Manager application use the character set encoded in the repository code page when writing data to the Metadata Manager repository. Note: The Metadata Manager repository code page, the code page on the machine where the associated PowerCenter Integration Service runs, and the code page for any database management and PowerCenter resources that you load into the Metadata Manager warehouse must be the same.
Connect String	Native connect string to the Metadata Manager repository database. The Metadata Manager Service uses the connect string to create a connection object to the Metadata Manager repository in the PowerCenter repository.

Property	Description
Database User	User account for the Metadata Manager repository database. Set up this account with the appropriate database client tools.
Database Password	Password for the Metadata Manager repository database user. Must be in 7-bit ASCII.
Tablespace Name	<p>Tablespace name for Metadata Manager repositories on IBM DB2. When you specify the tablespace name, the Metadata Manager Service creates all repository tables in the same tablespace. You cannot use spaces in the tablespace name.</p> <p>To improve repository performance on IBM DB2 EEE repositories, specify a tablespace name with one node.</p>
Database Hostname	Host name for the Metadata Manager repository database.
Database Port	Port number for the Metadata Manager repository database.
SID/Service Name	Indicates whether the Database Name property contains an Oracle full service name or SID.
Database Name	Full service name or SID for Oracle databases. Service name for IBM DB2 databases. Database name for Microsoft SQL Server databases.
Additional JDBC Parameters	<p>Additional JDBC parameters that you want to append to the database connection URL. Enter the parameters as name=value pairs separated by semicolon characters (;). For example:</p> <pre>param1=value1;param2=value2</pre> <p>You can use this property to specify the following information:</p> <ul style="list-style-type: none"> - Backup server location. If you use a database server that is highly available such as Oracle RAC, enter the location of a backup server. - Oracle Advanced Security Option (ASO) parameters. If the Metadata Manager repository database is an Oracle database that uses ASO, enter the following additional parameters: EncryptionLevel=[encryption level];EncryptionTypes=[encryption types];DataIntegrityLevel=[data integrity level];DataIntegrityTypes=[data integrity types] The parameter values must match the values in the sqlnet.ora file on the machine where the Metadata Manager Service runs. - Authentication information for Microsoft SQL Server. <p>Note: The Metadata Manager Service does not support the alternatID option for DB2.</p> <p>To authenticate the user credentials with Windows authentication and establish a trusted connection to a Microsoft SQL Server repository, enter the following text: AuthenticationMethod=ntlm;LoadLibraryPath=[directory containing DDJDBCx64Auth04.dll].</p> <pre>jdbc:informatica:sqlserver://[host]:[port];DatabaseName=[DB name];AuthenticationMethod=ntlm;LoadLibraryPath=[directory containing DDJDBCx64Auth04.dll]</pre> <p>When you use a trusted connection to connect to a Microsoft SQL Server database, the Metadata Manager Service connects to the repository with the credentials of the user logged in to the machine on which the service is running.</p> <p>To start the Metadata Manager Service as a Windows service with a trusted connection, configure the Windows service properties to log on with a trusted user account.</p>

Property	Description
Secure JDBC Parameters	Secure JDBC parameters that you want to append to the database connection URL. Use this property to specify secure connection parameters such as passwords. The Administrator tool does not display secure parameters or parameter values in the Metadata Manager Service properties. Enter the parameters as name=value pairs separated by semicolon characters (;). For example: param1=value1;param2=value2 If secure communication is enabled for the Metadata Manager repository database, enter the secure JDBC parameters in this property.
Port Number	Port number the Metadata Manager application runs on. Default is 10250.
Enable Secured Socket Layer	Indicates that you want to configure a secure connection for the Metadata Manager web application. If you enable this option, you must create a keystore file that contains the required keys and certificates. You can create a keystore file with keytool. keytool is a utility that generates and stores private or public key pairs and associated certificates in a keystore file. When you generate a public or private key pair, keytool wraps the public key into a self-signed certificate. You can use the self-signed certificate or use a certificate signed by a certificate authority.
Keystore File	Keystore file that contains the keys and certificates required if you configure a secure connection for the Metadata Manager web application. Required if you select Enable Secured Socket Layer.
Keystore Password	Password for the keystore file. Required if you select Enable Secured Socket Layer.

JDBC Parameters for Secure Databases

If secure communication is enabled for the Metadata Manager repository database, you must configure additional JDBC parameters in the **Secure JDBC Parameters** property.

Enter the following parameters in the **Secure JDBC Parameters** property:

```
EncryptionMethod=SSL;TrustStore=<truststore
location>;TrustStorePassword=<password>;HostNameInCertificate=<host
name>;ValidateServerCertificate=<true|false>;KeyStore=<keystore
location>;keyStorePassword=<password>
```

Configure the parameters as follows:

EncryptionMethod

Encryption method for data transfer between Metadata Manager and the database server. Must be set to SSL.

TrustStore

Path and file name of the truststore file that contains the security certificate of the database server.

TrustStorePassword

Password used to access the truststore file.

HostNameInCertificate

Host name of the machine that hosts the secure database. If you specify a host name, the Metadata Manager Service validates the host name included in the connection string against the host name in the security certificate.

ValidateServerCertificate

Indicates whether the Metadata Manager Service validates the certificate that the database server presents. If you set this parameter to true, the Metadata Manager Service validates the certificate. If you specify the HostNameInCertificate parameter, the Metadata Manager Service also validates the host name in the certificate.

If you set this parameter to false, the Metadata Manager Service does not validate the certificate that the database server presents. The Metadata Manager Service ignores any truststore information that you specify.

KeyStore

Path and file name of the keystore file that contains the security certificates that the Metadata Manager Service presents to the database server.

KeyStorePassword

Password used to access the keystore file.

Database Connect Strings

When you create a database connection, specify a connect string for that connection. The Metadata Manager Service uses the connect string to create a connection object to the Metadata Manager repository database in the PowerCenter repository.

The following table lists the native connect string syntax for each supported database:

Database	Connect String Syntax	Example
IBM DB2	<i>dbname</i>	mydatabase
Microsoft SQL Server	<i>servername@dbname</i>	sqlserver@mydatabase Note: If you do not specify the connect string in the syntax specified, you must specify the ODBC entry specified for the data source.
Oracle	<i>dbname.world</i> (same as TNSNAMES entry)	oracle.world

Note: The Metadata Manager Service uses the DataDirect drivers included with the Informatica installation. Informatica does not support the use of any other database driver.

Overriding the Repository Database Code Page

You can override the default database code page for the Metadata Manager repository database when you create or configure the Metadata Manager Service. Override the code page if the Metadata Manager repository contains characters that the database code page does not support.

To override the code page, add the CODEPAGEOVERRIDE parameter to the Additional JDBC Options property. Specify a code page that is compatible with the default repository database code page.

For example, use the following parameter to override the default Shift-JIS code page with MS932:

```
CODEPAGEOVERRIDE=MS932;
```

Creating and Deleting Repository Content

You can create and delete contents for the following repositories used by Metadata Manager:

- Metadata Manager repository. Create the Metadata Manager warehouse tables and import models for metadata sources into the Metadata Manager repository.
- PowerCenter repository. Restore a repository backup file packaged with PowerCenter to the PowerCenter repository database. The repository backup file includes the metadata objects used by Metadata Manager to load metadata into the Metadata Manager warehouse. When you restore the repository, the Service Manager creates a folder named Metadata Load in the PowerCenter repository. The Metadata Load folder contains the metadata objects, including sources, targets, sessions, and workflows.

The tasks you complete depend on whether the Metadata Manager repository contains contents or if the PowerCenter repository contains the PowerCenter objects for Metadata Manager.

The following table describes the tasks you must complete for each repository:

Repository	Condition	Action
Metadata Manager repository	Does not have content.	Create the Metadata Manager repository.
Metadata Manager repository	Has content.	No action.
PowerCenter repository	Does not have content.	Restore the PowerCenter repository if the PowerCenter Repository Service runs in exclusive mode.
PowerCenter repository	Has content.	No action if the PowerCenter repository has the objects required for Metadata Manager in the Metadata Load folder. The Service Manager imports the required objects from an XML file when you enable the service.

Creating the Metadata Manager Repository

When you create the Metadata Manager repository, you create the Metadata Manager warehouse tables and import models for metadata sources.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the Metadata Manager Service for which the Metadata Manager repository has no content.
3. Click **Actions** > **Repository Contents** > **Create**.
4. Optionally, choose to restore the PowerCenter repository. You can restore the repository if the PowerCenter Repository Service runs in exclusive mode and the repository does not contain contents.
5. Click **OK**.

The activity log displays the results of the create contents operation.

Restoring the PowerCenter Repository

Restore the repository backup file for the PowerCenter repository to create the objects used by Metadata Manager in the PowerCenter repository database.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.

2. In the Domain Navigator, select the Metadata Manager Service for which the PowerCenter repository has no contents.
3. Click **Actions > Restore PowerCenter Repository**.
4. Optionally, choose to restart the PowerCenter Repository Service in normal mode.
5. Click **OK**.

The activity log displays the results of the restore repository operation.

Deleting the Metadata Manager Repository

Delete Metadata Manager repository content when you want to delete all metadata and repository database tables from the repository. Delete the repository content if the metadata is obsolete. If the repository contains information that you want to save, back up the repository with the database client or mmRepoCmd before you delete it.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the Metadata Manager Service for which you want to delete Metadata Manager repository content.
3. Click **Actions > Repository Contents > Delete**.
4. Enter the user name and password for the database account.
5. Click **OK**.

The activity log displays the results of the delete contents operation.

Enabling and Disabling the Metadata Manager Service

Use the Administrator tool to enable, disable, or recycle the Metadata Manager Service. Disable a Metadata Manager Service to perform maintenance or to temporarily restrict users from accessing Metadata Manager. When you disable the Metadata Manager Service, you also stop Metadata Manager. You might recycle a service if you modified a property. When you recycle the service, the Metadata Manager Service is disabled and enabled.

When you enable the Metadata Manager Service, the Service Manager starts the Metadata Manager application on the node where the Metadata Manager Service runs. If the PowerCenter repository does not contain the Metadata Load folder, the Administrator tool imports the metadata objects required by Metadata Manager into the PowerCenter repository.

You can enable, disable, and recycle the Metadata Manager Service from the **Actions** menu.

Note: The PowerCenter Repository Service for Metadata Manager must be enabled and running before you can enable the Metadata Manager Service.

Metadata Manager Service Properties

You can configure general, Metadata Manager Service, database, configuration, connection pool, advanced, and custom properties for the Metadata Manager Service.

After you create a Metadata Manager Service, you can configure it. After you configure Metadata Manager Service properties, you must disable and enable the Metadata Manager Service for the changes to take effect.

Use the Administrator tool to configure the following Metadata Manager Service properties:

- General properties. Include the name and description of the service, the license object for the service, and the node where the service runs.
- Metadata Manager Service properties. Include port numbers for the Metadata Manager application and the Metadata Manager Agent, and the Metadata Manager file location.
- Database properties. Include database properties for the Metadata Manager repository.
- Configuration properties. Include the HTTP security protocol and keystore file, and maximum concurrent and queued requests to the Metadata Manager application.
- Connection pool properties. Metadata Manager maintains a connection pool for connections to the Metadata Manager repository. Connection pool properties include the number of active available connections to the Metadata Manager repository database and the amount of time that Metadata Manager holds database connection requests in the connection pool.
- Advanced properties. Include properties for the Java Virtual Manager (JVM) memory settings, and Metadata Manager Browse and Load tab options.
- Custom properties. Configure custom properties that are unique to specific environments.

If you update any of the properties, restart the Metadata Manager Service for the modifications to take effect.

General Properties

To edit the general properties, select the Metadata Manager Service in the Navigator, select the **Properties** view, and then click **Edit** in the General Properties section.

The following table describes the general properties for the service:

Property	Description
Name	Name of the service. The name is not case sensitive and must be unique within the domain. It cannot exceed 128 characters or begin with @. It also cannot contain spaces or the following special characters: ` ~ % ^ * + = { } \ ; : ' " / ? . , < > ! () [] You cannot change the name of the service after you create it.
Description	Description of the service. The description cannot exceed 765 characters.
License	License object that allows use of the service.
Node	Node on which the service runs. To assign the Metadata Manager Service to a different node, you must first disable the service.

Assigning the Metadata Manager Service to a Different Node

1. Disable the Metadata Manager Service.

2. Click Edit in the General Properties section.
3. Select another node for the Node property, and then click OK.
4. Click Edit in the Metadata Manager Service Properties section.
5. Change the Metadata Manager File Location property to a location that is accessible from the new node, and then click OK.
6. Copy the contents of the Metadata Manager file location directory on the original node to the location on the new node.
7. If the Metadata Manager Service is running in HTTPS security mode, click Edit in the Configuration Properties section. Change the Keystore File location to a location that is accessible from the new node, and then click OK.
8. Enable the Metadata Manager Service.

Metadata Manager Service Properties

To edit the Metadata Manager Service properties, select the Metadata Manager Service in the Navigator, select the **Properties** view, and then click **Edit** in the Metadata Manager Service Properties section.

The following table describes the Metadata Manager Service properties:

Property	Description
Port Number	Port number that the Metadata Manager application runs on. Default is 10250.
Agent Port	<p>Port number for the Metadata Manager Agent when the Metadata Manager Service runs on Windows. The agent uses this port to communicate with metadata source repositories. Default is 10251.</p> <p>If the Metadata Manager Service runs on UNIX, you must install the Metadata Manager Agent on a separate Windows machine.</p>
Metadata Manager File Location	<p>Location of the files used by the Metadata Manager application. Files include the following file types:</p> <ul style="list-style-type: none"> - Index files. Index files created by Metadata Manager required to search the Metadata Manager warehouse. - Log files. Log files generated by Metadata Manager when you load resources. - Parameter files. Files generated by Metadata Manager and used by PowerCenter workflows. - Repository backup files. Metadata Manager repository backup files that are generated by the mmRepoCmd command line program. <p>By default, Metadata Manager stores the files in the following directory:</p> <pre><Informatica services installation directory>\services \MetadataManagerService\mm_files\<Metadata Manager Service name></pre>
Metadata Manager Lineage Graph Location	<p>Location that Metadata Manager uses to store graph database files for data lineage.</p> <p>By default, Metadata Manager stores the graph database files in the following directory:</p> <pre><Informatica services installation directory>\services \MetadataManagerService\mm_files\<Metadata Manager Service name></pre>

Metadata Manager File Location Rules and Guidelines

Use the following rules and guidelines when you configure the Metadata Manager file location:

- If you change the Metadata Manager file location, copy the contents of the directory to the new location.

- If you configure a shared file location, the location must be accessible to all nodes running a Metadata Manager Service and to all users of the Metadata Manager application.
- To decrease the load times for Cloudera Navigator resources, ensure that the Metadata Manager file location directory is on a disk with a fast input/output rate.

Metadata Manager Lineage Graph Location Rules and Guidelines

Use the following rules and guidelines when you configure the Metadata Manager lineage graph location:

- To change the Metadata Manager lineage graph location, you must disable the Metadata Manager Service, copy the contents of the directory to the new location, and then restart the Metadata Manager Service.
- The lineage graph location must be accessible to all nodes that run the Metadata Manager Service and to the Informatica domain administrator user account.

Database Properties

You can edit the Metadata Manager repository database properties. Select the Metadata Manager Service in the Navigator, select the **Properties** view, and then click **Edit** in the **Database Properties** section.

The following table describes the database properties for a Metadata Manager repository database:

Property	Description
Database Type	Type of database for the Metadata Manager repository. To apply changes, restart the Metadata Manager Service.
Code Page	Metadata Manager repository code page. The Metadata Manager Service and Metadata Manager use the character set encoded in the repository code page when writing data to the Metadata Manager repository. To apply changes, restart the Metadata Manager Service. Note: The Metadata Manager repository code page, the code page on the machine where the associated PowerCenter Integration Service runs, and the code page for any database management and PowerCenter resources that you load into the Metadata Manager warehouse must be the same.
Connect String	Native connect string to the Metadata Manager repository database. The Metadata Manager Service uses the connection string to create a target connection to the Metadata Manager repository in the PowerCenter repository. To apply changes, restart the Metadata Manager Service.
Database User	User account for the Metadata Manager repository database. Set up this account using the appropriate database client tools. To apply changes, restart the Metadata Manager Service.
Database Password	Password for the Metadata Manager repository database user. Must be in 7-bit ASCII. To apply changes, restart the Metadata Manager Service.
Tablespace Name	Tablespace name for the Metadata Manager repository on IBM DB2. When you specify the tablespace name, the Metadata Manager Service creates all repository tables in the same tablespace. You cannot use spaces in the tablespace name. To apply changes, restart the Metadata Manager Service. To improve repository performance on IBM DB2 EEE repositories, specify a tablespace name with one node.
Database Hostname	Host name for the Metadata Manager repository database. To apply changes, restart the Metadata Manager Service.

Property	Description
Database Port	Port number for the Metadata Manager repository database. To apply changes, restart the Metadata Manager Service.
SID/Service Name	Indicates whether the Database Name property contains an Oracle full service name or an SID.
Database Name	Full service name or SID for Oracle databases. Service name for IBM DB2 databases. Database name for Microsoft SQL Server databases. To apply changes, restart the Metadata Manager Service.
Additional JDBC Parameters	<p>Additional JDBC parameters that you want to append to the database connection URL. Enter the parameters as name=value pairs separated by semicolon characters (;). For example:</p> <pre>param1=value1;param2=value2</pre> <p>You can use this property to specify the following information:</p> <ul style="list-style-type: none"> - Backup server location. If you use a database server that is highly available such as Oracle RAC, enter the location of a backup server. - Oracle Advanced Security Option (ASO) parameters. If the Metadata Manager repository database is an Oracle database that uses ASO, enter the following additional parameters: EncryptionLevel=[encryption level];EncryptionTypes=[encryption types];DataIntegrityLevel=[data integrity level];DataIntegrityTypes=[data integrity types] The parameter values must match the values in the sqlnet.ora file on the machine where the Metadata Manager Service runs. - Authentication information for Microsoft SQL Server. Note: The Metadata Manager Service does not support the alternateld option for DB2. To authenticate the user credentials using Windows authentication and establish a trusted connection to a Microsoft SQL Server repository, enter the following text: AuthenticationMethod=ntlm;LoadLibraryPath=[directory containing DDJDBCx64Auth04.dll]. jdbc:informatica:sqlserver://[host]:[port];DatabaseName=[DB name];AuthenticationMethod=ntlm;LoadLibraryPath=[directory containing DDJDBCx64Auth04.dll] When you use a trusted connection to connect to a Microsoft SQL Server database, the Metadata Manager Service connects to the repository with the credentials of the user logged in to the machine on which the service is running. To start the Metadata Manager Service as a Windows service using a trusted connection, configure the Windows service properties to log on using a trusted user account.
Secure JDBC Parameters	<p>Secure JDBC parameters that you want to append to the database connection URL. Use this property to specify secure connection parameters such as passwords. The Administrator tool does not display secure parameters or parameter values in the Metadata Manager Service properties. Enter the parameters as name=value pairs separated by semicolon characters (;). For example:</p> <pre>param1=value1;param2=value2</pre> <p>If secure communication is enabled for the Metadata Manager repository database, enter the secure JDBC parameters in this property.</p> <p>To update the secure JDBC parameters, click Modify Secure JDBC Parameters and enter the new values.</p>

JDBC Parameters for Secure Databases

If secure communication is enabled for the Metadata Manager repository database, you must configure additional JDBC parameters in the **Secure JDBC Parameters** property.

Enter the following parameters in the **Secure JDBC Parameters** property:

```
EncryptionMethod=SSL;TrustStore=<truststore  
location>;TrustStorePassword=<password>;HostNameInCertificate=<host  
name>;ValidateServerCertificate=<true|false>;KeyStore=<keystore  
location>;keyStorePassword=<password>
```

Configure the parameters as follows:

EncryptionMethod

Encryption method for data transfer between Metadata Manager and the database server. Must be set to SSL.

TrustStore

Path and file name of the truststore file that contains the security certificate of the database server.

TrustStorePassword

Password used to access the truststore file.

HostNameInCertificate

Host name of the machine that hosts the secure database. If you specify a host name, the Metadata Manager Service validates the host name included in the connection string against the host name in the security certificate.

ValidateServerCertificate

Indicates whether the Metadata Manager Service validates the certificate that the database server presents. If you set this parameter to true, the Metadata Manager Service validates the certificate. If you specify the HostNameInCertificate parameter, the Metadata Manager Service also validates the host name in the certificate.

If you set this parameter to false, the Metadata Manager Service does not validate the certificate that the database server presents. The Metadata Manager Service ignores any truststore information that you specify.

KeyStore

Path and file name of the keystore file that contains the security certificates that the Metadata Manager Service presents to the database server.

KeyStorePassword

Password used to access the keystore file.

Configuration Properties

To edit the configuration properties, select the Metadata Manager Service in the Navigator, select the **Properties** view, and then click **Edit** in the Configuration Properties section.

The following table describes the configuration properties for a Metadata Manager Service:

Property	Description
URLScheme	Indicates the security protocol that you configure for the Metadata Manager application: HTTP or HTTPS.
Keystore File	Keystore file that contains the keys and certificates required if you configure a secure connection for the Metadata Manager web application. You must use the same security protocol for the Metadata Manager Agent if you install it on another machine.
Keystore Password	Password for the keystore file.
MaxConcurrentRequests	Maximum number of request processing threads available, which determines the maximum number of client requests that Metadata Manager can handle simultaneously. Default is 100.
MaxQueueLength	Maximum queue length for incoming connection requests when all possible request processing threads are in use by the Metadata Manager application. Metadata Manager refuses client requests when the queue is full. Default is 500.

You can use the MaxConcurrentRequests property to set the number of clients that can connect to Metadata Manager. You can use the MaxQueueLength property to set the number of client requests Metadata Manager can process at one time.

You can change the parameter values based on the number of clients that you expect to connect to Metadata Manager. For example, you can use smaller values in a test environment. In a production environment, you can increase the values. If you increase the values, more clients can connect to Metadata Manager, but the connections might use more system resources.

Connection Pool Properties

To edit the connection pool properties, select the Metadata Manager Service in the Navigator, select the **Properties** view, and then click **Edit** in the Connection Pool Properties section.

The following table describes the connection pool properties for a Metadata Manager Service:

Property	Description
Maximum Active Connections	<p>Number of active connections to the Metadata Manager repository database available. The Metadata Manager application maintains a connection pool for connections to the repository database.</p> <p>Increase the number of maximum active connections when you increase the number of maximum concurrent resource loads. For example, if you set the Max Concurrent Resource Load property to 10, Informatica recommends that you also set this property to 50 or more.</p> <p>Default is 20.</p>
Maximum Wait Time	<p>Amount of time in seconds that Metadata Manager holds database connection requests in the connection pool. If Metadata Manager cannot process the connection request to the repository within the wait time, the connection fails.</p> <p>Default is 180.</p>

Advanced Properties

To edit the advanced properties, select the Metadata Manager Service in the Navigator, select the **Properties** view, and then click **Edit** in the Advanced Properties section.

The following table describes the advanced properties for a Metadata Manager Service:

Property	Description
Max Heap Size	<p>Amount of RAM in megabytes allocated to the Java Virtual Manager (JVM) that runs Metadata Manager. Use this property to increase the performance of Metadata Manager.</p> <p>For example, you can use this value to increase the performance of Metadata Manager during indexing.</p> <p>Note: If you create Cloudera Navigator resources, set this property to at least 4096 MB (4 GB). Default is 4096.</p>
Maximum Catalog Child Objects	<p>Number of child objects that appear in the Metadata Manager metadata catalog for any parent object. The child objects can include folders, logical groups, and metadata objects. Use this option to limit the number of child objects that appear in the metadata catalog for any parent object.</p> <p>Default is 100.</p>
Error Severity Level	<p>Level of error messages written to the Metadata Manager Service log. Specify one of the following message levels:</p> <ul style="list-style-type: none">- Fatal- Error- Warning- Info- Trace- Debug <p>When you specify a severity level, the log includes all errors at that level and above. For example, if the severity level is Warning, the log includes fatal, error, and warning messages. Use Trace or Debug if Informatica Global Customer Support instructs you to use that logging level for troubleshooting purposes.</p> <p>Default is Error.</p>

Property	Description
Max Concurrent Resource Load	<p>Maximum number of resources that Metadata Manager can load simultaneously. Maximum is 10. Metadata Manager adds resource loads to the load queue in the order that you request the loads. If you simultaneously load more than the maximum, Metadata Manager adds the resource loads to the load queue in a random order. For example, you set the property to 5 and schedule eight resource loads to run at the same time. Metadata Manager adds the eight loads to the load queue in a random order. Metadata Manager simultaneously processes the first five resource loads in the queue. The last three resource loads wait in the load queue.</p> <p>If a resource load succeeds, fails and cannot be resumed, or fails during the path building task and can be resumed, Metadata Manager removes the resource load from the queue. Metadata Manager starts processing the next load waiting in the queue.</p> <p>If a resource load fails when the PowerCenter Integration Service runs the workflows and the workflows can be resumed, the resource load is resumable. Metadata Manager keeps the resumable load in the load queue until the timeout interval is exceeded or until you resume the failed load. Metadata Manager includes a resumable load due to a failure during workflow processing in the concurrent load count.</p> <p>Default is 3.</p> <p>Note: If you increase the number of maximum concurrent resource loads, increase the number of maximum active connections to the Metadata Manager repository database. For example, if you set this property to 10, Informatica recommends that you also set the Maximum Active Connections property to 50 or more.</p>
Timeout Interval	<p>Amount of time in minutes that Metadata Manager holds a resumable resource load in the load queue. You can resume a resource load within the timeout period if the load fails when PowerCenter runs the workflows and the workflows can be resumed. If you do not resume a failed load within the timeout period, Metadata Manager removes the resource from the load queue.</p> <p>Default is 30.</p> <p>Note: If a resource load fails during the path building task, you can resume the failed load at any time.</p>

Custom Properties for the Metadata Manager Service

Configure custom properties that are unique to specific environments.

You might need to apply custom properties in special cases. When you define a custom property, enter the property name and an initial value. Define custom properties only at the request of Informatica Global Customer Support.

Configuring the Associated PowerCenter Integration Service

You can configure or remove the PowerCenter Integration Service that Metadata Manager uses to load metadata into the Metadata Manager warehouse. If you remove the PowerCenter Integration Service, configure another PowerCenter Integration Service to enable the Metadata Manager Service.

To edit the associated PowerCenter Integration Service properties, select the Metadata Manager Service in the Navigator, select the **Associated Services** view, and click **Edit**. To apply changes, restart the Metadata Manager Service.

The following table describes the associated PowerCenter Integration Service properties:

Property	Description
Associated Integration Service	Name of the PowerCenter Integration Service that you want to use with Metadata Manager.
Repository User Name	Name of the PowerCenter repository user that has the required privileges. Not available for a domain with Kerberos authentication.
Repository Password	Password for the PowerCenter repository user. Not available for a domain with Kerberos authentication.
Security Domain	Name of the security domain to which the PowerCenter repository user belongs.

Privileges for the Associated PowerCenter Integration Service User

The PowerCenter repository user for the associated PowerCenter Integration Service must be able to perform the following tasks:

- Restore the PowerCenter repository.
- Import and export PowerCenter repository objects.
- Create, edit, and delete connection objects in the PowerCenter repository.
- Create folders in the PowerCenter repository.
- Load metadata into the Metadata Manager warehouse.

To perform these tasks, the user must have the required privileges and permissions for the domain, PowerCenter Repository Service, and Metadata Manager Service.

The following table lists the required privileges and permissions that the PowerCenter repository user for the associated PowerCenter Integration Service must have:

Service	Privileges	Permissions
Domain	<ul style="list-style-type: none">- Access Informatica Administrator- Manage Services	Permission on PowerCenter Repository Service
PowerCenter Repository Service	<ul style="list-style-type: none">- Access Repository Manager- Create Folders- Create, Edit, and Delete Design Objects- Create, Edit, and Delete Sources and Targets- Create, Edit, and Delete Run-time Objects- Manage Run-time Object Execution- Create Connections	<ul style="list-style-type: none">- Read, Write, and Execute on all connection objects created by the Metadata Manager Service- Read, Write, and Execute on the Metadata Load folder and all folders created to extract profiling data from the Metadata Manager source
Metadata Manager Service	Load Resource	-

In the PowerCenter repository, the user who creates a folder or connection object is the owner of the object. The object owner or a user assigned the Administrator role for the PowerCenter Repository Service can delete repository folders and connection objects. If you change the associated PowerCenter Integration

Service user, you must assign this user as the owner of the following repository objects in the PowerCenter Client:

- All connection objects created by the Metadata Manager Service
- The Metadata Load folder and all profiling folders created by the Metadata Manager Service

CHAPTER 9

Model Repository Service

This chapter includes the following topics:

- [Model Repository Service Overview, 192](#)
- [Model Repository Architecture, 193](#)
- [Model Repository Connectivity, 193](#)
- [Model Repository Database Requirements, 194](#)
- [Enable and Disable Model Repository Services and Processes, 196](#)
- [Properties for the Model Repository Service, 198](#)
- [Properties for the Model Repository Service Process, 203](#)
- [High Availability for the Model Repository Service, 206](#)
- [Model Repository Service Management, 206](#)
- [Repository Object Administration, 216](#)
- [Creating a Model Repository Service, 218](#)

Model Repository Service Overview

The Model Repository Service manages the Model repository. The Model repository stores metadata created by Informatica products in a relational database to enable collaboration among the products. Informatica Developer, Informatica Analyst, Data Integration Service, and the Administrator tool store metadata in the Model repository.

Use the Administrator tool or the *infacmd* command line program to administer the Model Repository Service. Create one Model Repository Service for each Model repository. When you create a Model Repository Service, you can create a Model repository or use an existing Model repository. You can run multiple Model Repository Services on the same node.

Manage users, groups, privileges, and roles on the Security tab of the Administrator tool. Manage permissions for Model repository objects in the Informatica Developer and the Informatica Analyst.

Based on your license, the Model Repository Service can be highly available.

Model Repository Architecture

The Model Repository Service process fetches, inserts, and updates metadata in the Model repository database tables. A Model Repository Service process is an instance of the Model Repository Service on the node where the Model Repository Service runs.

The Model Repository Service receives requests from the following client applications:

- Informatica Developer. Informatica Developer connects to the Model Repository Service to create, update, and delete objects. Informatica Developer and Informatica Analyst share objects in the Model repository.
- Informatica Analyst. Informatica Analyst connects to the Model Repository Service to create, update, and delete objects. Informatica Developer and Informatica Analyst client applications share objects in the Model repository.
- Data Integration Service. When you start a Data Integration Service, it connects to the Model Repository Service. The Data Integration Service connects to the Model Repository Service to run or preview project components. The Data Integration Service also connects to the Model Repository Service to store run-time metadata in the Model repository. Application configuration and objects within an application are examples of run-time metadata.

Note: A Model Repository Service can be associated with one Analyst Service and multiple Data Integration Services.

Model Repository Objects

The Model Repository Service stores design-time and run-time objects in the Model repository. The Developer and Analyst tools create, update, and manage the design-time objects in the Model repository. The Data Integration Service creates and manages run-time objects and metadata in the Model repository.

When you deploy an application to the Data Integration Service, the Deployment Manager copies application objects to the Model repository associated with the Data Integration Service. Run-time metadata generated during deployment are stored in the Model repository.

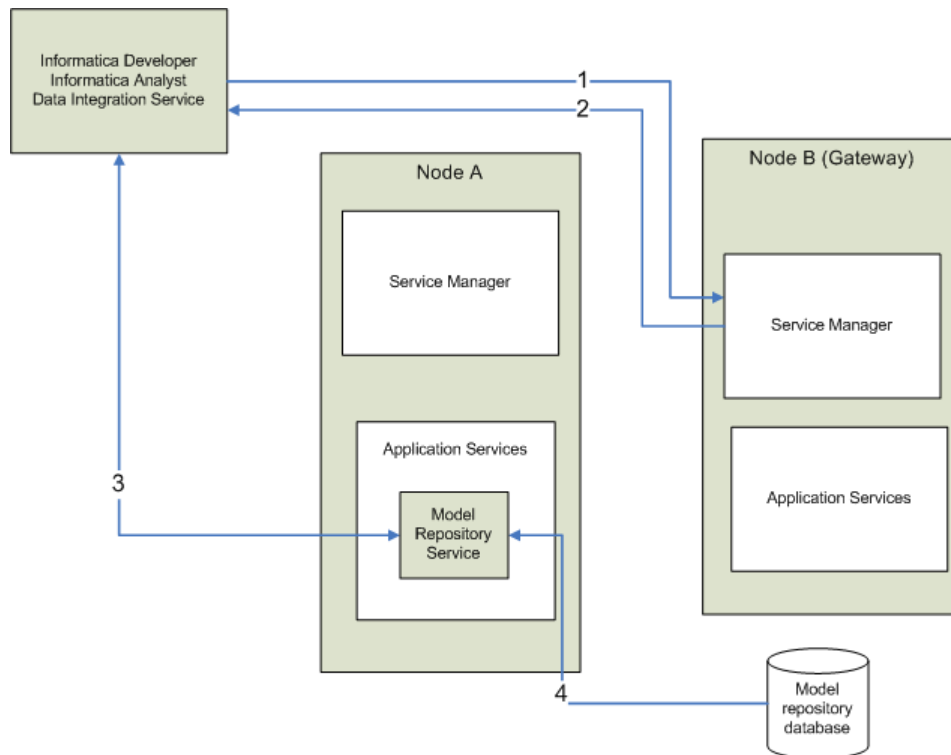
If you replace or redeploy an application, the previous version is deleted from the repository. If you rename an application, the previous application remains in the Model repository.

The Model repository locks objects by default, and when the Model repository is integrated with a version control system, you can manage checked out objects. For more information, see [“Repository Object Administration” on page 216](#).

Model Repository Connectivity

The Model Repository Service connects to the Model repository using JDBC drivers. Informatica Developer, Informatica Analyst, Informatica Administrator, and the Data Integration Service communicate with the Model Repository Service over TCP/IP. Informatica Developer, Informatica Analyst, and Data Integration Service are Model repository clients.

The following figure shows how a Model repository client connects to the Model repository database:



1. A Model repository client sends a repository connection request to the master gateway node, which is the entry point to the domain.
2. The Service Manager sends back the host name and port number of the node running the Model Repository Service. In the diagram, the Model Repository Service is running on node A.
3. The repository client establishes a TCP/IP connection with the Model Repository Service process on node A.
4. The Model Repository Service process communicates with the Model repository database over JDBC. The Model Repository Service process stores objects in or retrieves objects from the Model repository database based on requests from the Model repository client.

Note: The Model repository tables have an open architecture. Although you can view the repository tables, never manually edit them through other utilities. Informatica is not responsible for corrupted data that is caused by customer alteration of the repository tables or data within those tables.

Model Repository Database Requirements

Before you create a repository, you need a database to store repository tables. Use the database client to create the database. After you create a database, you can use the Administrator tool to create a Model Repository Service.

Each Model repository must meet the following requirements:

- Each Model repository must have its own schema. Two Model repositories or the Model repository and the domain configuration database cannot share the same schema.
- Each Model repository must have a unique database name.

In addition, each Model repository must meet database-specific requirements.

Note: The Model Repository Service uses the DataDirect drivers included with the Informatica installation. Informatica does not support the use of any other database driver.

IBM DB2 Database Requirements

Use the following guidelines when you set up the repository on IBM DB2:

- If the repository is in an IBM DB2 9.7 database, verify that IBM DB2 Version 9.7 Fix Pack 7 or a later fix pack is installed.
- On the IBM DB2 instance where you create the database, set the following parameters to ON:
 - DB2_SKIPINSERTED
 - DB2_EVALUNCOMMITTED
 - DB2_SKIPDELETED
 - AUTO_RUNSTATS
- On the database, set the configuration parameters.

The following table lists the configuration parameters that you must set:

Parameter	Value
logfilsiz	8000
maxlocks	98
locklist	50000
auto_stmt_stats	ON

- Set the tablespace pageSize parameter to 32768 bytes.

In a single-partition database, specify a tablespace that meets the pageSize requirements. If you do not specify a tablespace, the default tablespace must meet the pageSize requirements.

In a multi-partition database, specify a tablespace that meets the pageSize requirements. Define the tablespace in the catalog partition of the database.

- Set the NPAGES parameter to at least 5000. The NPAGES parameter determines the number of pages in the tablespace.
- Verify that the database user has CREATETAB, CONNECT, and BINDADD privileges.
- Informatica does not support IBM DB2 table aliases for repository tables. Verify that table aliases have not been created for any tables in the database.
- In the DataDirect Connect for JDBC utility, update the DynamicSections parameter to 3000.

The default value for DynamicSections is too low for the Informatica repositories. Informatica requires a larger DB2 package than the default. When you set up the DB2 database for the domain configuration repository or a Model repository, you must set the DynamicSections parameter to at least 3000. If the DynamicSections parameter is set to a lower number, you can encounter problems when you install or run Informatica services.

For more information about updating the DynamicSections parameter, see [Appendix D, "Updating the DynamicSections Parameter of a DB2 Database" on page 435](#).

IBM DB2 Version 9.1

If the Model repository is in an IBM DB2 9.1 database, run the DB2 reorgchk command to optimize database operations. The reorgchk command generates the database statistics used by the DB2 optimizer in queries and updates.

Use the following command:

```
REORGCHK UPDATE STATISTICS on SCHEMA <SchemaName>
```

Run the command on the database after you create the repository content.

Microsoft SQL Server Database Requirements

Use the following guidelines when you set up the repository on Microsoft SQL Server:

- Set the allow snapshot isolation and read committed isolation level to ALLOW_SNAPSHOT_ISOLATION and READ_COMMITTED_SNAPSHOT to minimize locking contention.

To set the isolation level for the database, run the following commands:

```
ALTER DATABASE DatabaseName SET ALLOW_SNAPSHOT_ISOLATION ON
```

```
ALTER DATABASE DatabaseName SET READ_COMMITTED_SNAPSHOT ON
```

To verify that the isolation level for the database is correct, run the following commands:

```
SELECT snapshot_isolation_state FROM sys.databases WHERE name=[DatabaseName]
```

```
SELECT is_read_committed_snapshot_on FROM sys.databases WHERE name = DatabaseName
```

- The database user account must have the CONNECT, CREATE TABLE, and CREATE VIEW privileges.

Oracle Database Requirements

Use the following guidelines when you set up the repository on Oracle:

- Verify that the database user has the following privileges:

```
CREATE SEQUENCE
```

```
CREATE SESSION
```

```
CREATE SYNONYM
```

```
CREATE TABLE
```

```
CREATE VIEW
```

- Informatica does not support Oracle public synonyms for repository tables. Verify that public synonyms have not been created for any tables in the database.

Enable and Disable Model Repository Services and Processes

You can enable and disable the entire Model Repository Service or a single Model Repository Service process on a particular node. If you run the Model Repository Service with the high availability option, you have one Model Repository Service process configured for each node. The Model Repository Service runs the Model Repository Service process on the primary node.

Enable, Disable, or Recycle the Model Repository Service

You can enable, disable, or recycle the Model Repository Service. You might disable the service to perform maintenance or to temporarily restrict users from accessing the Model Repository Service or Model repository. You might recycle the service if you changed a service property.

You must enable the Model Repository Service to perform the following tasks in the Administrator tool:

- Create, back up, restore, delete, or upgrade Model repository content.
- Create and delete the Model repository search index.
- Manage permissions on the Model repository.
- Synchronize the Model repository with a version control system.

Note: When you enable the Model Repository Service, the machine on which the service runs requires at least 750 MB of free memory. If enough free memory is not available, the service might fail to start.

When you enable a Model Repository Service that runs on a single node, a service process starts on the node. When you enable a Model Repository Service configured to run on primary and back-up nodes, a service process is available to run on each node, but it might not start. For example, you have the high availability option and you configure a Model Repository Service to run on a primary node and two back-up nodes. You enable the Model Repository Service, which enables a service process on each of the three nodes. A single process runs on the primary node, and the other processes on the back-up nodes maintain standby status.

When you disable the Model Repository Service, you shut down the Model Repository Service and disable all service processes.

When you disable the Model Repository Service, you must choose the mode to disable it in. You can choose one of the following options:

- **Complete.** Allows the service operations to run to completion before disabling the service.
- **Abort.** Tries to stop all service operations before aborting them and disabling the service.

When you recycle the Model Repository Service, the Service Manager restarts the Model Repository Service.

Enabling, Disabling, or Recycling the Service

You can enable, disable, or recycle the service from the Administrator tool.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the service.
3. On the **Manage** tab **Actions** menu, click one of the following options:
 - **Enable Service** to enable the service.
 - **Disable Service** to disable the service.

Choose the mode to disable the service in. Optionally, you can choose to specify whether the action was planned or unplanned, and enter comments about the action. If you complete these options, the information appears in the **Events** and **Command History** panels in the **Domain** view on the **Manage** tab.

- **Recycle Service** to recycle the service.

Enable or Disable a Model Repository Service Process

You can enable or disable a Model Repository Service process on a particular node.

When the Model Repository Service runs on a single node, disabling the service process disables the service.

When you have the high availability option and you configure the Model Repository Service to run on primary and back-up nodes, disabling a service process does not disable the service. Disabling a service process that is running causes the service to fail over to another node.

Enabling or Disabling a Service Process

You can enable or disable a service process from the Administrator tool.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the service.
3. In the contents panel, click the **Processes** view.
4. On the **Manage** tab **Actions** menu, click one of the following options:
 - **Enable Process** to enable the service process.
 - **Disable Process** to disable the service process. Choose the mode to disable the service process in.

Properties for the Model Repository Service

Use the Administrator tool to configure the following service properties:

- General properties
- Repository database properties
- Search properties
- Advanced properties
- Cache properties
- Versioning properties
- Custom properties

If you update any of the properties, you must restart the Model Repository Service for the modifications to take effect.

If you modify the repository database for a Model Repository Service that is configured for monitoring, then you must restart the domain. If you do not restart the domain after you modify the repository database, then the Model Repository Service does not resume statistics collection.

General Properties for the Model Repository Service

The following table describes the general properties for the service:

Property	Description
Name	Name of the service. The name is not case sensitive and must be unique within the domain. It cannot exceed 128 characters or begin with @. It also cannot contain spaces or the following special characters: ` ~ % ^ * + = { } \ ; : ' " / ? . , < > ! ()] [You cannot change the name of the service after you create it.
Description	Description of the service. The description cannot exceed 765 characters.

Property	Description
License	License object that allows use of the service.
Node	Node on which the service runs.
Backup Nodes	If your license includes high availability, nodes on which the service can run if the primary node is unavailable.

Repository Database Properties for the Model Repository Service

The following table describes the database properties for the Model repository:

Property	Description
Database Type	The type of database.
Username	The database user name for the Model repository.
Password	Repository database password for the database user.
JDBC Connect String	<p>The JDBC connection string used to connect to the Model repository database. Use the following JDBC connect string syntax for each supported database:</p> <ul style="list-style-type: none"> - IBM DB2. <code>jdbc:informatica:db2://<host_name>:<port_number>;DatabaseName=<database_name>;BatchPerformanceWorkaround=true;DynamicSections=3000</code> - Microsoft SQL Server that uses the default instance. <code>jdbc:informatica:sqlserver://<host_name>:<port_number>;DatabaseName=<database_name>;SnapshotSerializable=true</code> - Microsoft SQL Server that uses a named instance. <code>jdbc:informatica:sqlserver://<host_name>\<named_instance_name>;DatabaseName=<database_name>;SnapshotSerializable=true</code> - Oracle. <code>jdbc:informatica:oracle://<host_name>:<port_number>;SID=<database_name>;MaxPooledStatements=20;CatalogOptions=0;BatchPerformanceWorkaround=true</code>
Secure JDBC Parameters	<p>If the Model repository database is secured with the SSL protocol, you must enter the secure database parameters.</p> <p>Enter the parameters as <code>name=value</code> pairs separated by semicolon characters (;). For example:</p> <pre>param1=value1;param2=value2</pre>
Dialect	<p>The SQL dialect for a particular database. The dialect maps java objects to database objects.</p> <p>For example:</p> <pre>org.hibernate.dialect.Oracle9Dialect</pre>
Driver	<p>The Data Direct driver used to connect to the database.</p> <p>For example:</p> <pre>com.informatica.jdbc.oracle.OracleDriver</pre>

Property	Description
Database Schema	The schema name for a particular database.
Database Tablespace	The tablespace name for a particular database. For a multi-partition IBM DB2 database, the tablespace must span a single node and a single partition.

JDBC Parameters for Secure Databases

If the Model repository database is secured with the SSL protocol, you must enter the secure database parameters in the **Secure JDBC Parameters** field.

Enter the parameters as `name=value` pairs separated by semicolon characters (;). For example:

```
param1=value1;param2=value2
```

Enter the following secure database parameters:

Secure Database Parameter	Description
EncryptionMethod	Required. Indicates whether data is encrypted when transmitted over the network. This parameter must be set to <code>SSL</code> .
ValidateServerCertificate	Optional. Indicates whether Informatica validates the certificate that the database server sends. If this parameter is set to <code>True</code> , Informatica validates the certificate that the database server sends. If you specify the <code>HostNameInCertificate</code> parameter, Informatica also validates the host name in the certificate. If this parameter is set to <code>False</code> , Informatica does not validate the certificate that the database server sends. Informatica ignores any truststore information that you specify.
HostNameInCertificate	Optional. Host name of the machine that hosts the secure database. If you specify a host name, Informatica validates the host name included in the connection string against the host name in the SSL certificate.
cryptoProtocolVersion	Required. Specifies the cryptographic protocol to use to connect to a secure database. You can set the parameter to <code>cryptoProtocolVersion=TLSv1.1</code> or <code>cryptoProtocolVersion=TLSv1.2</code> based on the cryptographic protocol used by the database server.
TrustStore	Required. Path and file name of the truststore file that contains the SSL certificate for the database. If you do not include the path for the truststore file, Informatica looks for the file in the following default directory: <code><Informatica installation directory>/tomcat/bin</code>
TrustStorePassword	Required. Password for the truststore file for the secure database.

Note: Informatica appends the secure JDBC parameters to the JDBC connection string. If you include the secure JDBC parameters directly in the connection string, do not enter any parameter in the **Secure JDBC Parameters** field.

Search Properties for the Model Repository Service

The following table describes the search properties for the Model Repository Service:

Property	Description
Search Analyzer	<p>Fully qualified Java class name of the search analyzer.</p> <p>By default, the Model Repository Service uses the following search analyzer for English:</p> <pre>com.informatica.repository.service.provider.search.analysis.MMStandardAnalyzer</pre> <p>You can specify the following Java class name of the search analyzer for Chinese, Japanese and Korean languages:</p> <pre>org.apache.lucene.analysis.cjk.CJKAnalyzer</pre> <p>Or, you can create and specify a custom search analyzer.</p>
Search Analyzer Factory	<p>Fully qualified Java class name of the factory class if you used a factory class when you created a custom search analyzer.</p> <p>If you use a custom search analyzer, enter the name of either the search analyzer class or the search analyzer factory class.</p>

Advanced Properties for the Model Repository Service

The following table describes the Advanced properties for the Model Repository Service:

Property	Description
Maximum Heap Size	<p>Amount of RAM allocated to the Java Virtual Machine (JVM) that runs the Model Repository Service. Use this property to increase the performance. Append one of the following letters to the value to specify the units:</p> <ul style="list-style-type: none">- b for bytes.- k for kilobytes.- m for megabytes.- g for gigabytes. <p>Default value is 1024m.</p>
JVM Command Line Options	<p>Java Virtual Machine (JVM) command line options to run Java-based programs. When you configure the JVM options, you must set the Java SDK classpath, Java SDK minimum memory, and Java SDK maximum memory properties.</p> <p>You must set the following JVM command line options:</p> <ul style="list-style-type: none">- Xms. Minimum heap size. Default value is 256 m.- Xss. Stack size. Default value is 512 k.- MaxMetaspaceSize . Maximum permanent generation size. Default is 512 m.- Dfile.encoding. File encoding. Default is UTF-8.

Cache Properties for the Model Repository Service

The following table describes the cache properties for the Model Repository Service:

Property	Description
Enable Cache	Enables the Model Repository Service to store Model repository objects in cache memory. To apply changes, restart the Model Repository Service.
Cache JVM Options	JVM options for the Model Repository Service cache. To configure the amount of memory allocated to cache, configure the maximum heap size. This field must include the maximum heap size, specified by the -Xmx option. The default value and minimum value for the maximum heap size is -Xmx128m. The options you configure apply when Model Repository Service cache is enabled. To apply changes, restart the Model Repository Service. The options you configure in this field do not apply to the JVM that runs the Model Repository Service.

Versioning Properties for the Model Repository Service

To connect to a version control system, you must configure versioning properties in the Model Repository Service.

You can configure versioning properties for the Perforce or Subversion version control systems. Subversion is abbreviated "SVN".

Some of the properties refer to the version control system host machine and user accounts. Contact the version control system administrator for this information.

After you configure versioning properties, you restart the Model repository, and then run `infacmd mrs PopulateVCS` to synchronize Model repository contents to the version control system.

Note: While the Model repository synchronizes its contents with the version control system for the first time, the Model repository is unavailable. Model repository users must close all editable objects before the process starts.

The following table describes the versioning properties for the Model Repository Service:

Property	Description
Version control system type	The supported version control system that you want to connect to. You can choose Perforce or SVN.
Host	The URL, IP address, or host name of the machine where the Perforce version control system runs. Not available when you configure SVN as the version control system.
URL	The URL of the SVN version control system repository or the subfolder. Not available when you configure Perforce as the version control system.
Port	Required. Port number that the version control system host uses to listen for requests from the Model Repository Service.

Property	Description
Path to repository objects	<p>Path to the root directory of the version control system that stores the Model repository objects.</p> <p>Note: When you complete editing Versioning properties, the Model repository connects to the version control system and generates the specified directory if the directory does not exist yet.</p> <p>Only one Model Repository Service can use this directory.</p> <p>For Perforce, use the syntax:</p> <pre>//directory/path</pre> <p>where <code>directory</code> is the Perforce directory root, and <code>path</code> is the remainder of the path to the root directory of Model repository objects.</p> <p>Example:</p> <pre>//depot/Informatica/repository_copy</pre> <p>When you configure SVN as the version control system, this option is not available.</p> <p>Note: If you change the depot path after you synchronize the Model repository with the version control system, version history for objects in the Model repository is lost.</p>
Username	<p>User account for the version control system user.</p> <p>The account type for a Subversion version control system must be a Subversion user and not Windows login user.</p> <p>This account must have write permissions on the version control system.</p> <p>The account type for a Perforce version control system must be a Standard user.</p>
Password	Password for the version control system user.

Custom Properties for the Model Repository Service

Configure custom properties that are unique to specific environments.

You might need to apply custom properties in special cases. When you define a custom property, enter the property name and an initial value. Define custom properties only at the request of Informatica Global Customer Support.

Properties for the Model Repository Service Process

The Model Repository Service runs the Model Repository Service process on one node. When you select the Model Repository Service in the Administrator tool, you can view information about the Model Repository Service process on the Processes tab. You can also configure search and logging for the Model Repository Service process.

Note: You must select the node to view the service process properties in the Service Process Properties section.

Node Properties for the Model Repository Service Process

Use the Administrator tool to configure the following types of Model Repository Service process properties:

- Search properties

- Repository performance properties
- Audit properties
- Repository log properties
- Custom properties
- Environment variables

Search Properties for the Model Repository Service Process

Search properties for the Model Repository Service process.

The following table describes the search properties for the Model Repository Service process:

Property	Description
Search Index Root Directory	<p>The directory that contains the search index files.</p> <p>Default is:</p> <pre><Informatica_Installation_Directory>/tomcat/bin/target/repository/ <system_time>/<service_name>/index</pre> <p><code>system_time</code> is the system time when the directory is created.</p>

Repository Performance Properties for the Model Repository Service Process

Performance tuning properties for storage of data objects in the Model Repository Service.

The Model Repository Service uses an open source object-relational mapping tool called Hibernate to map and store data objects and metadata to the Model repository database. For each service process, you can set Hibernate options to configure connection and statement pooling for the Model repository.

The following table describes the performance properties for the Model Repository Service process:

Property	Description
Hibernate Connection Pool Size	The maximum number of pooled connections in the Hibernate internal connection pooling. Equivalent to the <code>hibernate.connection.pool_size</code> property. Default is 10.
Hibernate c3p0 Minimum Size	Minimum number of connections a pool will maintain at any given time. Equivalent to the <code>c3p0 minPoolSize</code> property. Default is 1.
Hibernate c3p0 Maximum Statements	<p>Size of the c3p0 global cache for prepared statements. This property controls the total number of statements cached. Equivalent to the <code>c3p0 maxStatements</code> property. Default is 1000.</p> <p>The Model Repository Service uses the value of this property to set the <code>c3p0 maxStatementsPerConnection</code> property based on the number of connections set in the Hibernate Connection Pool Size property.</p>

Audit Properties for the Model Repository Service Process

Audit properties for the Model Repository Service process.

The following table describes the audit properties for the Model Repository Service process:

Property	Description
Audit Enabled	Displays audit logs in the Log Viewer. Default is False.

Repository Logs for the Model Repository Service Process

Repository log properties for the Model Repository Service process.

The following table describes the repository log properties for the Model Repository Service process:

Property	Description
Repository Logging Directory	The directory that stores logs for Log Persistence Configuration or Log Persistence SQL. To disable the logs, do not specify a logging directory. These logs are not the repository logs that appear in the Log Viewer. Default is blank.
Log Level	<p>The severity level for repository logs.</p> <ul style="list-style-type: none">- Fatal. Writes FATAL messages to the log. FATAL messages include nonrecoverable system failures that cause the service to shut down or become unavailable.- Error. Writes FATAL and ERROR code messages to the log. ERROR messages include connection failures, failures to save or retrieve metadata, service errors.- Warning. Writes FATAL, WARNING, and ERROR messages to the log. WARNING errors include recoverable system failures or warnings.- Info. Writes FATAL, INFO, WARNING, and ERROR messages to the log. INFO messages include system and service change messages.- Trace. Write FATAL, TRACE, INFO, WARNING, and ERROR code messages to the log. TRACE messages log user request failures.- Debug. Write FATAL, DEBUG, TRACE, INFO, WARNING, and ERROR messages to the log. DEBUG messages are user request logs. <p>The default value is Info.</p>
Log Persistence Configuration to File	Indicates whether to write persistence configuration to a log file. The Model Repository Service logs information about the database schema, object relational mapping, repository schema change audit log, and registered IMF packages. The Model Repository Service creates the log file when the Model repository is enabled, created, or upgraded. The Model Repository Service stores the logs in the specified repository logging directory. If a repository logging directory is not specified, the Model Repository Service does not generate the log files. You must disable and re-enable the Model Repository Service after you change this option. Default is False.
Log Persistence SQL to File	Indicates whether to write parameterized SQL statements to a log file in the specified repository logging directory. If a repository logging directory is not specified, the Model Repository Service does not generate the log files. You must disable and re-enable the Model Repository Service after you change this option. Default is False.

Custom Properties for the Model Repository Service Process

Configure custom properties that are unique to specific environments.

You might need to apply custom properties in special cases. When you define a custom property, enter the property name and an initial value. Define custom properties only at the request of Informatica Global Customer Support.

Environment Variables for the Model Repository Service Process

You can edit environment variables for a Model Repository Service process.

The following table describes the environment variables for the Model Repository Service process:

Property	Description
Environment Variables	Environment variables defined for the Model Repository Service process.

High Availability for the Model Repository Service

Model Repository high availability features minimize interruptions to data integration tasks by enabling the Service Manager and Model Repository Service to react to network failures and failures of the Model Repository Service.

Model Repository Service high availability includes restart and failover of the service. When the Model Repository Service becomes unavailable, the Service Manager can restart the Model Repository Service on the same node or on a backup node.

For more information about how to configure a highly available domain, see the *Informatica Administrator Guide*.

Model Repository Service Restart and Failover

To minimize Model Repository Service downtime, the Service Manager restarts the Model Repository Service on the same node or on a backup node if the Model Repository Service is unavailable.

The Model Repository Service fails over to a backup node in the following situations:

- The Model Repository Service fails and the primary node is not available.
- The Model Repository Service is running on a node that fails.

The Service Manager restarts the Model Repository Service based on domain property values set for the amount of time spent trying to restart the service and the maximum number of attempts to try within the restart period.

Model Repository Service clients are resilient to temporary connection failures during failover and restart of the service.

Model Repository Service Management

Use the Administrator tool to manage the Model Repository Service and the Model repository content. For example, you can use the Administrator tool to manage repository content, search, and repository logs.

Content Management for the Model Repository Service

When you create the Model Repository Service, you can create the repository content. Alternatively, you can create the Model Repository Service using existing repository content. The repository name is the same as the name of the Model Repository Service.

You can also delete the repository content. You may choose to delete repository content to delete a corrupted repository or to increase disk or database space.

Creating and Deleting Repository Content

1. On the **Manage** tab, select the **Services and Nodes** view.
2. In the Domain Navigator, select the Model Repository Service.
3. To create the repository content, on the **Manage** tab **Actions** menu, click **Repository Contents > Create**.
4. Or, to delete repository content, on the **Manage** tab **Actions** menu, click **Repository Contents > Delete**.

If you delete and create new repository content for a Model Repository Service that is configured for monitoring, then you must restart the domain after you create new content. If you do not restart the domain, then the Model Repository Service does not resume statistics collection.

Model Repository Backup and Restoration

Regularly back up repositories to prevent data loss due to hardware or software problems. When you back up a repository, the Model Repository Service saves the repository to a file, including the repository objects and the search index. If you need to recover the repository, you can restore the content of the repository from this file.

When you back up a repository, the Model Repository Service writes the file to the service backup directory. The service backup directory is a subdirectory of the node backup directory with the name of the Model Repository Service. For example, a Model Repository Service named MRS writes repository backup files to the following location:

```
<node_backup_directory>\MRS
```

You specify the node backup directory when you set up the node. View the general properties of the node to determine the path of the backup directory. The Model Repository Service uses the extension .mrep for all Model repository backup files.

To ensure that the Model Repository Service creates a consistent backup file, the backup operation blocks all other repository operations until the backup completes. You might want to schedule repository backups when users are not logged in.

To restore the backup file of a Model Repository Service to a different Model Repository Service, you must copy the backup file and place it in backup directory of the Model Repository Service to which you want to restore the backup. For example, you want to restore the backup file of a Model Repository Service named MRS1 to a Model Repository Service named MRS2. You must copy the backup file of MRS1 from

```
<node_backup_directory>\MRS1 and place the file in <node_backup_directory>\MRS2.
```

Note: When you back up and then delete the contents of a Model repository, you must restart the Model Repository Service before you restore the contents from the backup. If you try to restore the Model repository contents and have not recycled the service, you may get an error related to search indices.

Backing Up the Repository Content

You can back up the content of a Model repository to restore the repository content to another repository or to retain a copy of the repository.

1. On the **Manage** tab, select the **Services and Nodes** view.
2. In the Domain Navigator, select the Model Repository Service.
3. On the **Manage** tab **Actions** menu, click **Repository Contents > Back Up**.
The **Back Up Repository Contents** dialog box appears.
4. Enter the following information:

Option	Description
Username	User name of any user in the domain.
Password	Password of the domain user.
SecurityDomain	Domain to which the domain user belongs. Default is Native.
Output File Name	Name of the output file.
Description	Description of the contents of the output file.

5. Click **Overwrite** to overwrite a file with the same name.
6. Click **OK**.
The Model Repository Service writes the backup file to the service backup directory.

Restoring the Repository Content

You can restore repository content to a Model repository from a repository backup file.

Verify that the repository is empty. If the repository contains content, the restore option is disabled.

1. On the **Manage** tab, select the **Services and Nodes** view.
2. In the Navigator, select the Model Repository Service.
3. On the **Manage** tab **Actions** menu, click **Repository Contents > Restore**.
The **Restore Repository Contents** dialog box appears.
4. Select a backup file to restore.
5. Enter the following information:

Option	Description
Username	User name of any user in the domain.
Password	Password of the domain user.
Security Domain	Domain to which the domain user belongs. Default is Native.

6. Click **OK**.

If the Model Repository service is configured for monitoring, then you must recycle the Model Repository Service. If you do not recycle the Model Repository Service, then the service does not resume statistics collection.

Viewing Repository Backup Files

You can view the repository backup files written to the Model Repository Service backup directory.

1. On the **Manage** tab, select the **Services and Nodes** view.
2. In the Navigator, select the Model Repository Service.
3. On the **Manage** tab **Actions** menu, click **Repository Contents > View Backup Files**.

The **View Repository Backup Files** dialog box appears and shows the backup files for the Model Repository Service.

Security Management for the Model Repository Service

You manage users, groups, privileges, and roles on the Security tab of the Administrator tool.

You manage permissions for repository objects in Informatica Developer and Informatica Analyst. Permissions control access to projects in the repository. Even if a user has the privilege to perform certain actions, the user may also require permission to perform the action on a particular object.

To secure data in the repository, you can create a project and assign permissions to it. When you create a project, you are the owner of the project by default. The owner has all permissions, which you cannot change. The owner can assign permissions to users or groups in the repository.

Search Management for the Model Repository Service

The Model Repository Service uses a search engine to create search index files.

When users perform a search, the Model Repository Service searches for metadata objects in the index files instead of the Model repository.

To correctly index the metadata, the Model Repository Service uses a search analyzer appropriate for the language of the metadata that you are indexing. The Model Repository Service includes the following packaged search analyzers:

- `com.informatica.repository.service.provider.search.analysis.MMStandardAnalyzer`. Default search analyzer for English.
- `org.apache.lucene.analysis.cjk.CJKAnalyzer`. Search analyzer for Chinese, Japanese, and Korean.

You can change the default search analyzer. You can use a packaged search analyzer or you can create and use a custom search analyzer.

The Model Repository Service stores the index files in the search index root directory that you define for the service process. The Model Repository Service updates the search index files each time a user saves, modifies, or deletes a Model repository object. You must manually update the search index if you change the search analyzer, if you create a Model Repository Service to use existing repository content, if you upgrade the Model Repository Service, or if the search index files become corrupted.

Creating a Custom Search Analyzer

If you do not want to use one of the packaged search analyzers, you can create a custom search analyzer.

1. Extend the following Apache Lucene Java class:

```
org.apache.lucene.analysis.Analyzer
```

2. If you use a factory class when you extend the Analyzer class, the factory class implementation must have a public method with the following signature:

```
public org.apache.lucene.analysis.Analyzer createAnalyzer(Properties settings)
```

The Model Repository Service uses the factory to connect to the search analyzer.

3. Place the custom search analyzer and required .jar files in the following directory:

```
<Informatica_Installation_Directory>/services/ModelRepositoryService
```

Changing the Search Analyzer

You can change the default search analyzer that the Model Repository Service uses. You can use a packaged search analyzer or you can create and use a custom search analyzer.

1. In the Administrator tool, select the **Services and Nodes** view on the **Manage** tab.
2. In the Navigator, select the Model Repository Service.
3. To use one of the packaged search analyzers, specify the fully qualified java class name of the search analyzer in the Model Repository Service search properties.
4. To use a custom search analyzer, specify the fully qualified java class name of either the search analyzer or the search analyzer factory in the Model Repository Service search properties.
5. Recycle the Model Repository Service to apply the changes.
6. Click **Actions > Search Index > Re-Index** on the **Manage** tab **Actions** menu to re-index the search index.

Manually Updating Search Index Files

You manually update the search index if you change the search analyzer, if you create a Model Repository Service to use existing repository content, if you upgrade the Model Repository Service, or if the search index files become corrupted. For example, search index files can become corrupted due to insufficient disk space in the search index root directory.

The amount of time needed to re-index depends on the number of objects in the Model repository. During the re-indexing process, design-time objects in the Model repository are read-only.

Users in the Developer tool and Analyst tool can view design-time objects but cannot edit or create design-time objects.

If you re-index after changing the search analyzer, users can perform searches on the existing index while the re-indexing process runs. When the re-indexing process completes, any subsequent user search request uses the new index.

To correct corrupted search index files, you must delete, create, and then re-index the search index. When you delete and create a search index, users cannot perform a search until the re-indexing process finishes.

You might want to manually update the search index files during a time when most users are not logged in.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the Model Repository Service.

3. To re-index after changing the search analyzer, creating the Model Repository Service to use existing repository content, or upgrading the Model Repository Service, click **Actions > Search Index > Re-Index** on the **Manage** tab **Actions** menu.
4. To correct corrupted search index files, complete the following steps on the **Manage** tab **Actions** menu:
 - a. Click **Actions > Search Index > Delete** to delete the corrupted search index.
 - b. Click **Actions > Search Index > Create** to create a search index.
 - c. Click **Actions > Search Index > Re-Index** to re-index the search index.

Repository Log Management for the Model Repository Service

The Model Repository Service generates repository logs. The repository logs contain repository messages of different severity levels, such as fatal, error, warning, info, trace, and debug. You can configure the level of detail that appears in the repository log files. You can also configure where the Model Repository Service stores the log files.

Configuring Repository Logging

1. In the Administrator tool, click the **Manage** tab.
2. Click the **Services and Nodes** view.
3. In the Domain Navigator, select the Model Repository Service.
4. In the contents panel, select the **Processes** view.
5. Select the node.

The service process details appear in the Service Process Properties section.
6. Click **Edit** in the Repository section.

The Edit Processes page appears.
7. Enter the directory path in the **Repository Logging Directory** field.
8. Specify the level of logging in the **Repository Logging Severity Level** field.
9. Click OK.

Audit Log Management for the Model Repository Service

The Model Repository Service can generate audit logs in the Log Viewer.

The audit log provides information about the following types of operations performed on the Model repository:

- Logging in and out of the Model repository.
- Creating a project.
- Creating a folder.

By default, audit logging is disabled.

Enabling and Disabling Audit Logging

1. In the Administrator tool, click the **Manage** tab.
2. Click the **Services and Nodes** view.
3. In the Domain Navigator, select the Model Repository Service.

4. In the contents panel, select the Processes view.
5. Select the node.
The service process details appear in the Service Process Properties section.
6. Click **Edit** in the Audit section.
The Edit Processes page appears.
7. Enter one of the following values in the Audit Enabled field:
 - True. Enables audit logging.
 - False. Disables audit logging. Default is false.
8. Click OK.

Cache Management for the Model Repository Service

To improve Model Repository Service performance, you can configure the Model Repository Service to use cache memory. When you configure the Model Repository Service to use cache memory, the Model Repository Service stores objects that it reads from the Model repository in memory. The Model Repository Service can read the repository objects from memory instead of the Model repository. Reading objects from memory reduces the load on the database server and improves response time.

Model Repository Cache Processing

When the cache process starts, the Model Repository Service stores each object it reads in memory. When the Model Repository Service gets a request for an object from a client application, the Model Repository Service compares the object in memory with the object in the repository. If the latest version of the object is not in memory, the Model repository updates the cache and then returns the object to the client application that requested the object. When the amount of memory allocated to cache is full, the Model Repository Service deletes the cache for least recently used objects to allocate space for another object.

The Model Repository Service cache process runs as a separate process. The Java Virtual Manager (JVM) that runs the Model Repository Service is not affected by the JVM options you configure for the Model Repository Service cache.

Configuring Cache

1. In the Administrator tool, click the **Manage** tab.
2. Click the **Services and Nodes** view.
3. In the Domain Navigator, select the Model Repository Service.
4. Click **Edit** in the **Cache Properties** section.
5. Select **Enable Cache**.
6. Specify the amount of memory allocated to cache in the **Cache JVM Options** field.
7. Restart the Model Repository Service.
8. Verify that the cache process is running.
The Model Repository Service logs display the following message when the cache process is running:

```
MRSI_35204 "Caching process has started on host [host name] at port [port number]
with JVM options [JVM options]."
```

Version Control for the Model Repository Service

You can integrate a Model repository with a version control system. Version control system integration protects Model repository objects from overwriting on a team where multiple developers work on the same projects.

To enable version control, configure versioning properties, and then synchronize the Model repository with the version control system.

You can integrate the Model repository with the Perforce or Subversion version control systems. You must use a version control system that has not been integrated with a Model repository. Only one Model repository can use a version control system instance.

You can configure versioning properties when you create or update a Model repository service.

The versioning properties include a version control system user ID and password. The connection uses this single account to access the version control system for all actions related to versioned object management. For more information about versioning properties, see [“Versioning Properties for the Model Repository Service” on page 202](#).

After you configure version control, ask Model repository users to close all editable objects, and then you restart the Model repository.

When the Model repository restarts, it checks whether the version control system is in use by another Model repository. If the version control system connects to a different Model repository, the Model Repository Service cannot restart. You must configure versioning properties to connect to a version control system that has not been integrated with a Model repository.

When you synchronize Model repository contents to the version control system, the Model repository populates a directory in the version control system depot with Model repository objects. After the Model repository copies its contents to the version control system directory, you cannot disable version control system integration.

When the Model repository is integrated with a version control system, you can perform the following tasks:

- Check in revised objects.
- Undo the checkout of objects.
- Reassign the checked-out state of objects to another user.

How to Configure and Synchronize a Model Repository with a Version Control System

To enable version control, you configure versioning properties, and then synchronize the Model repository with the version control system.

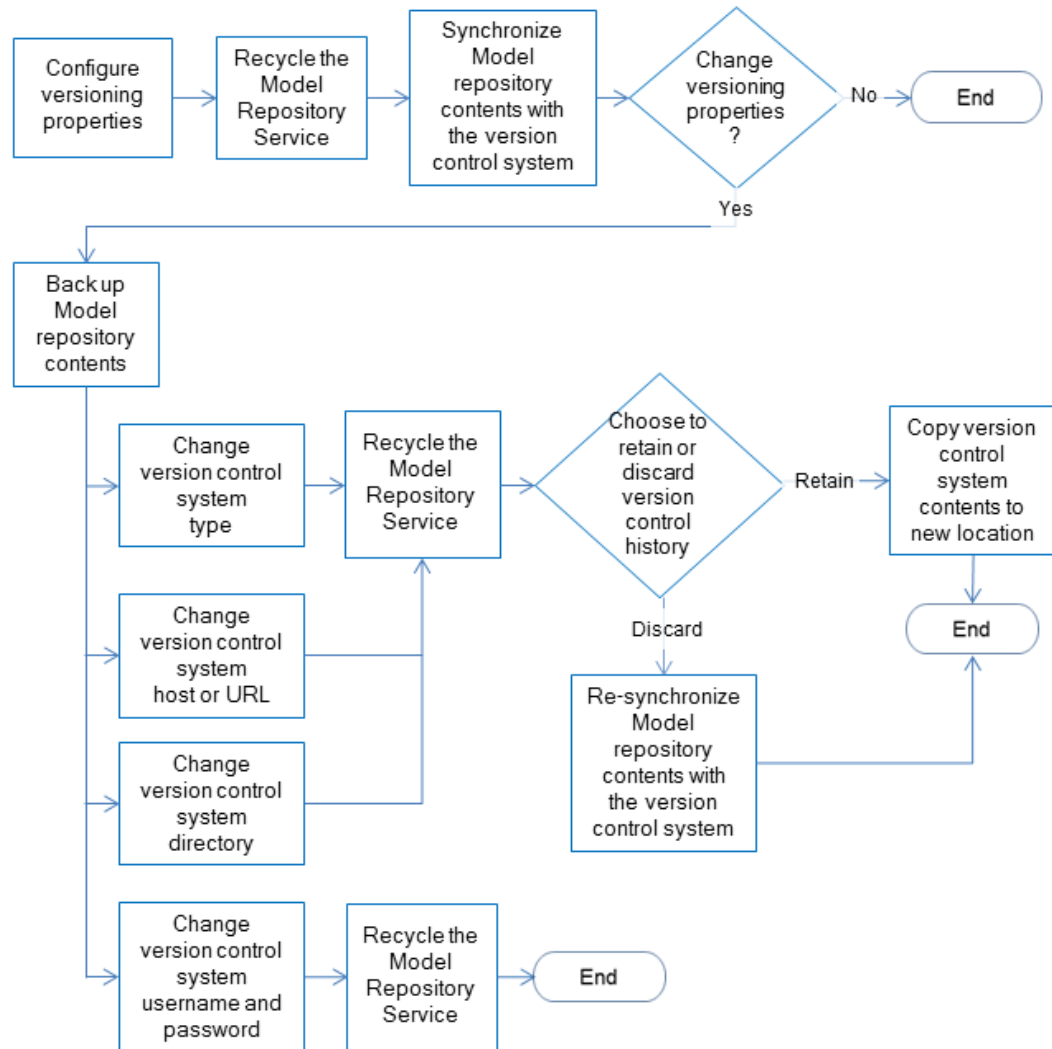
After you configure versioning and synchronize the Model repository with the version control system, the version control system begins to save version history. If you change the version control system type, host, URL, or directory properties, you can choose to retain or discard version history.

Perform one of the following tasks:

- To retain version history, manually copy the contents of the version control system directory to the new version control system location, change versioning properties, and then recycle the Model Repository Service.
- To discard version history, change versioning properties, recycle the Model Repository Service, and then re-synchronize the Model repository with the new version control system type or location.

Note: When you change Model repository properties, you must recycle the Model Repository Service for your changes to take effect. Ask users to save changes and close Model repository objects that they have open for editing. While synchronization is in progress, the Model repository is unavailable.

The following image shows the process of configuring, synchronizing, and re-synchronizing the Model repository with a version control system:



1. Configure versioning properties and restart the Model repository service.
2. Synchronize the Model repository contents with the version control system.
3. Optionally, change the version control system type.
 - a. Back up the Model repository contents.
 - b. Change the version control system type and restart the Model Repository Service.
 - c. Choose whether to retain or discard the version history:
 - To retain version history, copy the contents of the existing version control system directory to the new version control system, and configure the Model repository for the new location.

- To discard version history, re-synchronize the Model repository to the new version control system.
4. Optionally, change the version control system host or URL.
If you use Perforce as the version control system, you can change the Perforce host or port number. If you use Subversion, you can change the URL.
 - a. Back up the Model repository contents.
 - b. Change the version control system location and restart the Model Repository Service.
 - c. Choose whether to retain or discard the version history:
 - To retain version history, copy the contents of the existing version control system directory to the new version control system location, and configure the Model repository for the new location.
 - To discard version history, re-synchronize the Model repository to the new version control system host or URL.
 5. Optionally, change the version control system directory location.
 - a. Back up the Model repository contents.
 - b. Change the version control system directory and restart the Model Repository Service.
 - c. Choose whether to retain or discard the version history:
 - To retain version history, copy the contents of the existing version control system directory to the new directory, and configure the Model repository for the new location.
 - To discard version history, re-synchronize the Model repository to the new version control system directory.
 6. Optionally, change the version control system username or password.
 - a. Back up the Model repository contents.
 - b. Change the version control system type.
 - c. Restart the Model Repository Service.

You can perform these tasks from the command line or from the Administrator tool.

Synchronizing the Model Repository with a Version Control System

Before you synchronize the Model repository with the version control system, you configure versioning properties, and then recycle the Model Repository Service for property changes to take effect. Then synchronize the contents of the Model repository with the version control system.

Note: While synchronization is in progress, the Model repository is unavailable. Ask users to save changes and close Model repository objects before synchronization begins.

1. Instruct Model repository users to save changes to and close repository objects.
2. On the **Manage** tab, select the **Services and Nodes** view.
3. Select the Model repository to synchronize with the version control system.
4. Click **Actions > Synchronize With Version Control System**.
5. Click OK.

The Model Repository Service copies the contents of the repository to the version control system directory. During synchronization, the Model repository is unavailable.

When synchronization is complete, versioning is active for Model repository objects. All Model repository objects are checked in to the version control system. Users can check out, check in, view version history, and retrieve historical versions of objects.

After the Model repository is synchronized with the version control system, you cannot disable version control system integration.

Troubleshooting Team-based Development

Consider the following troubleshooting tips when you use features related to team-based development:

The Perforce version control system fails to check in some objects, with an error about excessively long object path names.

Due to Windows OS limitations on the number of characters in a file path, Model repository objects with long path and file names fail when you try to check them in. The Perforce error message reads "Submit aborted" and says the file path exceeds the internal length limit.

To work around this problem, limit the length of directory names in the path to the Perforce depot, and limit the length of project, folder, and object names in the Model repository. Shorter names in all instances help limit the total number of characters in the object path name.

The operation to synchronize the Model repository with the version control system fails.

When you attempt to synchronize the Model repository with the version control system, the operation fails with an error message from the version control system. For example, you might see an error like:

```
The Repository Service operation failed.
['[RSVCSHARED_01524] Unable to submit changes to the version control system.
Encountered the following error: '4'.']
```

To address this problem, check that the code page settings for the Model repository and the version control system are compatible, depending on your locale.

Repository Object Administration

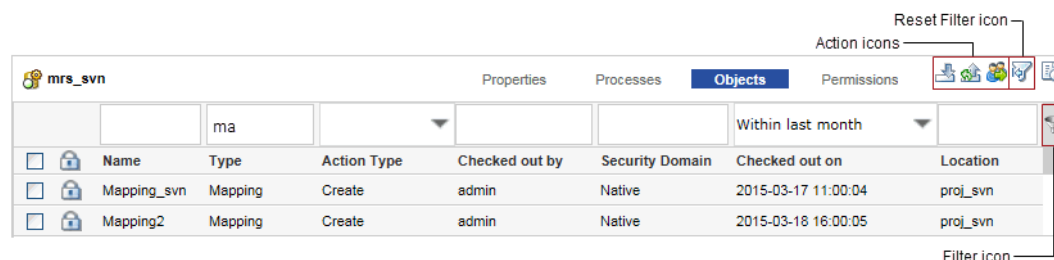
The Model repository locks objects to prevent users from overwriting work. The Model repository can lock any object that the Developer tool or the Analyst tool displays, except for projects and folders.

You can manage locked objects in a Model repository that is not integrated with a version control system. You can manage checked out objects in a Model repository that is integrated with a version control system. When the Model repository is integrated with a version control system, you can view, undo, or re-assign the checked-out state of an object.

Objects View

You can view and manage repository objects from the **Objects** tab of the Model Repository Service.

The following image shows the **Objects** tab with a filter on the Type column:



The screenshot shows the 'Objects' tab of the Model Repository Service. At the top, there are tabs for 'Properties', 'Processes', 'Objects' (selected), and 'Permissions'. Below the tabs, there is a search bar with 'ma' entered. To the right of the search bar, there is a 'Reset Filter icon' and a 'Filter icon'. The table below shows a list of objects with columns: Name, Type, Action Type, Checked out by, Security Domain, Checked out on, and Location. The table is filtered to show only objects of type 'Mapping'.

	Name	Type	Action Type	Checked out by	Security Domain	Checked out on	Location
<input type="checkbox"/>	Mapping_svn	Mapping	Create	admin	Native	2015-03-17 11:00:04	proj_svn
<input type="checkbox"/>	Mapping2	Mapping	Create	admin	Native	2015-03-18 16:00:05	proj_svn

Note: If a Model repository is not integrated with a version control system, the **Checked out on** column is replaced with **Locked on**, and the **Checked out by** column is replaced with **Locked by**.

When you manage Model repository objects, you filter the list of objects and then select an action:

1. When you open the **Objects** tab, the display is empty. Enter filter criteria in the filter bar and then click the **Filter** icon to get a list of objects to manage. For example, to display a list of objects with Type names beginning with "ma," type `ma` in the filter bar, and then click the Filter icon.
2. Select one or more objects. Then right-click a selected object and select an action, or click one of the action icons.

To reset the **Objects** tab, click the Reset Filter icon.

Locked Object Administration

If the Developer tool or the Analyst tool shuts down, or if the Model repository becomes unavailable, objects retain locks. After the Model repository becomes available, you can view locked objects and unlock them.

You might want to unlock objects if the user who locked them is unavailable and another user is assigned to edit them.

You can perform the following operations:

List locked objects.

You can list the objects that are locked in the Model repository. You can filter the list by the time that a user locked the object. You might want to do this to identify the developers working on each object.

Unlock an object.

You can unlock any object that is locked in the Model repository.

Note: When you unlock a locked object that a user edited, the changes are lost.

Versioned Object Administration

If a developer is not available to check in a checked-out object, you can list and undo or reassign the checked-out state of an object.

You can view objects that are locked or checked out by all users. You can select locked objects and unlock them so that another user can edit them. You can select checked out objects and undo the checked-out state, or assign the checked-out state to another user.

You can perform the following operations:

List checked-out objects.

You can list the objects that are checked out from the Model repository. You can filter the list by the time that a user checked out the object. You might want to do this to identify the developers working on each object.

Check in an object.

You can check in any object that is checked out from the Model repository.

Undo the checkout of a checked-out object.

When a developer has checked out an object from the Model repository and is unavailable to check it in, you can undo the checkout. When you undo the checkout of an object that a user edited, the changes are lost.

Note: If a user moved an object while it was checked out and you undo the checkout, the object remains in its current location, and its version history restarts. Undoing the checkout does not restore it to its pre-checkout location.

Reassign the ownership of checked-out objects.

You can reassign ownership of a checked-out object from one user to another. You might want to do this if a team member goes on vacation with objects still checked out.

If the owner of a checked-out object saved changes, the changes are retained when you reassign the object. If the changes are not saved, the changes are lost when you reassign the object.

Versioned Object Administration Example

You are the Model repository administrator for a development team. One of the team members, abcar, begins an extended, unexpected absence. The user had objects checked out when the absence began.

To assign the checked-out objects to other team members, complete the following steps:

1. Filter the list of checked out objects to list all the objects that abcar has checked out.
2. Select some objects and undo the checkout.
The objects are checked in to the Model repository, and any changes that abcar made are lost.
3. Select the remainder of the objects and reassign them to user zovar.
Any changes that abcar made are retained. User zovar can continue development on the objects, or check in the objects without additional changes. User zovar can also choose to undo the check-out of the objects and lose any changes that abcar made.

Troubleshooting Team-based Development

Consider the following troubleshooting tips when you use features related to team-based development:

The Perforce version control system fails to check in some objects, with an error about excessively long object path names.

Due to Windows OS limitations on the number of characters in a file path, Model repository objects with long path and file names fail when you try to check them in. The Perforce error message reads "Submit aborted" and says the file path exceeds the internal length limit.

To work around this problem, limit the length of directory names in the path to the Perforce depot, and limit the length of project, folder, and object names in the Model repository. Shorter names in all instances help limit the total number of characters in the object path name.

Alternatively, you can install Informatica or the Perforce instance on non-Windows hosts that do not have this limitation.

Creating a Model Repository Service

1. Create a database for the Model repository.
2. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
3. On the Domain Actions menu, click **New** > **Model Repository Service**.
4. In the properties view, enter the general properties for the Model Repository Service.

5. Click **Next**.
6. Enter the database properties for the Model Repository Service.
7. Click **Test Connection** to test the connection to the database.
8. Select one of the following options:
 - Do Not Create New Content. Select this option if the specified database contains existing content for the Model repository. This is the default.
 - Create New Content. Select this option to create content for the Model repository in the specified database.
9. Click **Finish**.
10. If you created the Model Repository Service to use existing content, select the Model Repository Service in the Navigator, and then click **Actions > Search Index > Re-Index** on the **Manage** tab **Actions** menu.

CHAPTER 10

PowerCenter Integration Service

This chapter includes the following topics:

- [PowerCenter Integration Service Overview, 220](#)
- [Creating a PowerCenter Integration Service, 221](#)
- [Enabling and Disabling PowerCenter Integration Services and Processes, 223](#)
- [Operating Mode, 224](#)
- [PowerCenter Integration Service Properties, 228](#)
- [Operating System Profiles for the PowerCenter Integration Service, 238](#)
- [Associated Repository for the PowerCenter Integration Service, 239](#)
- [PowerCenter Integration Service Processes, 240](#)
- [Configuration for the PowerCenter Integration Service Grid, 246](#)
- [Load Balancer for the PowerCenter Integration Service , 251](#)

PowerCenter Integration Service Overview

The PowerCenter Integration Service is an application service that runs sessions and workflows. Use the Administrator tool to manage the PowerCenter Integration Service.

You can use the Administrator tool to complete the following configuration tasks for the PowerCenter Integration Service:

- Create a PowerCenter Integration Service. Create a PowerCenter Integration Service to replace an existing PowerCenter Integration Service or to use multiple PowerCenter Integration Services.
- Enable or disable the PowerCenter Integration Service. Enable the PowerCenter Integration Service to run sessions and workflows. You might disable the PowerCenter Integration Service to prevent users from running sessions and workflows while performing maintenance on the machine or modifying the repository.
- Configure normal or safe mode. Configure the PowerCenter Integration Service to run in normal or safe mode.
- Configure the PowerCenter Integration Service properties. Configure the PowerCenter Integration Service properties to change behavior of the PowerCenter Integration Service.
- Configure the associated repository. You must associate a repository with a PowerCenter Integration Service. The PowerCenter Integration Service uses the mappings in the repository to run sessions and workflows.

- Configure the PowerCenter Integration Service processes. Configure service process properties for each node, such as the code page and service process variables.
- Configure permissions on the PowerCenter Integration Service.
- Remove a PowerCenter Integration Service. You may need to remove a PowerCenter Integration Service if it becomes obsolete.

Based on your license, the PowerCenter Integration Service can be highly available.

Creating a PowerCenter Integration Service

You can create a PowerCenter Integration Service when you configure Informatica application services. You may need to create an additional PowerCenter Integration Service to replace an existing one or create multiple PowerCenter Integration Services.

You must assign a PowerCenter repository to the PowerCenter Integration Service. You can assign the repository when you create the PowerCenter Integration Service or after you create the PowerCenter Integration Service. You must assign a repository before you can run the PowerCenter Integration Service. The repository that you assign to the PowerCenter Integration Service is called the *associated repository*. The PowerCenter Integration Service retrieves metadata, such as workflows and mappings, from the associated repository.

After you create a PowerCenter Integration Service, you must assign a code page for each PowerCenter Integration Service process. The code page for each PowerCenter Integration Service process must be a subset of the code page of the associated repository. You must select the associated repository before you can select the code page for a PowerCenter Integration Service process. The PowerCenter Repository Service must be enabled to set up a code page for a PowerCenter Integration Service process.

Note: If you configure a PowerCenter Integration Service to run on a node that is unavailable, you must start the node and configure \$PMRootDir for the service process before you run workflows with the PowerCenter Integration Service.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. On the Domain Navigator Actions menu, click New > PowerCenter Integration Service.
The New Integration Service dialog box appears.
3. Enter values for the following PowerCenter Integration Service options.

The following table describes the PowerCenter Integration Service options:

Property	Description
Name	Name of the PowerCenter Integration Service. The characters must be compatible with the code page of the associated repository. The name is not case sensitive and must be unique within the domain. It cannot exceed 128 characters or begin with @. It also cannot contain spaces or the following special characters: ` ~ % ^ * + = { } \ ; ' " / ? . , < > ! () []
Description	Description of the PowerCenter Integration Service. The description cannot exceed 765 characters.

Property	Description
Location	Domain and folder where the service is created. Click Browse to choose a different folder. You can also move the PowerCenter Integration Service to a different folder after you create it.
License	License to assign to the PowerCenter Integration Service. If you do not select a license now, you can assign a license to the service later. Required if you want to enable the PowerCenter Integration Service. The options allowed in your license determine the properties you must set for the PowerCenter Integration Service.
Node	Node on which the PowerCenter Integration Service runs. Required if you do not select a license or your license does not include the high availability option.
Assign	Indicates whether the PowerCenter Integration Service runs on a grid or nodes.
Grid	Name of the grid on which the PowerCenter Integration Service run. Available if your license includes the high availability option. Required if you assign the PowerCenter Integration Service to run on a grid.
Primary Node	Primary node on which the PowerCenter Integration Service runs. Required if you assign the PowerCenter Integration Service to run on nodes.
Backup Nodes	Nodes used as backup to the primary node. Displays if you configure the PowerCenter Integration Service to run on multiple nodes and you have the high availability option. Click Select to choose the nodes to use for backup.
Associated Repository Service	PowerCenter Repository Service associated with the PowerCenter Integration Service. If you do not select the associated PowerCenter Repository Service now, you can select it later. You must select the PowerCenter Repository Service before you run the PowerCenter Integration Service.
Repository User Name	User name to access the repository.
Repository Password	Password for the user. Required when you select an associated PowerCenter Repository Service.
Security Domain	Security domain for the user. Required when you select an associated PowerCenter Repository Service. To apply changes, restart the PowerCenter Integration Service. The Security Domain field appears when the Informatica domain contains an LDAP security domain.
Data Movement Mode	Mode that determines how the PowerCenter Integration Service handles character data. Choose ASCII or Unicode. ASCII mode passes 7-bit ASCII or EBCDIC character data. Unicode mode passes 8-bit ASCII and multibyte character data from sources to targets. Default is ASCII.

4. Click Finish.

You must specify a PowerCenter Repository Service before you can enable the PowerCenter Integration Service.

You can specify the code page for each PowerCenter Integration Service process node and select the Enable Service option to enable the service. If you do not specify the code page information now, you can specify it later. You cannot enable the PowerCenter Integration Service until you assign the code page for each PowerCenter Integration Service process node.

5. Click OK.

Enabling and Disabling PowerCenter Integration Services and Processes

You can enable and disable a PowerCenter Integration Service process or the entire PowerCenter Integration Service. If you run the PowerCenter Integration Service on a grid or with the high availability option, you have one PowerCenter Integration Service process configured for each node. For a grid, the PowerCenter Integration Service runs all enabled PowerCenter Integration Service processes. With high availability, the PowerCenter Integration Service runs the PowerCenter Integration Service process on the primary node.

Enabling or Disabling a PowerCenter Integration Service Process

Use the Administrator tool to enable and disable a PowerCenter Integration Service process. Each service process runs on one node. You must enable the PowerCenter Integration Service process if you want the node to perform PowerCenter Integration Service tasks. You may want to disable the service process on a node to perform maintenance on that node or to enable safe mode for the PowerCenter Integration Service.

To enable or disable a PowerCenter Integration Service process:

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the PowerCenter Integration Service.
3. In the contents panel, click the **Processes** view.
4. Select a process.
5. To disable a process, click **Actions > Disable Process**.
The Disable Process dialog box displays.
6. Choose a disable mode, and then click **OK**.
7. To enable a process, click **Actions > Enable Process**.

Enabling or Disabling the PowerCenter Integration Service

Use the Administrator tool to enable and disable a PowerCenter Integration Service. You may want to disable a PowerCenter Integration Service if you need to perform maintenance or if you want temporarily restrict users from using the service. You can enable a disabled PowerCenter Integration Service to make it available again.

When you disable the PowerCenter Integration Service, you shut down the PowerCenter Integration Service and disable all service processes for the PowerCenter Integration Service. If you are running a PowerCenter Integration Service on a grid, you disable all service processes on the grid.

When you disable the PowerCenter Integration Service, you must choose what to do if a process or workflow is running. You must choose one of the following options:

- **Complete.** Allows the sessions and workflows to run to completion before shutting down the service.

- **Stop.** Stops all sessions and workflows and then shuts down the service.
- **Abort.** Tries to stop all sessions and workflows before aborting them and shutting down the service.

When you enable the PowerCenter Integration Service, the service starts. The associated PowerCenter Repository Service must be started before you can enable the PowerCenter Integration Service. If you enable a PowerCenter Integration Service when the associated PowerCenter Repository Service is not running, the following error appears:

```
The Service Manager could not start the service due to the following error: [DOM_10076]
Unable to enable service [<Integration Service>] because of dependent services
[<PowerCenter Repository Service>] are not initialized.
```

If the PowerCenter Integration Service is unable to start, the Service Manager keeps trying to start the service until it reaches the maximum restart attempts defined in the domain properties. For example, if you try to start the PowerCenter Integration Service without specifying the code page for each PowerCenter Integration Service process, the domain tries to start the service. The service does not start without specifying a valid code page for each PowerCenter Integration Service process. The domain keeps trying to start the service until it reaches the maximum number of attempts.

If the service fails to start, review the logs for this PowerCenter Integration Service to determine the reason for failure and fix the problem. After you fix the problem, you must disable and re-enable the PowerCenter Integration Service to start it.

To enable or disable a PowerCenter Integration Service:

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the PowerCenter Integration Service.
3. On the **Manage** tab **Actions** menu, select **Disable Service** to disable the service or select **Enable Service** to enable the service.
4. To disable and immediately enable the PowerCenter Integration Service, select **Recycle**.

Operating Mode

You can run the PowerCenter Integration Service in normal or safe operating mode. Normal mode provides full access to users with permissions and privileges to use a PowerCenter Integration Service. Safe mode limits user access to the PowerCenter Integration Service and workflow activity during environment migration or PowerCenter Integration Service maintenance activities.

Run the PowerCenter Integration Service in normal mode during daily operations. In normal mode, users with workflow privileges can run workflows and get session and workflow information for workflows assigned to the PowerCenter Integration Service.

You can configure the PowerCenter Integration Service to run in safe mode or to fail over in safe mode. When you enable the PowerCenter Integration Service to run in safe mode or when the PowerCenter Integration Service fails over in safe mode, it limits access and workflow activity to allow administrators to perform migration or maintenance activities.

Run the PowerCenter Integration Service in safe mode to control which workflows a PowerCenter Integration Service runs and which users can run workflows during migration and maintenance activities. Run in safe mode to verify a production environment, manage workflow schedules, or maintain a PowerCenter Integration Service. In safe mode, users that have the Administrator role for the associated PowerCenter Repository Service can run workflows and get information about sessions and workflows assigned to the PowerCenter Integration Service.

Normal Mode

When you enable a PowerCenter Integration Service to run in normal mode, the PowerCenter Integration Service begins running scheduled workflows. It also completes workflow failover for any workflows that failed while in safe mode, recovers client requests, and recovers any workflows configured for automatic recovery that failed in safe mode.

Users with workflow privileges can run workflows and get session and workflow information for workflows assigned to the PowerCenter Integration Service.

When you change the operating mode from safe to normal, the PowerCenter Integration Service begins running scheduled workflows and completes workflow failover and workflow recovery for any workflows configured for automatic recovery. You can use the Administrator tool to view the log events about the scheduled workflows that started, the workflows that failed over, and the workflows recovered by the PowerCenter Integration Service.

Safe Mode

In safe mode, access to the PowerCenter Integration Service is limited. You can configure the PowerCenter Integration Service to run in safe mode or to fail over in safe mode:

- **Enable in safe mode.** Enable the PowerCenter Integration Service in safe mode to perform migration or maintenance activities. When you enable the PowerCenter Integration Service in safe mode, you limit access to the PowerCenter Integration Service.

When you enable a PowerCenter Integration Service in safe mode, you can choose to have the PowerCenter Integration Service complete, abort, or stop running workflows. In addition, the operating mode on failover also changes to safe.

- **Fail over in safe mode.** Configure the PowerCenter Integration Service process to fail over in safe mode during migration or maintenance activities. When the PowerCenter Integration Service process fails over to a backup node, it restarts in safe mode and limits workflow activity and access to the PowerCenter Integration Service. The PowerCenter Integration Service restores the state of operations for any workflows that were running when the service process failed over, but does not fail over or automatically recover the workflows. You can manually recover the workflow.

After the PowerCenter Integration Service fails over in safe mode during normal operations, you can correct the error that caused the PowerCenter Integration Service process to fail over and restart the service in normal mode.

The behavior of the PowerCenter Integration Service when it fails over in safe mode is the same as when you enable the PowerCenter Integration Service in safe mode. All scheduled workflows, including workflows scheduled to run continuously or start on service initialization, do not run. The PowerCenter Integration Service does not fail over schedules or workflows, does not automatically recover workflows, and does not recover client requests.

Running the PowerCenter Integration Service in Safe Mode

This section describes the specific migration and maintenance activities that you can complete in the PowerCenter Workflow Manager and PowerCenter Workflow Monitor, the behavior of the PowerCenter Integration Service in safe mode, and the privileges required to run and monitor workflows in safe mode.

Performing Migration or Maintenance

You might want to run a PowerCenter Integration Service in safe mode for the following reasons:

- Test a development environment. Run the PowerCenter Integration Service in safe mode to test a development environment before migrating to production. You can run workflows that contain session and command tasks to test the environment. Run the PowerCenter Integration Service in safe mode to limit access to the PowerCenter Integration Service when you run the test sessions and command tasks.
- Manage workflow schedules. During migration, you can unschedule workflows that only run in a development environment. You can enable the PowerCenter Integration Service in safe mode, unschedule the workflow, and then enable the PowerCenter Integration Service in normal mode. After you enable the service in normal mode, the workflows that you unscheduled do not run.
- Troubleshoot the PowerCenter Integration Service. Configure the PowerCenter Integration Service to fail over in safe mode and troubleshoot errors when you migrate or test a production environment configured for high availability. After the PowerCenter Integration Service fails over in safe mode, you can correct the error that caused the PowerCenter Integration Service to fail over.
- Perform maintenance on the PowerCenter Integration Service. When you perform maintenance on a PowerCenter Integration Service, you can limit the users who can run workflows. You can enable the PowerCenter Integration Service in safe mode, change PowerCenter Integration Service properties, and verify the PowerCenter Integration Service functionality before allowing other users to run workflows. For example, you can use safe mode to test changes to the paths for PowerCenter Integration Service files for PowerCenter Integration Service processes.

Workflow Tasks

The following table describes the tasks that users with the Administrator role can perform when the PowerCenter Integration Service runs in safe mode:

Task	Task Description
Run workflows.	Start, stop, abort, and recover workflows. The workflows may contain session or command tasks required to test a development or production environment.
Unschedule workflows.	Unschedule workflows in the PowerCenter Workflow Manager.
Monitor PowerCenter Integration Service properties.	Connect to the PowerCenter Integration Service in the PowerCenter Workflow Monitor. Get PowerCenter Integration Service details and monitor information.
Monitor workflow and task details.	Connect to the PowerCenter Integration Service in the PowerCenter Workflow Monitor and get task, session, and workflow details.
Recover workflows.	Manually recover failed workflows.

PowerCenter Integration Service Behavior

Safe mode affects PowerCenter Integration Service behavior for the following workflow and high availability functionality:

- Workflow schedules. Scheduled workflows remain scheduled, but they do not run if the PowerCenter Integration Service is running in safe mode. This includes workflows scheduled to run continuously and run on service initialization.

Workflow schedules do not fail over when a PowerCenter Integration Service fails over in safe mode. For example, you configure a PowerCenter Integration Service to fail over in safe mode. The PowerCenter Integration Service process fails for a workflow scheduled to run five times, and it fails over after it runs the workflow three times. The PowerCenter Integration Service does not complete the remaining workflows when it fails over to the backup node. The PowerCenter Integration Service completes the workflows when you enable the PowerCenter Integration Service in safe mode.

- Workflow failover. When a PowerCenter Integration Service process fails over in safe mode, workflows do not fail over. The PowerCenter Integration Service restores the state of operations for the workflow. When you enable the PowerCenter Integration Service in normal mode, the PowerCenter Integration Service fails over the workflow and recovers it based on the recovery strategy for the workflow.
- Workflow recovery. The PowerCenter Integration Service does not recover workflows when it runs in safe mode or when the operating mode changes from normal to safe.

The PowerCenter Integration Service recovers a workflow that failed over in safe mode when you change the operating mode from safe to normal, depending on the recovery strategy for the workflow. For example, you configure a workflow for automatic recovery and you configure the PowerCenter Integration Service to fail over in safe mode. If the PowerCenter Integration Service process fails over, the workflow is not recovered while the PowerCenter Integration Service runs in safe mode. When you enable the PowerCenter Integration Service in normal mode, the workflow fails over and the PowerCenter Integration Service recovers it.

You can manually recover the workflow if the workflow fails over in safe mode. You can recover the workflow after the resilience timeout for the PowerCenter Integration Service expires.

- Client request recovery. The PowerCenter Integration Service does not recover client requests when it fails over in safe mode. For example, you stop a workflow and the PowerCenter Integration Service process fails over before the workflow stops. The PowerCenter Integration Service process does not recover your request to stop the workflow when the workflow fails over.

When you enable the PowerCenter Integration Service in normal mode, it recovers the client requests.

Configuring the PowerCenter Integration Service Operating Mode

You can use the Administrator tool to configure the PowerCenter Integration Service to run in safe mode, run in normal mode, or run in safe or normal mode on failover. To configure the operating mode on failover, you must have the high availability option.

Note: When you change the operating mode on fail over from safe to normal, the change takes effect immediately.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select a PowerCenter Integration Service.
3. Click the Properties view.
4. Go to the Operating Mode Configuration section and click Edit.
5. To run the PowerCenter Integration Service in normal mode, set OperatingMode to Normal.
To run the service in safe mode, set OperatingMode to Safe.

6. To run the service in normal mode on failover, set `OperatingModeOnFailover` to `Normal`.
To run the service in safe mode on failover, set `OperatingModeOnFailover` to `Safe`.
7. Click OK.
8. Restart the PowerCenter Integration Service.

The PowerCenter Integration Service starts in the selected mode. The service status at the top of the content pane indicates when the service has restarted.

PowerCenter Integration Service Properties

You can configure general properties, PowerCenter Integration Services properties, custom properties, and more for the PowerCenter Integration Service.

Use the Administrator tool to configure the following PowerCenter Integration Service properties:

- General properties. Assign a license and configure the PowerCenter Integration Service to run on a grid or nodes.
- PowerCenter Integration Service properties. Set the values for the PowerCenter Integration Service variables.
- Advanced properties. Configure advanced properties that determine security and control the behavior of sessions and logs
- Operating mode configuration. Set the PowerCenter Integration Service to start in normal or safe mode and to fail over in normal or safe mode.
- Compatibility and database properties. Configure the source and target database properties, such the maximum number of connections, and configure properties to enable compatibility with previous versions of PowerCenter.
- Configuration properties. Configure the configuration properties, such as the data display format.
- HTTP proxy properties. Configure the connection to the HTTP proxy server.
- Custom properties. Configure custom properties that are unique to specific environments.

To view the properties, select the PowerCenter Integration Service in the Navigator and click Properties view. To modify the properties, edit the section for the property you want to modify.

General Properties

The amount of system resources that the PowerCenter Integration Services uses depends on how you set up the PowerCenter Integration Service. You can configure a PowerCenter Integration Service to run on a grid or on nodes. You can view the system resource usage of the PowerCenter Integration Service using the PowerCenter Workflow Monitor.

When you use a grid, the PowerCenter Integration Service distributes workflow tasks and session threads across multiple nodes. You can increase performance when you run sessions and workflows on a grid. If you choose to run the PowerCenter Integration Service on a grid, select the grid. You must have the server grid option to run the PowerCenter Integration Service on a grid. You must create the grid before you can select the grid.

If you configure the PowerCenter Integration Service to run on nodes, choose one or more PowerCenter Integration Service process nodes. If you have only one node and it becomes unavailable, the domain cannot

accept service requests. With the high availability option, you can run the PowerCenter Integration Service on multiple nodes. To run the service on multiple nodes, choose the primary and backup nodes.

To edit the general properties, select the PowerCenter Integration Service in the Navigator, and then click the Properties view. Edit the section General Properties section. To apply changes, restart the PowerCenter Integration Service.

The following table describes the general properties for the service:

Property	Description
Name	Name of the service. The name is not case sensitive and must be unique within the domain. It cannot exceed 128 characters or begin with @. It also cannot contain spaces or the following special characters: ` ~ % ^ * + = { } \ ; : ' " / ? . , < > ! () [] You cannot change the name of the service after you create it.
Description	Description of the service. The description cannot exceed 765 characters.
License	License object that allows use of the service.
Assign	Indicates whether the PowerCenter Integration Service runs on a grid or on nodes.
Grid	Name of the grid on which the PowerCenter Integration Service runs. Required if you run the PowerCenter Integration Service on a grid.
Primary Node	Primary node on which the PowerCenter Integration Service runs. Required if you run the PowerCenter Integration Service on nodes and you specify at least one backup node. You can select any node in the domain.
Backup Node	Backup node on which the PowerCenter Integration Service can run on. If the primary node becomes unavailable, the PowerCenter Integration Service runs on a backup node. You can select multiple nodes as backup nodes. Available if you have the high availability option and you run the PowerCenter Integration Service on nodes.

PowerCenter Integration Service Properties

You can set the values for the service variables at the service level. You can override some of the PowerCenter Integration Service variables at the session level or workflow level. To override the properties, configure the properties for the session or workflow.

To edit the service properties, select the PowerCenter Integration Service in the Navigator, and then click the Properties view. Edit the PowerCenter Integration Service Properties section.

The following table describes the service properties:

Property	Description
DataMovementMode	<p>Mode that determines how the PowerCenter Integration Service handles character data.</p> <p>In ASCII mode, the PowerCenter Integration Service recognizes 7-bit ASCII and EBCDIC characters and stores each character in a single byte. Use ASCII mode when all sources and targets are 7-bit ASCII or EBCDIC character sets.</p> <p>In Unicode mode, the PowerCenter Integration Service recognizes multibyte character sets as defined by supported code pages. Use Unicode mode when sources or targets use 8-bit or multibyte character sets and contain character data. Default is ASCII.</p> <p>To apply changes, restart the PowerCenter Integration Service.</p>
\$PMSuccessEmailUser	<p>Service variable that specifies the email address of the user to receive email messages when a session completes successfully. Use this variable for the Email User Name attribute for success email. If multiple email addresses are associated with a single user, messages are sent to all of the addresses.</p> <p>If the Integration Service runs on UNIX, you can enter multiple email addresses separated by a comma. If the Integration Service runs on Windows, you can enter multiple email addresses separated by a semicolon or use a distribution list. The PowerCenter Integration Service does not expand this variable when you use it for any other email type.</p>
\$PMFailureEmailUser	<p>Service variable that specifies the email address of the user to receive email messages when a session fails to complete. Use this variable for the Email User Name attribute for failure email. If multiple email addresses are associated with a single user, messages are sent to all of the addresses.</p> <p>If the Integration Service runs on UNIX, you can enter multiple email addresses separated by a comma. If the Integration Service runs on Windows, you can enter multiple email addresses separated by a semicolon or use a distribution list. The PowerCenter Integration Service does not expand this variable when you use it for any other email type.</p>
\$PMSessionLogCount	<p>Service variable that specifies the number of session logs the PowerCenter Integration Service archives for the session.</p> <p>Minimum value is 0. Default is 0.</p>
\$PMWorkflowLogCount	<p>Service variable that specifies the number of workflow logs the PowerCenter Integration Service archives for the workflow.</p> <p>Minimum value is 0. Default is 0.</p>
\$PMSessionErrorThreshold	<p>Service variable that specifies the number of non-fatal errors the PowerCenter Integration Service allows before failing the session. Non-fatal errors include reader, writer, and DTM errors. If you want to stop the session on errors, enter the number of non-fatal errors you want to allow before stopping the session. The PowerCenter Integration Service maintains an independent error count for each source, target, and transformation. Use to configure the Stop On option in the session properties.</p> <p>Defaults to 0. If you use the default setting 0, non-fatal errors do not cause the session to stop.</p>

Advanced Properties

You can configure the properties that control the behavior of PowerCenter Integration Service security, sessions, and logs. To edit the advanced properties, select the PowerCenter Integration Service in the Navigator, and then click the Properties view. Edit the Advanced Properties section.

The following table describes the advanced properties:

Property	Description
Error Severity Level	Level of error logging for the domain. These messages are written to the Log Manager and log files. Specify one of the following message levels: <ul style="list-style-type: none">- Error. Writes ERROR code messages to the log.- Warning. Writes WARNING and ERROR code messages to the log.- Information. Writes INFO, WARNING, and ERROR code messages to the log.- Tracing. Writes TRACE, INFO, WARNING, and ERROR code messages to the log.- Debug. Writes DEBUG, TRACE, INFO, WARNING, and ERROR code messages to the log. Default is INFO.
Resilience Timeout	Number of seconds that the service tries to establish or reestablish a connection to another service. If blank, the value is derived from the domain-level settings. Valid values are between 0 and 2,592,000, inclusive. Default is 180 seconds.
Limit on Resilience Timeouts	Number of seconds that the service holds on to resources for resilience purposes. This property places a restriction on clients that connect to the service. Any resilience timeouts that exceed the limit are cut off at the limit. If blank, the value is derived from the domain-level settings. Valid values are between 0 and 2,592,000, inclusive. Default is 180 seconds.
Timestamp Workflow Log Messages	Appends a timestamp to messages that are written to the workflow log. Default is No.
Allow Debugging	Allows you to run debugger sessions from the Designer. Default is Yes.
LogsInUTF8	Writes to all logs using the UTF-8 character set. Disable this option to write to the logs using the PowerCenter Integration Service code page. This option is available when you configure the PowerCenter Integration Service to run in Unicode mode. When running in Unicode data movement mode, default is Yes. When running in ASCII data movement mode, default is No.
Use Operating System Profiles	Enables the use of operating system profiles. You can select this option if the PowerCenter Integration Service runs on UNIX. To apply changes, restart the PowerCenter Integration Service.
TrustStore	Enter the value for TrustStore using the following syntax: <code><path>/<filename></code> For example: <code>./Certs/trust.keystore</code>
ClientStore	Enter the value for ClientStore using the following syntax: <code><path>/<filename></code> For example: <code>./Certs/client.keystore</code>

Property	Description
JCEProvider	Enter the JCEProvider class name to support NTLM authentication. For example: com.unix.crypto.provider.UnixJCE.
IgnoreResourceRequirements	<p>Ignores task resource requirements when distributing tasks across the nodes of a grid. Used when the PowerCenter Integration Service runs on a grid. Ignored when the PowerCenter Integration Service runs on a node.</p> <p>Enable this option to cause the Load Balancer to ignore task resource requirements. It distributes tasks to available nodes whether or not the nodes have the resources required to run the tasks.</p> <p>Disable this option to cause the Load Balancer to match task resource requirements with node resource availability when distributing tasks. It distributes tasks to nodes that have the required resources.</p> <p>Default is Yes.</p>
Run sessions impacted by dependency updates	Runs sessions that are impacted by dependency updates. By default, the PowerCenter Integration Service does not run impacted sessions. When you modify a dependent object, the parent object can become invalid. The PowerCenter client marks a session with a warning if the session is impacted. At run time, the PowerCenter Integration Service fails the session if it detects errors.
Persist Run-time Statistics to Repository	<p>Level of run-time information stored in the repository. Specify one of the following levels:</p> <ul style="list-style-type: none"> - None. PowerCenter Integration Service does not store any session or workflow run-time information in the repository. - Normal. PowerCenter Integration Service stores workflow details, task details, session statistics, and source and target statistics in the repository. Default is Normal. - Verbose. PowerCenter Integration Service stores workflow details, task details, session statistics, source and target statistics, partition details, and performance details in the repository. <p>To store session performance details in the repository, you must also configure the session to collect performance details and write them to the repository.</p> <p>The PowerCenter Workflow Monitor shows run-time statistics stored in the repository.</p>

Property	Description
Flush Session Recovery Data	<p>Flushes session recovery data for the recovery file from the operating system buffer to the disk. For real-time sessions, the PowerCenter Integration Service flushes the recovery data after each flush latency interval. For all other sessions, the PowerCenter Integration Service flushes the recovery data after each commit interval or user-defined commit. Use this property to prevent data loss if the PowerCenter Integration Service is not able to write recovery data for the recovery file to the disk.</p> <p>Specify one of the following levels:</p> <ul style="list-style-type: none"> - Auto. PowerCenter Integration Service flushes recovery data for all real-time sessions with a JMS or WebSphere MQ source and a non-relational target. - Yes. PowerCenter Integration Service flushes recovery data for all sessions. - No. PowerCenter Integration Service does not flush recovery data. Select this option if you have highly available external systems or if you need to optimize performance. <p>Required if you enable session recovery.</p> <p>Default is Auto.</p> <p>Note: If you select Yes or Auto, you might impact performance.</p>
Store High Availability Persistence in Database	<p>Enables the PowerCenter Integration Service to store process state information in the high availability persistence tables in the PowerCenter repository database.</p> <p>The process state information contains information about which node was running the master PowerCenter Integration Service and which node was running the sessions.</p> <p>Default is no.</p> <p>Note: This property does not determine where the service stores the state of operation files used for recovery. The PowerCenter Integration Service always stores the state of each workflow and session operation in files in the \$PMStorageDir directory of the PowerCenter Integration Service process.</p>

Operating Mode Configuration

The operating mode determines how much user access and workflow activity the PowerCenter Integration Service allows when runs. You can set the service to run in normal mode to allow users full access or in safe mode to limit access. You can also set how the services operates when it fails over to another node.

The following table describes the operating mode properties:

Property	Description
OperatingMode	Mode in which the PowerCenter Integration Service runs.
OperatingModeOnFailover	Operating mode of the PowerCenter Integration Service when the service process fails over to another node.

Compatibility and Database Properties

You can configure properties to reinstate previous Informatica behavior or to configure database behavior. To edit the compatibility and database properties, select the PowerCenter Integration Service in the Navigator, and then click the Properties view > Compatibility and Database Properties > Edit.

The following table describes the compatibility and database properties:

Property	Description
PMServer3XCompatibility	<p>Handles Aggregator transformations as it did in version 3.5. The PowerCenter Integration Service treats null values as zeros in aggregate calculations and performs aggregate calculations before flagging records for insert, update, delete, or reject in Update Strategy expressions.</p> <p>Disable this option to treat null values as NULL and perform aggregate calculations based on the Update Strategy transformation.</p> <p>This overrides both <i>Aggregate treat nulls as zero</i> and <i>Aggregate treat rows as insert</i>.</p> <p>Default is No.</p>
JoinerSourceOrder6xCompatibility	<p>Processes master and detail pipelines sequentially as it did in versions prior to 7.0. The PowerCenter Integration Service processes all data from the master pipeline before it processes the detail pipeline. When the target load order group contains multiple Joiner transformations, the PowerCenter Integration Service processes the detail pipelines sequentially.</p> <p>The PowerCenter Integration Service fails sessions when the mapping meets any of the following conditions:</p> <ul style="list-style-type: none"> - The mapping contains a multiple input group transformation, such as the Custom transformation. Multiple input group transformations require the PowerCenter Integration Service to read sources concurrently. - You configure any Joiner transformation with transaction level transformation scope. <p>Disable this option to process the master and detail pipelines concurrently.</p> <p>Default is No.</p>
AggregateTreatNullAsZero	<p>Treats null values as zero in Aggregator transformations.</p> <p>Disable this option to treat null values as NULL in aggregate calculations.</p> <p>Default is No.</p>
AggregateTreatRowAsInsert	<p>When enabled, the PowerCenter Integration Service ignores the update strategy of rows when it performs aggregate calculations. This option ignores sorted input option of the Aggregator transformation. When disabled, the PowerCenter Integration Service uses the update strategy of rows when it performs aggregate calculations.</p> <p>Default is No.</p>
DateHandling40Compatibility	<p>Handles dates as in version 4.0.</p> <p>Disable this option to handle dates as defined in the current version of PowerCenter.</p> <p>Date handling significantly improved in version 4.5. Enable this option to revert to version 4.0 behavior.</p> <p>Default is No.</p>
TreatCHARasCHARonRead	<p>If you have PowerExchange for PeopleSoft, use this option for PeopleSoft sources on Oracle. You cannot, however, use it for PeopleSoft lookup tables on Oracle or PeopleSoft sources on Microsoft SQL Server.</p>

Property	Description
Max Lookup SP DB Connections	<p>Maximum number of connections to a lookup or stored procedure database when you start a session.</p> <p>If the number of connections needed exceeds this value, session threads must share connections. This can result in decreased performance. If blank, the PowerCenter Integration Service allows an unlimited number of connections to the lookup or stored procedure database.</p> <p>If the PowerCenter Integration Service allows an unlimited number of connections, but the database user does not have permission for the number of connections required by the session, the session fails.</p> <p>Minimum value is 0. Default is 0.</p>
Max Sybase Connections	<p>Maximum number of connections to a Sybase ASE database when you start a session. If the number of connections required by the session is greater than this value, the session fails.</p> <p>Minimum value is 100. Maximum value is 2147483647. Default is 100.</p>
Max MSSQL Connections	<p>Maximum number of connections to a Microsoft SQL Server database when you start a session. If the number of connections required by the session is greater than this value, the session fails.</p> <p>Minimum value is 100. Maximum value is 2147483647. Default is 100.</p>
NumOfDeadlockRetries	<p>Number of times the PowerCenter Integration Service retries a target write on a database deadlock.</p> <p>Minimum value is 10. Maximum value is 1,000,000,000.</p> <p>Default is 10.</p>
DeadlockSleep	<p>Number of seconds before the PowerCenter Integration Service retries a target write on database deadlock. If set to 0 seconds, the PowerCenter Integration Service retries the target write immediately.</p> <p>Minimum value is 0. Maximum value is 2147483647. Default is 0.</p>

Configuration Properties

You can configure session and miscellaneous properties, such as whether to enforce code page compatibility.

To edit the configuration properties, select the PowerCenter Integration Service in the Navigator, and then click the Properties view > Configuration Properties > Edit.

The following table describes the configuration properties:

Property	Description
XMLWarnDupRows	<p>Writes duplicate row warnings and duplicate rows for XML targets to the session log.</p> <p>Default is Yes.</p>
CreateIndicatorFiles	<p>Creates indicator files when you run a workflow with a flat file target.</p> <p>Default is No.</p>

Property	Description
OutputMetaDataForFF	Writes column headers to flat file targets. The PowerCenter Integration Service writes the target definition port names to the flat file target in the first line, starting with the # symbol. Default is No.
TreatDBPartitionAsPassThrough	Uses pass-through partitioning for non-DB2 targets when the partition type is Database Partitioning. Enable this option if you specify Database Partitioning for a non-DB2 target. Otherwise, the PowerCenter Integration Service fails the session. Default is No.
ExportSessionLogLibName	Name of an external shared library to handle session event messages. Typically, shared libraries in Windows have a file name extension of .dll. In UNIX, shared libraries have a file name extension of .sl. If you specify a shared library and the PowerCenter Integration Service encounters an error when loading the library or getting addresses to the functions in the shared library, then the session will fail. The library name you specify can be qualified with an absolute path. If you do not provide the path for the shared library, the PowerCenter Integration Service will locate the shared library based on the library path environment variable specific to each platform.
TreatNullInComparisonOperatorsAs	Determines how the PowerCenter Integration Service evaluates null values in comparison operations. Specify one of the following options: <ul style="list-style-type: none"> - Null. The PowerCenter Integration Service evaluates null values as NULL in comparison expressions. If either operand is NULL, the result is NULL. - High. The PowerCenter Integration Service evaluates null values as greater than non-null values in comparison expressions. If both operands are NULL, the PowerCenter Integration Service evaluates them as equal. When you choose High, comparison expressions never result in NULL. - Low. The PowerCenter Integration Service evaluates null values as less than non-null values in comparison expressions. If both operands are NULL, the PowerCenter Integration Service treats them as equal. When you choose Low, comparison expressions never result in NULL. Default is NULL.
WriterWaitTimeOut	In target-based commit mode, the amount of time in seconds the writer remains idle before it issues a commit when the following conditions are true: <ul style="list-style-type: none"> - The PowerCenter Integration Service has written data to the target. - The PowerCenter Integration Service has not issued a commit. The PowerCenter Integration Service may commit to the target before or after the configured commit interval. Minimum value is 60. Maximum value is 2147483647. Default is 60. If you configure the timeout to be 0 or a negative number, the PowerCenter Integration Service defaults to 60 seconds.
MSExchangeProfile	Microsoft Exchange profile used by the Service Start Account to send post-session email. The Service Start Account must be set up as a Domain account to use this feature.

Property	Description
DateDisplayFormat	Date format the PowerCenter Integration Service uses in log entries. The PowerCenter Integration Service validates the date format you enter. If the date display format is invalid, the PowerCenter Integration Service uses the default date display format. Default is DY MON DD HH24:MI:SS YYYY.
ValidateDataCodePages	Enforces data code page compatibility. Disable this option to lift restrictions for source and target data code page selection, stored procedure and lookup database code page selection, and session sort order selection. The PowerCenter Integration Service performs data code page validation in Unicode data movement mode only. Option available if you run the PowerCenter Integration Service in Unicode data movement mode. Option disabled if you run the PowerCenter Integration Service in ASCII data movement mode. Default is Yes.

HTTP Proxy Properties

You can configure properties for the HTTP proxy server for Web Services and the HTTP transformation.

To edit the HTTP proxy properties, select the PowerCenter Integration Service in the Navigator, and click the Properties view > HTTP Proxy Properties > Edit.

The following table describes the HTTP proxy properties:

Property	Description
HttpProxyServer	Name of the HTTP proxy server.
HttpProxyPort	Port number of the HTTP proxy server. This must be a number.
HttpProxyUser	Authenticated user name for the HTTP proxy server. This is required if the proxy server requires authentication.
HttpProxyPassword	Password for the authenticated user. This is required if the proxy server requires authentication.
HttpProxyDomain	Domain for authentication.

Custom Properties for the PowerCenter Integration Service

Configure custom properties that are unique to specific environments.

You might need to apply custom properties in special cases. When you define a custom property, enter the property name and an initial value. Define custom properties only at the request of Informatica Global Customer Support.

Operating System Profiles for the PowerCenter Integration Service

By default, the PowerCenter Integration Service process runs all workflows using the permissions of the operating system user that starts Informatica Services. The PowerCenter Integration Service writes output files to a single shared location specified in the \$PMRootDir service process variable.

When you configure the PowerCenter Integration Service to use operating system profiles, the PowerCenter Integration Service process runs workflows with the permission of the operating system user you define in the operating system profile. The operating system profile contains the operating system user name, service process variables, and environment variables. The operating system user must have access to the directories you configure in the profile and the directories the PowerCenter Integration Service accesses at run time. You can use operating system profiles for a PowerCenter Integration Service that runs on UNIX. When you configure operating system profiles on UNIX, you must enable `setuid` for the file system that contains the Informatica installation.

To use an operating system profile, assign the profile to a repository folder or assign the profile to a workflow when you start a workflow. You must have permission on the operating system profile to assign it to a folder or workflow. For example, you assign operating system profile Sales to workflow A. The user that runs workflow A must also have permissions to use operating system profile Sales. The PowerCenter Integration Service stores the output files for workflow A in a location specified in the \$PMRootDir service process variable that the profile can access.

To manage permissions for operating system profiles, go to the Security page of the Administrator tool.

Operating System Profile Components

Configure the following components in an operating system profile:

- Operating system user name. Configure the operating system user that the PowerCenter Integration Service uses to run workflows.
- Service process variables. Configure service process variables in the operating system profile to specify different output file locations based on the profile assigned to the workflow.
- Environment variables. Configure environment variables that the PowerCenter Integration Services uses at run time.
- Permissions. Configure permissions for users to use operating system profiles.

Configuring Operating System Profiles

To use operating system profiles to run workflows, complete the following steps:

1. On UNIX, verify that `setuid` is enabled on the file system that contains the Informatica installation. If necessary, remount the file system with `setuid` enabled.
2. Enable operating system profiles in the advanced properties section of the PowerCenter Integration Service properties.
Note: You can use the default umask value 0022. Or, set the value to 0027 or 0077 for better security.
3. Configure *pmimpprocess* on every node where the PowerCenter Integration Service runs. *pmimpprocess* is a tool that the DTM process, command tasks, and parameter files use to switch between operating system users.
4. Create the operating system profiles on the Security page of the Administrator tool.

On the Security tab Actions menu, select Configure operating system profiles

5. Assign permissions on operating system profiles to users or groups.
6. You can assign operating system profiles to repository folders or to a workflow.

To configure pmimpprocess:

1. At the command prompt, switch to the following directory:
`<Informatica installation directory>/server/bin`
2. Enter the following information at the command line to log in as root:
`su root`
3. Enter the following command to create a group for the administrator user:
`sudo groupadd <group name>`
4. Enter the following command to add the administrator user to the group:
`sudo usermod -G <group name> <Informatica administrator user>`

The administrator user is the Linux user whose permissions are used for all Informatica services.

5. Enter the following command to change the owner and group of pmimpprocess to root and the group that you created:
`chown root:<group name> pmimpprocess`
6. Set the following permissions:
`chmod 6710 pmimpprocess`

Troubleshooting Operating System Profiles

After I selected Use Operating System Profiles, the PowerCenter Integration Service failed to start.

The PowerCenter Integration Service will not start if operating system profiles is enabled on Windows or a grid that includes a Windows node. You can enable operating system profiles on PowerCenter Integration Services that run on UNIX.

Or, *pmimpprocess* was not configured. To use operating system profiles, you must set the owner and group of *pmimpprocess* to administrator and enable the setuid bit for *pmimpprocess*.

Associated Repository for the PowerCenter Integration Service

When you create the PowerCenter Integration Service, you specify the repository associated with the PowerCenter Integration Service. You may need to change the repository connection information. For example, you need to update the connection information if the repository is moved to another database. You may need to choose a different repository when you move from a development repository to a production repository.

When you update or choose a new repository, you must specify the PowerCenter Repository Service and the user account used to access the repository. The Administrator tool lists the PowerCenter Repository Services defined in the same domain as the PowerCenter Integration Service.

You can edit the associated repository properties in the **Services and Nodes** view on the **Manage** tab. In the Navigator, select the PowerCenter Integration Service. In **Associated Repository Properties**, click **Edit**.

The following table describes the associated repository properties:

Property	Description
Associated Repository Service	PowerCenter Repository Service name to which the PowerCenter Integration Service connects. To apply changes, restart the PowerCenter Integration Service.
Repository User Name	User name to access the repository. To apply changes, restart the PowerCenter Integration Service. Not available for a domain with Kerberos authentication.
Repository Password	Password for the user. To apply changes, restart the PowerCenter Integration Service. Not available for a domain with Kerberos authentication.
Security Domain	Security domain for the user. To apply changes, restart the PowerCenter Integration Service. The Security Domain field appears when the Informatica domain contains an LDAP security domain.

PowerCenter Integration Service Processes

The PowerCenter Integration Service can run each PowerCenter Integration Service process on a different node. When you select the PowerCenter Integration Service in the Administrator tool, you can view the PowerCenter Integration Service process nodes on the Processes tab.

You can change the following properties to configure the way that a PowerCenter Integration Service process runs on a node:

- General properties
- Custom properties
- Environment variables

General properties include the code page and directories for PowerCenter Integration Service files and Java components.

To configure the properties, select the PowerCenter Integration Service in the Administrator tool and click the Processes view. When you select a PowerCenter Integration Service process, the detail panel displays the properties for the service process.

Code Pages

You must specify the code page of each PowerCenter Integration Service process node. The node where the process runs uses the code page when it extracts, transforms, or loads data.

Before you can select a code page for a PowerCenter Integration Service process, you must select an associated repository for the PowerCenter Integration Service. The code page for each PowerCenter Integration Service process node must be a subset of the repository code page. When you edit this property, the field displays code pages that are a subset of the associated PowerCenter Repository Service code page.

When you configure the PowerCenter Integration Service to run on a grid or a backup node, you can use a different code page for each PowerCenter Integration Service process node. However, all codes pages for the PowerCenter Integration Service process nodes must be compatible.

Directories for PowerCenter Integration Service Files

PowerCenter Integration Service files include run-time files, state of operation files, and session log files.

The PowerCenter Integration Service creates files to store the state of operations for the service. The state of operations includes information such as the active service requests, scheduled tasks, and completed and running processes. If the service fails, the PowerCenter Integration Service can restore the state and recover operations from the point of interruption.

The PowerCenter Integration Service process uses run-time files to run workflows and sessions. Run-time files include parameter files, cache files, input files, and output files. If the PowerCenter Integration Service uses operating system profiles, the operating system user specified in the profile must have access to the run-time files.

By default, the installation program creates a set of PowerCenter Integration Service directories in the server \infa_shared directory. You can set the shared location for these directories by configuring the service process variable \$PMRootDir to point to the same location for each PowerCenter Integration Service process. Each PowerCenter Integration Service can use a separate shared location.

Configuring \$PMRootDir

When you configure the PowerCenter Integration Service process variables, you specify the paths for the root directory and its subdirectories. You can specify an absolute directory for the service process variables. Make sure all directories specified for service process variables exist before running a workflow.

Set the root directory in the \$PMRootDir service process variable. The syntax for \$PMRootDir is different for Windows and UNIX:

- On Windows, enter a path beginning with a drive letter, colon, and backslash. For example:

```
C:\Informatica\<infa_version>\server\infa_shared
```

- On UNIX: Enter an absolute path beginning with a slash. For example:

```
/Informatica/<infa_version>/server/infa_shared
```

You can use \$PMRootDir to define subdirectories for other service process variable values. For example, set the \$PMSessionLogDir service process variable to \$PMRootDir/SessLogs.

Configuring Service Process Variables for Multiple Nodes

When you configure the PowerCenter Integration Service to run on a grid or a backup node, all PowerCenter Integration Service processes associated with a PowerCenter Integration Service must use the same shared directories for PowerCenter Integration Service files.

Configure service process variables with identical absolute paths to the shared directories on each node that is configured to run the PowerCenter Integration Service. If you use a mounted drive or a mapped drive, the absolute path to the shared location must also be identical.

For example, if you have a primary and a backup node for the PowerCenter Integration Service, recovery fails when nodes use the following drives for the storage directory:

- Mapped drive on node1: F:\shared\Informatica\<infa_version>\infa_shared\Storage
- Mapped drive on node2: G:\shared\Informatica\<infa_version>\infa_shared\Storage

Recovery also fails when nodes use the following drives for the storage directory:

- Mounted drive on node1: /mnt/shared/Informatica/<infa_version>/infa_shared/Storage
- Mounted drive on node2: /mnt/shared_filesystem/Informatica/<infa_version>/infa_shared/Storage

To use the mapped or mounted drives successfully, both nodes must use the same drive.

Service Process Variables for Operating System Profiles

When you use operating system profiles, define the absolute or relative directory path for \$PMWorkflowLogDir in the PowerCenter Integration Service properties. Define the absolute directory path for \$PMStorageDir in the PowerCenter Integration Service properties and the operating system profile.

The PowerCenter Integration Service writes the workflow log file in the directory specified in \$PMWorkflowLogDir. The PowerCenter Integration Service saves workflow recovery files to the \$PMStorageDir configured in the PowerCenter Integration Service properties and saves the session recovery files to the \$PMStorageDir configured in the operating system profile. Define the other service process variables within each operating system profile.

You can use a relative directory path to define \$PMWorkflowLogDir, but you must use an absolute directory path to define \$PMStorageDir.

Directories for Java Components

You must specify the directory containing the Java components. The PowerCenter Integration Service uses the Java components for the following PowerCenter components:

- Custom transformation that uses Java code
- Java transformation
- PowerExchange for JMS
- PowerExchange for Web Services
- PowerExchange for webMethods

General Properties

The following table describes the general properties:

Property	Description
Codepage	Code page of the PowerCenter Integration Service process node.
\$PMRootDir	Root directory accessible by the node. This is the root directory for other service process variables. It cannot include the following special characters: * ? < > " , Default is <Installation_Directory>\server\infa_shared. The installation directory is based on the service version of the service that you created. When you upgrade the PowerCenter Integration Service, the \$PMRootDir is not updated to the upgraded service version installation directory.
\$PMSessionLogDir	Default directory for session logs. It cannot include the following special characters: * ? < > " , Default is \$PMRootDir/SessLogs.
\$PMBadFileDir	Default directory for reject files. It cannot include the following special characters: * ? < > " , Default is \$PMRootDir/BadFiles.

Property	Description
\$PMCacheDir	<p>Default directory for index and data cache files.</p> <p>You can increase performance when the cache directory is a drive local to the PowerCenter Integration Service process. Do not use a mapped or mounted drive for cache files. It cannot include the following special characters:</p> <p>* ? < > " ,</p> <p>Default is \$PMRootDir/Cache.</p>
\$PMTargetFileDir	<p>Default directory for target files. It cannot include the following special characters:</p> <p>* ? < > " ,</p> <p>Default is \$PMRootDir/TgtFiles.</p>
\$PMSourceFileDir	<p>Default directory for source files. It cannot include the following special characters:</p> <p>* ? < > " ,</p> <p>Default is \$PMRootDir/SrcFiles.</p> <p>Note: If you use Metadata Manager, use the default value. Metadata Manager stores transformed metadata for packaged and universal resources in files in the \$PMRootDir/SrcFiles directory. If you change this property, Metadata Manager cannot retrieve the transformed metadata when you load a packaged or universal resource.</p>
\$PMExtProcDir	<p>Default directory for external procedures. It cannot include the following special characters:</p> <p>* ? < > " ,</p> <p>Default is \$PMRootDir/ExtProc.</p>
\$PMTempDir	<p>Default directory for temporary files. It cannot include the following special characters:</p> <p>* ? < > " ,</p> <p>Default is \$PMRootDir/Temp.</p>
\$PMWorkflowLogDir	<p>Default directory for workflow logs. It cannot include the following special characters:</p> <p>* ? < > " ,</p> <p>Default is \$PMRootDir/WorkflowLogs.</p>
\$PMLookupFileDir	<p>Default directory for lookup files. It cannot include the following special characters:</p> <p>* ? < > " ,</p> <p>Default is \$PMRootDir/LkpFiles.</p>
\$PMStorageDir	<p>Default directory for state of operation files. The PowerCenter Integration Service uses these files for recovery if you have the high availability option or if you enable a workflow for recovery. These files store the state of each workflow and session operation. It cannot include the following special characters:</p> <p>* ? < > " ,</p> <p>Default is \$PMRootDir/Storage.</p>
Java SDK ClassPath	<p>Java SDK classpath. You can set the classpath to any JAR files you need to run a session that require java components. The PowerCenter Integration Service appends the values you set to the system CLASSPATH. For more information, see "Directories for Java Components" on page 242.</p>

Property	Description
Java SDK Minimum Memory	Minimum amount of memory the Java SDK uses during a session. If the session fails due to a lack of memory, you may want to increase this value. Default is 32 MB.
Java SDK Maximum Memory	Maximum amount of memory the Java SDK uses during a session. If the session fails due to a lack of memory, you may want to increase this value. Default is 64 MB.

Custom Properties for the PowerCenter Integration Service Process

Configure custom properties that are unique to specific environments.

You might need to apply custom properties in special cases. When you define a custom property, enter the property name and an initial value. Define custom properties only at the request of Informatica Global Customer Support.

Define the JVMClassPath custom property to enable communication between the Informatica Domain and the cluster. The following table describes the JVMClassPath value for the MapR cluster:

Property	Value
JVMClassPath	<Informatica Installation Directory>/source/services/shared/hadoop/ mapr<version>/*:<Informatica Installation Directory>/source/services/shared/ hadoop/*

Environment Variables

The database client path on a node is controlled by an environment variable.

Set the database client path environment variable for the PowerCenter Integration Service process if the PowerCenter Integration Service process requires a different database client than another PowerCenter Integration Service process that is running on the same node. For example, the service version of each PowerCenter Integration Service running on the node requires a different database client version. You can configure each PowerCenter Integration Service process to use a different value for the database client environment variable.

The database client code page on a node is usually controlled by an environment variable. For example, Oracle uses NLS_LANG, and IBM DB2 uses DB2CODEPAGE. All PowerCenter Integration Services and PowerCenter Repository Services that run on this node use the same environment variable. You can configure a PowerCenter Integration Service process to use a different value for the database client code page environment variable than the value set for the node.

You might want to configure the code page environment variable for a PowerCenter Integration Service process for the following reasons:

- A PowerCenter Integration Service and PowerCenter Repository Service running on the node require different database client code pages. For example, you have a Shift-JIS repository that requires that the code page environment variable be set to Shift-JIS. However, the PowerCenter Integration Service reads from and writes to databases using the UTF-8 code page. The PowerCenter Integration Service requires that the code page environment variable be set to UTF-8.

Set the environment variable on the node to Shift-JIS. Then add the environment variable to the PowerCenter Integration Service process properties and set the value to UTF-8.

- Multiple PowerCenter Integration Services running on the node use different data movement modes. For example, you have one PowerCenter Integration Service running in Unicode mode and another running in ASCII mode on the same node. The PowerCenter Integration Service running in Unicode mode requires that the code page environment variable be set to UTF-8. For optimal performance, the PowerCenter Integration Service running in ASCII mode requires that the code page environment variable be set to 7-bit ASCII.

Set the environment variable on the node to UTF-8. Then add the environment variable to the properties of the PowerCenter Integration Service process running in ASCII mode and set the value to 7-bit ASCII.

If the PowerCenter Integration Service uses operating system profiles, environment variables configured in the operating system profile override the environment variables set in the general properties for the PowerCenter Integration Service process.

Environment Variables for MapR

When the MapR cluster is secured with MapR Kerberos authentication, edit PowerCenter Integration Service properties to enable communication between the Informatica domain and the cluster.

The following table describes properties to define the Kerberos authentication protocol:

Property	Value
JAVA_OPTS	<code>-Dhadoop.login=<MAPR_ECOSYSTEM_LOGIN_OPTS> -Dhttps.protocols=TLSv1.2</code> where <MAPR_ECOSYSTEM_LOGIN_OPTS> is the value of the MAPR_ECOSYSTEM_LOGIN_OPTS property in the file <code>/opt/mapr/conf/env.sh</code> .
MAPR_HOME	Hadoop distribution directory location on the machine that runs the Data Integration Service. For example, <code><Informatica installation directory>/services/shared/hadoop/mapr_5.2.0/lib/*</code>
MAPR_TICKETFILE_LOCATION	Optional. Directory where an additional MapR Ticket file is stored on the machine that runs the Data Integration Service. When the MapR cluster is configured to enable a user to use Kerberos authentication and MapR Ticket authentication, generate a MapR ticketfile for the user for each authentication mode. Save one ticketfile in <code>/tmp</code> . Save the other ticketfile in any directory on the Data Integration Service machine, and provide the location as the value for this property. For example, for a user id 1234, save a MapR ticketfile named like <code>maprticket_1234</code> in <code>/tmp</code> , and save another MapR ticketfile named like <code>maprticket_1234</code> in the MAPR_TICKETFILE_LOCATION. Note: The ticketfiles can have the same or different names. You must generate the MapR ticketfiles separately and save one to the MAPR_TICKETFILE_LOCATION.

Changes take effect when you restart the PowerCenter Integration Service.

Configuration for the PowerCenter Integration Service Grid

A grid is an alias assigned to a group of nodes that run sessions and workflows. When you run a workflow on a grid, you improve scalability and performance by distributing Session and Command tasks to service processes running on nodes in the grid. When you run a session on a grid, you improve scalability and performance by distributing session threads to multiple DTM processes running on nodes in the grid.

To run a workflow or session on a grid, you assign resources to nodes, create and configure the grid, and configure the PowerCenter Integration Service to run on a grid.

To configure a grid, complete the following tasks:

1. Create a grid and assign nodes to the grid.
2. Configure the PowerCenter Integration Service to run on a grid.
3. Configure the PowerCenter Integration Service processes for the nodes in the grid. If the PowerCenter Integration Service uses operating system profiles, all nodes on the grid must run on UNIX.

4. Assign resources to nodes. You assign resources to a node to allow the PowerCenter Integration Service to match the resources required to run a task or session thread with the resources available on a node.

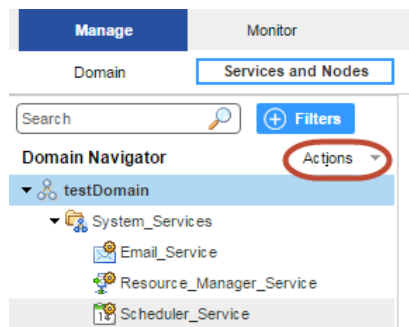
After you configure the grid and PowerCenter Integration Service, you configure a workflow to run on the PowerCenter Integration Service assigned to a grid.

Creating a Grid

To create a grid, create the grid object and assign nodes to the grid. You can assign a node to more than one grid.

When you create a grid for the Data Integration Service, the nodes assigned to the grid must have specific roles depending on the types of jobs that the Data Integration Service runs. For more information, see [“Grid Configuration by Job Type” on page 129](#).

1. In the Administrator tool, click the **Manage** tab.
2. Click the **Services and Nodes** view.
3. In the Domain Navigator, select the domain.



4. On the Navigator Actions menu, click **New > Grid**.
The **Create Grid** dialog box appears.
5. Enter the following properties:

Property	Description
Name	Name of the grid. The name is not case sensitive and must be unique within the domain. It cannot exceed 128 characters or begin with @. It also cannot contain spaces or the following special characters: ` ~ % ^ * + = { } \ ; : ' " / ? . , < > ! () []
Description	Description of the grid. The description cannot exceed 765 characters.
Nodes	Select nodes to assign to the grid.
Path	Location in the Navigator, such as: DomainName/ProductionGrids

6. Click **OK**.

Configuring the PowerCenter Integration Service to Run on a Grid

You configure the PowerCenter Integration Service by assigning the grid to the PowerCenter Integration Service.

To assign the grid to a PowerCenter Integration Service:

1. In the Administrator tool, select the PowerCenter Integration Service Properties tab.
2. Edit the grid and node assignments, and select Grid.
3. Select the grid you want to assign to the PowerCenter Integration Service.

Configuring the PowerCenter Integration Service Processes

When you run a session or a workflow on a grid, a service process runs on each node in the grid. Each service process running on a node must be compatible or configured the same. It must also have access to the directories and input files used by the PowerCenter Integration Service.

To ensure consistent results, complete the following tasks:

- Verify the shared storage location. Verify that the shared storage location is accessible to each node in the grid. If the PowerCenter Integration Service uses operating system profiles, the operating system user must have access to the shared storage location.
- Configure the service process. Configure \$PMRootDir to the shared location on each node in the grid. Configure service process variables with identical absolute paths to the shared directories on each node in the grid. If the PowerCenter Integration Service uses operating system profiles, the service process variables you define in the operating system profile override the service process variable setting for every node. The operating system user must have access to the \$PMRootDir configured in the operating system profile on every node in the grid.

Complete the following process to configure the service processes:

1. Select the PowerCenter Integration Service in the Navigator.
2. Click the Processes tab.
The tab displays the service process for each node assigned to the grid.
3. Configure \$PMRootDir to point to the shared location.
4. Configure the following service process settings for each node in the grid:
 - Code pages. For accurate data movement and transformation, verify that the code pages are compatible for each service process. Use the same code page for each node where possible.
 - Service process variables. Configure the service process variables the same for each service process. For example, the setting for \$PMCacheDir must be identical on each node in the grid.
 - Directories for Java components. Point to the same Java directory to ensure that java components are available to objects that access Java, such as Custom transformations that use Java coding.

Resources

Informatica resources are the database connections, files, directories, node names, and operating system types required by a task. You can configure the PowerCenter Integration Service to check resources. When you do this, the Load Balancer matches the resources available to nodes in the grid with the resources required by the workflow. It dispatches tasks in the workflow to nodes where the required resources are available. If the PowerCenter Integration Service is not configured to run on a grid, the Load Balancer ignores resource requirements.

For example, if a session uses a parameter file, it must run on a node that has access to the file. You create a resource for the parameter file and make it available to one or more nodes. When you configure the session, you assign the parameter file resource as a required resource. The Load Balancer dispatches the Session task to a node that has the parameter file resource. If no node has the parameter file resource available, the session fails.

Resources for a node can be predefined or user-defined. Informatica creates predefined resources during installation. Predefined resources include the connections available on a node, node name, and operating system type. When you create a node, all connection resources are available by default. Disable the connection resources that are not available on the node. For example, if the node does not have Oracle client libraries, disable the Oracle Application connections. If the Load Balancer dispatches a task to a node where the required resources are not available, the task fails. You cannot disable or remove node name or operating system type resources.

User-defined resources include file/directory and custom resources. Use file/directory resources for parameter files or file server directories. Use custom resources for any other resources available to the node, such as database client version.

The following table lists the types of resources you use in Informatica:

Type	Predefined/ User-Defined	Description
Connection	Predefined	Any resource installed with PowerCenter, such as a plug-in or a connection object. A connection object may be a relational, application, FTP, external loader, or queue connection. When you create a node, all connection resources are available by default. Disable the connection resources that are not available to the node. Any Session task that reads from or writes to a relational database requires one or more connection resources. The Workflow Manager assigns connection resources to the session by default.
Node Name	Predefined	A resource for the name of the node. A Session, Command, or predefined Event-Wait task requires a node name resource if it must run on a specific node.
Operating System Type	Predefined	A resource for the type of operating system on the node. A Session or Command task requires an operating system type resource if it must run a specific operating system.
Custom	User-defined	Any resource for all other resources available to the node, such as a specific database client version. For example, a Session task requires a custom resource if it accesses a Custom transformation shared library or if it requires a specific database client version.
File/Directory	User-defined	Any resource for files or directories, such as a parameter file or a file server directory. For example, a Session task requires a file resource if it accesses a session parameter file.

You configure resources required by Session, Command, and predefined Event-Wait tasks in the task properties.

You define resources available to a node on the Resources tab of the node in the Administrator tool.

Note: When you define a resource for a node, you must verify that the resource is available to the node. If the resource is not available and the PowerCenter Integration Service runs a task that requires the resource, the task fails.

You can view the resources available to all nodes in a domain on the Resources view of the domain. The Administrator tool displays a column for each node. It displays a checkmark when a resource is available for a node

Assigning Connection Resources

You can assign the connection resources available to a node in the Administrator tool.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select a node.
3. In the contents panel, click the **Resources** view.
4. Click on a resource that you want to edit.
5. On the **Manage** tab **Actions** menu, click **Enable Selected Resource** or **Disable Selected Resource**.

Defining Custom and File/Directory Resources

You can define custom and file/directory resources available to a node in the Administrator tool. When you define a custom or file/directory resource, you assign a resource name. The resource name is a logical name that you create to identify the resource.

You assign the resource to a PowerCenter task or PowerCenter mapping object instance using this name. To coordinate resource usage, you may want to use a naming convention for file/directory and custom resources.

To define a custom or file/directory resource:

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select a node.
3. In the contents panel, click the **Resources** view.
4. On the **Manage** tab **Actions** menu, click **New Resource**.
5. Enter a name for the resource.

The name is not case sensitive and must be unique within the domain. It cannot exceed 128 characters or begin with @. It also cannot contain spaces or the following special characters: ` ~ % ^ * + = { } \ ; / ? . , < > | ! () []

6. Select a resource type.
7. Click OK.

To remove a custom or file/directory resource, select a resource and click **Delete Selected Resource** on the **Manage** tab **Actions** menu.

Resource Naming Conventions

Using resources with PowerCenter requires coordination and communication between the domain administrator and the workflow developer. The domain administrator defines resources available to nodes. The workflow developer assigns resources required by Session, Command, and predefined Event-Wait tasks. To coordinate resource usage, you can use a naming convention for file/directory and custom resources.

Use the following naming convention:

`resourcetype_description`

For example, multiple nodes in a grid contain a session parameter file called sales1.txt. Create a file resource for it named sessionparamfile_sales1 on each node that contains the file. A workflow developer creates a session that uses the parameter file and assigns the sessionparamfile_sales1 file resource to the session.

When the PowerCenter Integration Service runs the workflow on the grid, the Load Balancer distributes the session assigned the sessionparamfile_sales1 resource to nodes that have the resource defined.

Editing and Deleting a Grid

You can edit or delete a grid from the domain. Edit the grid to change the description, add nodes to the grid, or remove nodes from the grid. You can delete the grid if the grid is no longer required.

Before you remove a node from the grid, disable the PowerCenter Integration Service process running on the node.

Before you delete a grid, disable any PowerCenter Integration Services running on the grid.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. Select the grid in the Domain Navigator.
3. To edit the grid, click **Edit** in the **Grid Details** section.
You can change the grid description, add nodes to the grid, or remove nodes from the grid.
4. To delete the grid, select **Actions** > **Delete**.

Troubleshooting a Grid

I changed the nodes assigned to the grid, but the Integration Service to which the grid is assigned does not show the latest Integration Service processes.

When you change the nodes in a grid, the Service Manager performs the following transactions in the domain configuration database:

1. Updates the grid based on the node changes. For example, if you add a node, the node appears in the grid.
2. Updates the Integration Services to which the grid is assigned. All nodes with the service role in the grid appear as service processes for the Integration Service.

If the Service Manager cannot update an Integration Service and the latest service processes do not appear for the Integration Service, restart the Integration Service. If that does not work, reassign the grid to the Integration Service.

Load Balancer for the PowerCenter Integration Service

The Load Balancer is a component of the PowerCenter Integration Service that dispatches tasks to PowerCenter Integration Service processes running on nodes in a grid. It matches task requirements with resource availability to identify the best PowerCenter Integration Service process to run a task. It can dispatch tasks on a single node or across nodes.

You can configure Load Balancer settings for the domain and for nodes in the domain. The settings you configure for the domain apply to all PowerCenter Integration Services in the domain.

You configure the following settings for the domain to determine how the Load Balancer dispatches tasks:

- **Dispatch mode.** The dispatch mode determines how the Load Balancer dispatches tasks. You can configure the Load Balancer to dispatch tasks in a simple round-robin fashion, in a round-robin fashion using node load metrics, or to the node with the most available computing resources.
- **Service level.** Service levels establish dispatch priority among tasks that are waiting to be dispatched. You can create different service levels that a workflow developer can assign to workflows.

You configure the following Load Balancer settings for each node:

- **Resources.** When the PowerCenter Integration Service runs on a grid, the Load Balancer can compare the resources required by a task with the resources available on each node. The Load Balancer dispatches tasks to nodes that have the required resources. You assign required resources in the task properties. You configure available resources using the Administrator tool or *infacmd*.
- **CPU profile.** In adaptive dispatch mode, the Load Balancer uses the CPU profile to rank the computing throughput of each CPU and bus architecture in a grid. It uses this value to ensure that more powerful nodes get precedence for dispatch.
- **Resource provision thresholds.** The Load Balancer checks one or more resource provision thresholds to determine if it can dispatch a task. The Load Balancer checks different thresholds depending on the dispatch mode.

Configuring the Dispatch Mode

The Load Balancer uses the dispatch mode to select a node to run a task. You configure the dispatch mode for the domain. Therefore, all PowerCenter Integration Services in a domain use the same dispatch mode.

When you change the dispatch mode for a domain, you must restart each PowerCenter Integration Service in the domain. The previous dispatch mode remains in effect until you restart the PowerCenter Integration Service.

You configure the dispatch mode in the domain properties.

The Load Balancer uses the following dispatch modes:

- **Round-robin.** The Load Balancer dispatches tasks to available nodes in a round-robin fashion. It checks the Maximum Processes threshold on each available node and excludes a node if dispatching a task causes the threshold to be exceeded. This mode is the least compute-intensive and is useful when the load on the grid is even and the tasks to dispatch have similar computing requirements.
- **Metric-based.** The Load Balancer evaluates nodes in a round-robin fashion. It checks all resource provision thresholds on each available node and excludes a node if dispatching a task causes the thresholds to be exceeded. The Load Balancer continues to evaluate nodes until it finds a node that can accept the task. This mode prevents overloading nodes when tasks have uneven computing requirements.
- **Adaptive.** The Load Balancer ranks nodes according to current CPU availability. It checks all resource provision thresholds on each available node and excludes a node if dispatching a task causes the thresholds to be exceeded. This mode prevents overloading nodes and ensures the best performance on a grid that is not heavily loaded.

The following table compares the differences among dispatch modes:

Dispatch Mode	Checks resource provision thresholds?	Uses task statistics?	Uses CPU profile?	Allows bypass in dispatch queue?
Round-Robin	Checks maximum processes.	No	No	No
Metric-Based	Checks all thresholds.	Yes	No	No
Adaptive	Checks all thresholds.	Yes	Yes	Yes

Round-Robin Dispatch Mode

In round-robin dispatch mode, the Load Balancer dispatches tasks to nodes in a round-robin fashion. The Load Balancer checks the Maximum Processes resource provision threshold on the first available node. It dispatches the task to this node if dispatching the task does not cause this threshold to be exceeded. If dispatching the task causes this threshold to be exceeded, the Load Balancer evaluates the next node. It continues to evaluate nodes until it finds a node that can accept the task.

The Load Balancer dispatches tasks for execution in the order the Workflow Manager or scheduler submits them. The Load Balancer does not bypass any task in the dispatch queue. Therefore, if a resource-intensive task is first in the dispatch queue, all other tasks with the same service level must wait in the queue until the Load Balancer dispatches the resource-intensive task.

Metric-Based Dispatch Mode

In metric-based dispatch mode, the Load Balancer evaluates nodes in a round-robin fashion until it finds a node that can accept the task. The Load Balancer checks the resource provision thresholds on the first available node. It dispatches the task to this node if dispatching the task causes none of the thresholds to be exceeded. If dispatching the task causes any threshold to be exceeded, or if the node is out of free swap space, the Load Balancer evaluates the next node. It continues to evaluate nodes until it finds a node that can accept the task.

To determine whether a task can run on a particular node, the Load Balancer collects and stores statistics from the last three runs of the task. It compares these statistics with the resource provision thresholds defined for the node. If no statistics exist in the repository, the Load Balancer uses the following default values:

- 40 MB memory
- 15% CPU

The Load Balancer dispatches tasks for execution in the order the Workflow Manager or scheduler submits them. The Load Balancer does not bypass any tasks in the dispatch queue. Therefore, if a resource intensive task is first in the dispatch queue, all other tasks with the same service level must wait in the queue until the Load Balancer dispatches the resource intensive task.

Adaptive Dispatch Mode

In adaptive dispatch mode, the Load Balancer evaluates the computing resources on all available nodes. It identifies the node with the most available CPU and checks the resource provision thresholds on the node. It dispatches the task if doing so does not cause any threshold to be exceeded. The Load Balancer does not dispatch a task to a node that is out of free swap space.

In adaptive dispatch mode, the Load Balancer can use the CPU profile to rank nodes according to the amount of computing resources on the node.

To identify the best node to run a task, the Load Balancer also collects and stores statistics from the last three runs of the task and compares them with node load metrics. If no statistics exist in the repository, the Load Balancer uses the following default values:

- 40 MB memory
- 15% CPU

In adaptive dispatch mode, the order in which the Load Balancer dispatches tasks from the dispatch queue depends on the task requirements and dispatch priority. For example, if multiple tasks with the same service level are waiting in the dispatch queue and adequate computing resources are not available to run a resource intensive task, the Load Balancer reserves a node for the resource intensive task and keeps dispatching less intensive tasks to other nodes.

Service Levels

Service levels establish priorities among tasks that are waiting to be dispatched.

When the Load Balancer has more tasks to dispatch than the PowerCenter Integration Service can run at the time, the Load Balancer places those tasks in the dispatch queue. When multiple tasks are waiting in the dispatch queue, the Load Balancer uses service levels to determine the order in which to dispatch tasks from the queue.

Service levels are domain properties. Therefore, you can use the same service levels for all repositories in a domain. You create and edit service levels in the domain properties or using *infacmd*.

When you create a service level, a workflow developer can assign it to a workflow in the Workflow Manager. All tasks in a workflow have the same service level. The Load Balancer uses service levels to dispatch tasks from the dispatch queue. For example, you create two service levels:

- Service level “Low” has dispatch priority 10 and maximum dispatch wait time 7,200 seconds.
- Service level “High” has dispatch priority 2 and maximum dispatch wait time 1,800 seconds.

When multiple tasks are in the dispatch queue, the Load Balancer dispatches tasks with service level High before tasks with service level Low because service level High has a higher dispatch priority. If a task with service level Low waits in the dispatch queue for two hours, the Load Balancer changes its dispatch priority to the maximum priority so that the task does not remain in the dispatch queue indefinitely.

The Administrator tool provides a default service level named Default with a dispatch priority of 5 and maximum dispatch wait time of 1800 seconds. You can update the default service level, but you cannot delete it.

When you remove a service level, the Workflow Manager does not update tasks that use the service level. If a workflow service level does not exist in the domain, the Load Balancer dispatches the tasks with the default service level.

Creating Service Levels

Create service levels in the Administrator tool.

1. In the Administrator tool, select a domain in the Navigator.
2. Click the **Properties** tab.
3. In the Service Level Management area, click Add.
4. Enter values for the service level properties.
5. Click **OK**.
6. To remove a service level, click the Remove button for the service level you want to remove.

Configuring Resources

When you configure the PowerCenter Integration Service to run on a grid and to check resource requirements, the Load Balancer dispatches tasks to nodes based on the resources available on each node. You configure the PowerCenter Integration Service to check available resources in the PowerCenter Integration Service properties in Informatica Administrator.

You assign resources required by a task in the task properties in the PowerCenter Workflow Manager.

You define the resources available to each node in the Administrator tool. Define the following types of resources:

- **Connection.** Any resource installed with PowerCenter, such as a plug-in or a connection object. When you create a node, all connection resources are available by default. Disable the connection resources that are not available to the node.
- **File/Directory.** A user-defined resource that defines files or directories available to the node, such as parameter files or file server directories.
- **Custom.** A user-defined resource that identifies any other resources available to the node. For example, you may use a custom resource to identify a specific database client version.

Enable and disable available resources on the Resources tab for the node in the Administrator tool or using *infacmd*.

Calculating the CPU Profile

In adaptive dispatch mode, the Load Balancer uses the CPU profile to rank the computing throughput of each CPU and bus architecture in a grid. This ensures that nodes with higher processing power get precedence for dispatch. This value is not used in round-robin or metric-based dispatch modes.

The CPU profile is an index of the processing power of a node compared to a baseline system. The baseline system is a Pentium 2.4 GHz computer running Windows 2000. For example, if a SPARC 480 MHz computer is 0.28 times as fast as the baseline computer, the CPU profile for the SPARC computer should be set to 0.28.

By default, the CPU profile is set to 1.0. To calculate the CPU profile for a node, select the node in the Navigator and click **Actions > Recalculate CPU Profile Benchmark**. To get the most accurate value, calculate the CPU profile when the node is idle. The calculation takes approximately five minutes and uses 100% of one CPU on the machine.

You can also calculate the CPU profile using *infacmd*. Or, you can edit the node properties and update the value manually.

Defining Resource Provision Thresholds

The Load Balancer dispatches tasks to PowerCenter Integration Service processes running on a node. It can continue to dispatch tasks to a node as long as the resource provision thresholds defined for the node are not exceeded. When the Load Balancer has more Session and Command tasks to dispatch than the PowerCenter Integration Service can run at a time, the Load Balancer places the tasks in the dispatch queue. It dispatches tasks from the queue when a PowerCenter Integration Service process becomes available.

You can define the following resource provision thresholds for each node in a domain:

- **Maximum CPU run queue length.** The maximum number of runnable threads waiting for CPU resources on the node. The Load Balancer does not count threads that are waiting on disk or network I/Os. If you set this threshold to 2 on a 4-CPU node that has four threads running and two runnable threads waiting, the Load Balancer does not dispatch new tasks to this node.

This threshold limits context switching overhead. You can set this threshold to a low value to preserve computing resources for other applications. If you want the Load Balancer to ignore this threshold, set it to a high number such as 200. The default value is 10.

The Load Balancer uses this threshold in metric-based and adaptive dispatch modes.

- **Maximum memory %.** The maximum percentage of virtual memory allocated on the node relative to the total physical memory size. If you set this threshold to 120% on a node, and virtual memory usage on the node is above 120%, the Load Balancer does not dispatch new tasks to the node.

The default value for this threshold is 150%. Set this threshold to a value greater than 100% to allow the allocation of virtual memory to exceed the physical memory size when dispatching tasks. If you want the Load Balancer to ignore this threshold, set it to a high number such as 1,000.

The Load Balancer uses this threshold in metric-based and adaptive dispatch modes.

- **Maximum processes.** The maximum number of running processes allowed for each PowerCenter Integration Service process that runs on the node. This threshold specifies the maximum number of running Session or Command tasks allowed for each PowerCenter Integration Service process that runs on the node. For example, if you set this threshold to 10 when two PowerCenter Integration Services are running on the node, the maximum number of Session tasks allowed for the node is 20 and the maximum number of Command tasks allowed for the node is 20. Therefore, the maximum number of processes that can run simultaneously is 40.

The default value for this threshold is 10. Set this threshold to a high number, such as 200, to cause the Load Balancer to ignore it. To prevent the Load Balancer from dispatching tasks to the node, set this threshold to 0.

The Load Balancer uses this threshold in all dispatch modes.

You define resource provision thresholds in the node properties.

CHAPTER 11

PowerCenter Integration Service Architecture

This chapter includes the following topics:

- [PowerCenter Integration Service Architecture Overview, 257](#)
- [PowerCenter Integration Service Connectivity, 258](#)
- [PowerCenter Integration Service Process, 258](#)
- [Load Balancer, 260](#)
- [Data Transformation Manager \(DTM\) Process, 263](#)
- [Processing Threads, 264](#)
- [DTM Processing, 267](#)
- [Grids, 269](#)
- [System Resources, 271](#)
- [Code Pages and Data Movement Modes, 272](#)
- [Output Files and Caches, 273](#)

PowerCenter Integration Service Architecture Overview

The PowerCenter Integration Service moves data from sources to targets based on PowerCenter workflow and mapping metadata stored in a PowerCenter repository. When a workflow starts, the PowerCenter Integration Service retrieves mapping, workflow, and session metadata from the repository. It extracts data from the mapping sources and stores the data in memory while it applies the transformation rules configured in the mapping. The PowerCenter Integration Service loads the transformed data into one or more targets.

To move data from sources to targets, the PowerCenter Integration Service uses the following components:

- **PowerCenter Integration Service process.** The PowerCenter Integration Service starts one or more PowerCenter Integration Service processes to run and monitor workflows. When you run a workflow, the PowerCenter Integration Service process starts and locks the workflow, runs the workflow tasks, and starts the process to run sessions.
- **Load Balancer.** The PowerCenter Integration Service uses the Load Balancer to dispatch tasks. The Load Balancer dispatches tasks to achieve optimal performance. It may dispatch tasks to a single node or across the nodes in a grid.

- Data Transformation Manager (DTM) process. The PowerCenter Integration Service starts a DTM process to run each Session and Command task within a workflow. The DTM process performs session validations, creates threads to initialize the session, read, write, and transform data, and handles pre- and post- session operations.

The PowerCenter Integration Service can achieve high performance using symmetric multi-processing systems. It can start and run multiple tasks concurrently. It can also concurrently process partitions within a single session. When you create multiple partitions within a session, the PowerCenter Integration Service creates multiple database connections to a single source and extracts a separate range of data for each connection. It also transforms and loads the data in parallel.

PowerCenter Integration Service Connectivity

The PowerCenter Integration Service is a repository client. It connects to the PowerCenter Repository Service to retrieve workflow and mapping metadata from the repository database. When the PowerCenter Integration Service process requests a repository connection, the request is routed through the master gateway, which sends back PowerCenter Repository Service information to the PowerCenter Integration Service process. The PowerCenter Integration Service process connects to the PowerCenter Repository Service. The PowerCenter Repository Service connects to the repository and performs repository metadata transactions for the client application.

The PowerCenter Workflow Manager communicates with the PowerCenter Integration Service process over a TCP/IP connection. The PowerCenter Workflow Manager communicates with the PowerCenter Integration Service process each time you schedule or edit a workflow, display workflow details, and request workflow and session logs. Use the connection information defined for the domain to access the PowerCenter Integration Service from the PowerCenter Workflow Manager.

The PowerCenter Integration Service process connects to the source or target database using ODBC or native drivers. The PowerCenter Integration Service process maintains a database connection pool for stored procedures or lookup databases in a workflow. The PowerCenter Integration Service process allows an unlimited number of connections to lookup or stored procedure databases. If a database user does not have permission for the number of connections a session requires, the session fails. You can optionally set a parameter to limit the database connections. For a session, the PowerCenter Integration Service process holds the connection as long as it needs to read data from source tables or write data to target tables.

The following table summarizes the software you need to connect the PowerCenter Integration Service to the platform components, source databases, and target databases:

Note: Both the Windows and UNIX versions of the PowerCenter Integration Service can use ODBC drivers to connect to databases. Use native drivers to improve performance.

PowerCenter Integration Service Process

The PowerCenter Integration Service starts a PowerCenter Integration Service process to run and monitor workflows. The PowerCenter Integration Service process is also known as the `pmserver` process. The PowerCenter Integration Service process accepts requests from the PowerCenter Client and from `pmcmd`. It performs the following tasks:

- Manage workflow scheduling.

- Lock and read the workflow.
- Read the parameter file.
- Create the workflow log.
- Run workflow tasks and evaluates the conditional links connecting tasks.
- Start the DTM process or processes to run the session.
- Write historical run information to the repository.
- Send post-session email in the event of a DTM failure.

Manage PowerCenter Workflow Scheduling

The PowerCenter Integration Service process manages workflow scheduling in the following situations:

- When you start the PowerCenter Integration Service. When you start the PowerCenter Integration Service, it queries the repository for a list of workflows configured to run on it.
- When you save a workflow. When you save a workflow assigned to a PowerCenter Integration Service to the repository, the PowerCenter Integration Service process adds the workflow to or removes the workflow from the schedule queue.

Lock and Read the PowerCenter Workflow

When the PowerCenter Integration Service process starts a workflow, it requests an execute lock on the workflow from the repository. The execute lock allows the PowerCenter Integration Service process to run the workflow and prevents you from starting the workflow again until it completes. If the workflow is already locked, the PowerCenter Integration Service process cannot start the workflow. A workflow may be locked if it is already running.

The PowerCenter Integration Service process also reads the workflow from the repository at workflow run time. The PowerCenter Integration Service process reads all links and tasks in the workflow except sessions and worklet instances. The PowerCenter Integration Service process reads session instance information from the repository. The DTM retrieves the session and mapping from the repository at session run time. The PowerCenter Integration Service process reads worklets from the repository when the worklet starts.

Read the Parameter File

When the workflow starts, the PowerCenter Integration Service process checks the workflow properties for use of a parameter file. If the workflow uses a parameter file, the PowerCenter Integration Service process reads the parameter file and expands the variable values for the workflow and any worklets invoked by the workflow.

The parameter file can also contain mapping parameters and variables and session parameters for sessions in the workflow, as well as service and service process variables for the service process that runs the workflow. When starting the DTM, the PowerCenter Integration Service process passes the parameter file name to the DTM.

Create the PowerCenter Workflow Log

The PowerCenter Integration Service process creates a log for the PowerCenter workflow. The workflow log contains a history of the workflow run, including initialization, workflow task status, and error messages. You can use information in the workflow log in conjunction with the PowerCenter Integration Service log and session log to troubleshoot system, workflow, or session problems.

Run the PowerCenter Workflow Tasks

The PowerCenter Integration Service process runs workflow tasks according to the conditional links connecting the tasks. Links define the order of execution for workflow tasks. When a task in the workflow completes, the PowerCenter Integration Service process evaluates the completed task according to specified

conditions, such as success or failure. Based on the result of the evaluation, the PowerCenter Integration Service process runs successive links and tasks.

Run the PowerCenter Workflows Across the Nodes in a Grid

When you run a PowerCenter Integration Service on a grid, the service processes run workflow tasks across the nodes of the grid. The domain designates one service process as the master service process. The master service process monitors the worker service processes running on separate nodes. The worker service processes run workflows across the nodes in a grid.

Start the DTM Process

When the workflow reaches a session, the PowerCenter Integration Service process starts the DTM process. The PowerCenter Integration Service process provides the DTM process with session and parameter file information that allows the DTM to retrieve the session and mapping metadata from the repository. When you run a session on a grid, the worker service process starts multiple DTM processes that run groups of session threads.

When you use operating system profiles, the PowerCenter Integration Services starts the DTM process with the system user account you specify in the operating system profile.

Write Historical Information

The PowerCenter Integration Service process monitors the status of workflow tasks during the workflow run. When workflow tasks start or finish, the PowerCenter Integration Service process writes historical run information to the repository. Historical run information for tasks includes start and completion times and completion status. Historical run information for sessions also includes source read statistics, target load statistics, and number of errors. You can view this information using the PowerCenter Workflow Monitor.

Send Post-Session Email

The PowerCenter Integration Service process sends post-session email if the DTM terminates abnormally. The DTM sends post-session email in all other cases.

Load Balancer

The Load Balancer dispatches tasks to achieve optimal performance and scalability. When you run a workflow, the Load Balancer dispatches the Session, Command, and predefined Event-Wait tasks within the workflow. The Load Balancer matches task requirements with resource availability to identify the best node to run a task. It dispatches the task to a PowerCenter Integration Service process running on the node. It may dispatch tasks to a single node or across nodes.

The Load Balancer dispatches tasks in the order it receives them. When the Load Balancer needs to dispatch more Session and Command tasks than the PowerCenter Integration Service can run, it places the tasks it cannot run in a queue. When nodes become available, the Load Balancer dispatches tasks from the queue in the order determined by the workflow service level.

The following concepts describe Load Balancer functionality:

- Dispatch process. The Load Balancer performs several steps to dispatch tasks.
- Resources. The Load Balancer can use PowerCenter resources to determine if it can dispatch a task to a node.
- Resource provision thresholds. The Load Balancer uses resource provision thresholds to determine whether it can start additional tasks on a node.
- Dispatch mode. The dispatch mode determines how the Load Balancer selects nodes for dispatch.

- Service levels. When multiple tasks are waiting in the dispatch queue, the Load Balancer uses service levels to determine the order in which to dispatch tasks from the queue.

Dispatch Process

The Load Balancer uses different criteria to dispatch tasks depending on whether the PowerCenter Integration Service runs on a node or a grid.

Dispatch Tasks on a Node

When the PowerCenter Integration Service runs on a node, the Load Balancer performs the following steps to dispatch a task:

1. The Load Balancer checks resource provision thresholds on the node. If dispatching the task causes any threshold to be exceeded, the Load Balancer places the task in the dispatch queue, and it dispatches the task later.

The Load Balancer checks different thresholds depending on the dispatch mode.

2. The Load Balancer dispatches all tasks to the node that runs the master PowerCenter Integration Service process.

Dispatch Tasks Across a Grid

When the PowerCenter Integration Service runs on a grid, the Load Balancer performs the following steps to determine on which node to run a task:

1. The Load Balancer verifies which nodes are currently running and enabled.
2. If you configure the PowerCenter Integration Service to check resource requirements, the Load Balancer identifies nodes that have the PowerCenter resources required by the tasks in the workflow.
3. The Load Balancer verifies that the resource provision thresholds on each candidate node are not exceeded. If dispatching the task causes a threshold to be exceeded, the Load Balancer places the task in the dispatch queue, and it dispatches the task later.

The Load Balancer checks thresholds based on the dispatch mode.

4. The Load Balancer selects a node based on the dispatch mode.

Resources

You can configure the PowerCenter Integration Service to check the resources available on each node and match them with the resources required to run the task. If you configure the PowerCenter Integration Service to run on a grid and to check resources, the Load Balancer dispatches a task to a node where the required PowerCenter resources are available. For example, if a session uses an SAP source, the Load Balancer dispatches the session only to nodes where the SAP client is installed. If no available node has the required resources, the PowerCenter Integration Service fails the task.

You configure the PowerCenter Integration Service to check resources in the Administrator tool.

You define resources available to a node in the Administrator tool. You assign resources required by a task in the task properties.

The PowerCenter Integration Service writes resource requirements and availability information in the workflow log.

Resource Provision Thresholds

The Load Balancer uses resource provision thresholds to determine the maximum load acceptable for a node. The Load Balancer can dispatch a task to a node when dispatching the task does not cause the resource provision thresholds to be exceeded.

The Load Balancer checks the following thresholds:

- **Maximum CPU Run Queue Length.** The maximum number of runnable threads waiting for CPU resources on the node. The Load Balancer excludes the node if the maximum number of waiting threads is exceeded.

The Load Balancer checks this threshold in metric-based and adaptive dispatch modes.

- **Maximum Memory %.** The maximum percentage of virtual memory allocated on the node relative to the total physical memory size. The Load Balancer excludes the node if dispatching the task causes this threshold to be exceeded.

The Load Balancer checks this threshold in metric-based and adaptive dispatch modes.

- **Maximum Processes.** The maximum number of running processes allowed for each PowerCenter Integration Service process that runs on the node. The Load Balancer excludes the node if dispatching the task causes this threshold to be exceeded.

The Load Balancer checks this threshold in all dispatch modes.

If all nodes in the grid have reached the resource provision thresholds before any PowerCenter task has been dispatched, the Load Balancer dispatches tasks one at a time to ensure that PowerCenter tasks are still executed.

You define resource provision thresholds in the node properties.

Dispatch Mode

The dispatch mode determines how the Load Balancer selects nodes to distribute workflow tasks. The Load Balancer uses the following dispatch modes:

- **Round-robin.** The Load Balancer dispatches tasks to available nodes in a round-robin fashion. It checks the Maximum Processes threshold on each available node and excludes a node if dispatching a task causes the threshold to be exceeded. This mode is the least compute-intensive and is useful when the load on the grid is even and the tasks to dispatch have similar computing requirements.
- **Metric-based.** The Load Balancer evaluates nodes in a round-robin fashion. It checks all resource provision thresholds on each available node and excludes a node if dispatching a task causes the thresholds to be exceeded. The Load Balancer continues to evaluate nodes until it finds a node that can accept the task. This mode prevents overloading nodes when tasks have uneven computing requirements.
- **Adaptive.** The Load Balancer ranks nodes according to current CPU availability. It checks all resource provision thresholds on each available node and excludes a node if dispatching a task causes the thresholds to be exceeded. This mode prevents overloading nodes and ensures the best performance on a grid that is not heavily loaded.

When the Load Balancer runs in metric-based or adaptive mode, it uses task statistics to determine whether a task can run on a node. The Load Balancer averages statistics from the last three runs of the task to estimate the computing resources required to run the task. If no statistics exist in the repository, the Load Balancer uses default values.

In adaptive dispatch mode, the Load Balancer can use the CPU profile for the node to identify the node with the most computing resources.

You configure the dispatch mode in the domain properties.

Service Levels

Service levels establish priority among tasks that are waiting to be dispatched.

When the Load Balancer has more Session and Command tasks to dispatch than the PowerCenter Integration Service can run at the time, the Load Balancer places the tasks in the dispatch queue. When nodes become available, the Load Balancer dispatches tasks from the queue. The Load Balancer uses service levels to determine the order in which to dispatch tasks from the queue.

You create and edit service levels in the domain properties in the Administrator tool. You assign service levels to workflows in the workflow properties in the PowerCenter Workflow Manager.

Data Transformation Manager (DTM) Process

The DTM process is the operating system process that the PowerCenter Integration Service creates to run a DTM instance. The PowerCenter Integration Service creates a DTM instance to run each session, and it runs each DTM instance within a DTM process. The DTM process is also called the pmdtm process.

The DTM process performs the following tasks:

Reads the session information

The PowerCenter Integration Service process provides the DTM with session instance information when it starts the DTM. The DTM retrieves the mapping and session metadata from the repository and validates it.

Performs pushdown optimization

If the session is configured for pushdown optimization, the DTM runs an SQL statement to push transformation logic to the source or target database.

Creates dynamic partitions

The DTM adds partitions to the session if you configure the session for dynamic partitioning. The DTM scales the number of session partitions based on factors such as source database partitions or the number of nodes in a grid.

Forms partition groups

If you run a session on a grid, the DTM forms partition groups. A partition group is a group of reader, writer, and transformation threads that runs in a single DTM process. The DTM process forms partition groups and distributes them to worker DTM processes running on nodes in the grid.

Expands variables and parameters

If the workflow uses a parameter file, the PowerCenter Integration Service process sends the parameter file to the DTM when it starts the DTM. The DTM creates and expands session-level, service-level, and mapping-level variables and parameters.

Creates the session log

The DTM creates logs for the session. The session log contains a complete history of the session run, including initialization, transformation, status, and error messages. You can use information in the session log in conjunction with the PowerCenter Integration Service log and the workflow log to troubleshoot system or session problems.

Validates code pages

The PowerCenter Integration Service processes data internally using the UCS-2 character set. When you disable data code page validation, the PowerCenter Integration Service verifies that the source query,

target query, lookup database query, and stored procedure call text convert from the source, target, lookup, or stored procedure data code page to the UCS-2 character set without loss of data in conversion. If the PowerCenter Integration Service encounters an error when converting data, it writes an error message to the session log.

Verifies connection object permissions

After validating the session code pages, the DTM verifies permissions for connection objects used in the session. The DTM verifies that the user who started or scheduled the workflow has execute permissions for connection objects associated with the session.

Starts worker DTM processes

The DTM sends a request to the PowerCenter Integration Service process to start worker DTM processes on other nodes when the session is configured to run on a grid.

Runs pre-session operations

After verifying connection object permissions, the DTM runs pre-session shell commands. The DTM then runs pre-session stored procedures and SQL commands.

Runs the processing threads

After initializing the session, the DTM uses reader, transformation, and writer threads to extract, transform, and load data. The number of threads the DTM uses to run the session depends on the number of partitions configured for the session.

Runs post-session operations

After the DTM runs the processing threads, it runs post-session SQL commands and stored procedures. The DTM then runs post-session shell commands.

Sends post-session email

When the session finishes, the DTM composes and sends email that reports session completion or failure. If the DTM terminates abnormally, the PowerCenter Integration Service process sends post-session email.

Note: If you use operating system profiles, the PowerCenter Integration Service runs the DTM process as the operating system user you specify in the operating system profile.

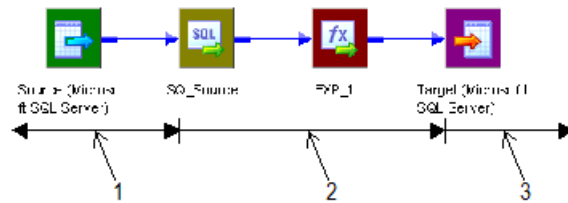
Processing Threads

The DTM allocates process memory for the session and divides it into buffers. This is also known as buffer memory. The DTM uses multiple threads to process data in a session. The main DTM thread is called the master thread.

The master thread creates and manages other threads. The master thread for a session can create mapping, pre-session, post-session, reader, transformation, and writer threads.

For each target load order group in a mapping, the master thread can create several threads. The types of threads depend on the session properties and the transformations in the mapping. The number of threads depends on the partitioning information for each target load order group in the mapping.

The following figure shows the threads the master thread creates for a simple mapping that contains one target load order group:

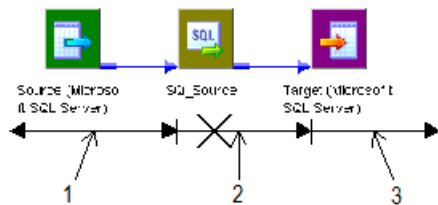


1. One reader thread.
2. One transformation thread.
3. One writer thread.

The mapping contains a single partition. In this case, the master thread creates one reader, one transformation, and one writer thread to process the data. The reader thread controls how the PowerCenter Integration Service process extracts source data and passes it to the source qualifier, the transformation thread controls how the PowerCenter Integration Service process handles the data, and the writer thread controls how the PowerCenter Integration Service process loads data to the target.

When the pipeline contains *only* a source definition, source qualifier, and a target definition, the data bypasses the transformation threads, proceeding directly from the reader buffers to the writer. This type of pipeline is a pass-through pipeline.

The following figure shows the threads for a pass-through pipeline with one partition:



1. One reader thread.
2. Bypassed transformation thread.
3. One writer thread.

Thread Types

The master thread creates different types of threads for a session. The types of threads the master thread creates depend on the pre- and post-session properties, as well as the types of transformations in the mapping.

The master thread can create the following types of threads:

- Mapping threads
- Pre- and post-session threads
- Reader threads
- Transformation threads
- Writer threads

Mapping Threads

The master thread creates one mapping thread for each session. The mapping thread fetches session and mapping information, compiles the mapping, and cleans up after session execution.

Pre- and Post-Session Threads

The master thread creates one pre-session and one post-session thread to perform pre- and post-session operations.

Reader Threads

The master thread creates reader threads to extract source data. The number of reader threads depends on the partitioning information for each pipeline. The number of reader threads equals the number of partitions. Relational sources use relational reader threads, and file sources use file reader threads.

The PowerCenter Integration Service creates an SQL statement for each reader thread to extract data from a relational source. For file sources, the PowerCenter Integration Service can create multiple threads to read a single source.

Transformation Threads

The master thread creates one or more transformation threads for each partition. Transformation threads process data according to the transformation logic in the mapping.

The master thread creates transformation threads to transform data received in buffers by the reader thread, move the data from transformation to transformation, and create memory caches when necessary. The number of transformation threads depends on the partitioning information for each pipeline.

Transformation threads store transformed data in a buffer drawn from the memory pool for subsequent access by the writer thread.

If the pipeline contains a Rank, Joiner, Aggregator, Sorter, or a cached Lookup transformation, the transformation thread uses cache memory until it reaches the configured cache size limits. If the transformation thread requires more space, it pages to local cache files to hold additional data.

When the PowerCenter Integration Service runs in ASCII mode, the transformation threads pass character data in single bytes. When the PowerCenter Integration Service runs in Unicode mode, the transformation threads use double bytes to move character data.

Writer Threads

The master thread creates writer threads to load target data. The number of writer threads depends on the partitioning information for each pipeline. If the pipeline contains one partition, the master thread creates one writer thread. If it contains multiple partitions, the master thread creates multiple writer threads.

Each writer thread creates connections to the target databases to load data. If the target is a file, each writer thread creates a separate file. You can configure the session to merge these files.

If the target is relational, the writer thread takes data from buffers and commits it to session targets. When loading targets, the writer commits data based on the commit interval in the session properties. You can configure a session to commit data based on the number of source rows read, the number of rows written to the target, or the number of rows that pass through a transformation that generates transactions, such as a Transaction Control transformation.

Pipeline Partitioning

When running sessions, the PowerCenter Integration Service process can achieve high performance by partitioning the pipeline and performing the extract, transformation, and load for each partition in parallel. To accomplish this, use the following session and PowerCenter Integration Service configuration:

- Configure the session with multiple partitions.
- Install the PowerCenter Integration Service on a machine with multiple CPUs.

You can configure the partition type at most transformations in the pipeline. The PowerCenter Integration Service can partition data using round-robin, hash, key-range, database partitioning, or pass-through partitioning.

You can also configure a session for dynamic partitioning to enable the PowerCenter Integration Service to set partitioning at run time. When you enable dynamic partitioning, the PowerCenter Integration Service scales the number of session partitions based on factors such as the source database partitions or the number of nodes in a grid.

For relational sources, the PowerCenter Integration Service creates multiple database connections to a single source and extracts a separate range of data for each connection.

The PowerCenter Integration Service transforms the partitions concurrently, it passes data between the partitions as needed to perform operations such as aggregation. When the PowerCenter Integration Service loads relational data, it creates multiple database connections to the target and loads partitions of data concurrently. When the PowerCenter Integration Service loads data to file targets, it creates a separate file for each partition. You can choose to merge the target files.

DTM Processing

When you run a session, the DTM process reads source data and passes it to the transformations for processing. To help understand DTM processing, consider the following DTM process actions:

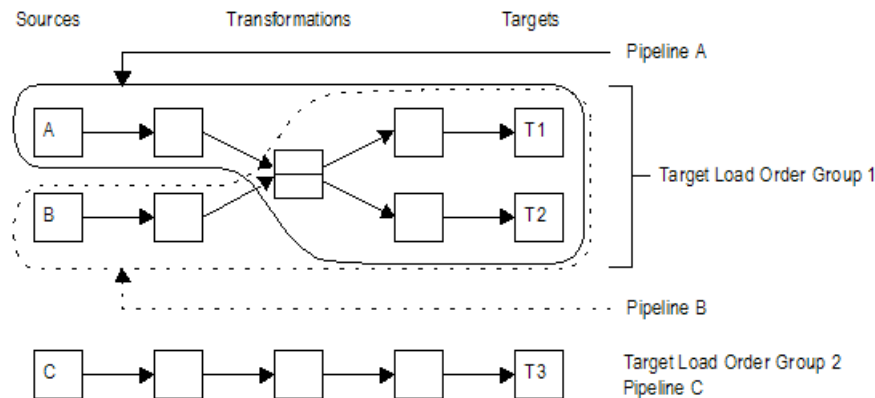
- Reading source data. The DTM reads the sources in a mapping at different times depending on how you configure the sources, transformations, and targets in the mapping.
- Blocking data. The DTM sometimes blocks the flow of data at a transformation in the mapping while it processes a row of data from a different source.
- Block processing. The DTM reads and processes a block of rows at a time.

Reading Source Data

Mappings contain one or more target load order groups. A target load order group is the collection of source qualifiers, transformations, and targets linked together in a mapping. Each target load order group contains one or more source pipelines. A source pipeline consists of a source qualifier and all of the transformations and target instances that receive data from that source qualifier.

By default, the DTM reads sources in a target load order group concurrently, and it processes target load order groups sequentially. You can configure the order that the DTM processes target load order groups.

The following figure shows a mapping that contains two target load order groups and three source pipelines:



In the mapping, the DTM processes the target load order groups sequentially. It first processes Target Load Order Group 1 by reading Source A and Source B at the same time. When it finishes processing Target Load Order Group 1, the DTM begins to process Target Load Order Group 2 by reading Source C.

Blocking Data

You can include multiple input group transformations in a mapping. The DTM passes data to the input groups concurrently. However, sometimes the transformation logic of a multiple input group transformation requires that the DTM block data on one input group while it waits for a row from a different input group.

Blocking is the suspension of the data flow into an input group of a multiple input group transformation. When the DTM blocks data, it reads data from the source connected to the input group until it fills the reader and transformation buffers. After the DTM fills the buffers, it does not read more source rows until the transformation logic allows the DTM to stop blocking the source. When the DTM stops blocking a source, it processes the data in the buffers and continues to read from the source.

The DTM blocks data at one input group when it needs a specific row from a different input group to perform the transformation logic. After the DTM reads and processes the row it needs, it stops blocking the source.

Block Processing

The DTM reads and processes a block of rows at a time. The number of rows in the block depend on the row size and the DTM buffer size. In the following circumstances, the DTM processes one row in a block:

- Log row errors. When you log row errors, the DTM processes one row in a block.
- Connect CURRVAL. When you connect the CURRVAL port in a Sequence Generator transformation, the session processes one row in a block. For optimal performance, connect only the NEXTVAL port in mappings.
- Configure array-based mode for Custom transformation procedure. When you configure the data access mode for a Custom transformation procedure to be row-based, the DTM processes one row in a block. By default, the data access mode is array-based, and the DTM processes multiple rows in a block.

Grids

When you run a PowerCenter Integration Service on a grid, a master service process runs on one node and worker service processes run on the remaining nodes in the grid. The master service process runs the workflow and workflow tasks, and it distributes the Session, Command, and predefined Event-Wait tasks to itself and other nodes. A DTM process runs on each node where a session runs. If you run a session on a grid, a worker service process can run multiple DTM processes on different nodes to distribute session threads.

Workflow on a Grid

When you run a workflow on a grid, the PowerCenter Integration Service designates one service process as the master service process, and the service processes on other nodes as worker service processes. The master service process can run on any node in the grid.

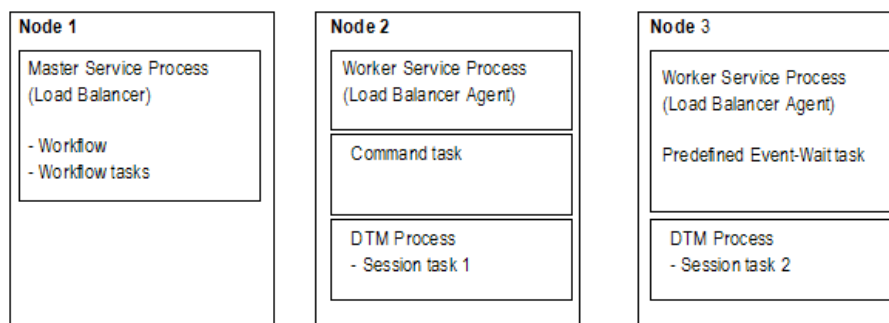
The master service process receives requests, runs the workflow and workflow tasks including the Scheduler, and communicates with worker service processes on other nodes. Because it runs on the master service process node, the Scheduler uses the date and time for the master service process node to start scheduled workflows. The master service process also runs the Load Balancer, which dispatches tasks to nodes in the grid.

Worker service processes running on other nodes act as Load Balancer agents. The worker service process runs predefined Event-Wait tasks within its process. It starts a process to run Command tasks and a DTM process to run Session tasks.

The master service process can also act as a worker service process. So the Load Balancer can distribute Session, Command, and predefined Event-Wait tasks to the node that runs the master service process or to other nodes.

For example, you have a workflow that contains two Session tasks, a Command task, and a predefined Event-Wait task.

The following figure shows an example of service process distribution when you run the workflow on a grid with three nodes:



When you run the workflow on a grid, the PowerCenter Integration Service process distributes the tasks in the following way:

- On Node 1, the master service process starts the workflow and runs workflow tasks other than the Session, Command, and predefined Event-Wait tasks. The Load Balancer dispatches the Session, Command, and predefined Event-Wait tasks to other nodes.
- On Node 2, the worker service process starts a process to run a Command task and starts a DTM process to run Session task 1.

- On Node 3, the worker service process runs a predefined Event-Wait task and starts a DTM process to run Session task 2.

Session on a Grid

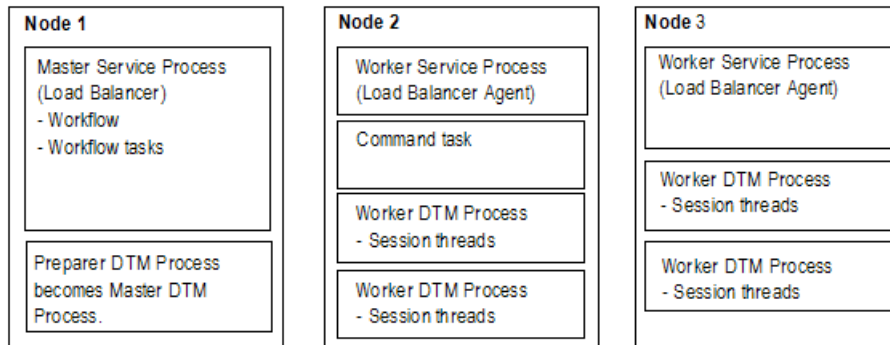
When you run a session on a grid, the master service process runs the workflow and workflow tasks, including the Scheduler. Because it runs on the master service process node, the Scheduler uses the date and time for the master service process node to start scheduled workflows. The Load Balancer distributes Command tasks as it does when you run a workflow on a grid. In addition, when the Load Balancer dispatches a Session task, it distributes the session threads to separate DTM processes.

The master service process starts a temporary preparer DTM process that fetches the session and prepares it to run. After the preparer DTM process prepares the session, it acts as the master DTM process, which monitors the DTM processes running on other nodes.

The worker service processes start the worker DTM processes on other nodes. The worker DTM runs the session. Multiple worker DTM processes running on a node might be running multiple sessions or multiple partition groups from a single session depending on the session configuration.

For example, you run a workflow on a grid that contains one Session task and one Command task. You also configure the session to run on the grid.

The following figure shows the service process and DTM distribution when you run a session on a grid on three nodes:



When the PowerCenter Integration Service process runs the session on a grid, it performs the following tasks:

- On Node 1, the master service process runs workflow tasks. It also starts a temporary preparer DTM process, which becomes the master DTM process. The Load Balancer dispatches the Command task and session threads to nodes in the grid.
- On Node 2, the worker service process runs the Command task and starts the worker DTM processes that run the session threads.
- On Node 3, the worker service process starts the worker DTM processes that run the session threads.

System Resources

To allocate system resources for read, transformation, and write processing, you should understand how the PowerCenter Integration Service allocates and uses system resources. The PowerCenter Integration Service uses the following system resources:

- CPU usage
- DTM buffer memory
- Cache memory

CPU Usage

The PowerCenter Integration Service process performs read, transformation, and write processing for a pipeline in parallel. It can process multiple partitions of a pipeline within a session, and it can process multiple sessions in parallel.

If you have a symmetric multi-processing (SMP) platform, you can use multiple CPUs to concurrently process session data or partitions of data. This provides increased performance, as true parallelism is achieved. On a single processor platform, these tasks share the CPU, so there is no parallelism.

The PowerCenter Integration Service process can use multiple CPUs to process a session that contains multiple partitions. The number of CPUs used depends on factors such as the number of partitions, the number of threads, the number of available CPUs, and amount of resources required to process the mapping.

DTM Buffer Memory

The PowerCenter Integration Service launches the DTM process. The DTM allocates buffer memory to the session based on the DTM Buffer Size setting in the session properties. By default, the PowerCenter Integration Service calculates the size of the buffer memory and the buffer block size.

The DTM divides the memory into buffer blocks as configured in the Buffer Block Size setting in the session properties. The reader, transformation, and writer threads use buffer blocks to move data from sources and to targets.

You may want to configure the buffer memory and buffer block size manually. In Unicode mode, the PowerCenter Integration Service uses double bytes to move characters, so increasing buffer memory might improve session performance.

If the DTM cannot allocate the configured amount of buffer memory for the session, the session cannot initialize. Informatica recommends you allocate no more than 1 GB for DTM buffer memory.

Cache Memory

The DTM process creates in-memory index and data caches to temporarily store data used by the following transformations:

- Aggregator transformation (without sorted input)
- Rank transformation
- Joiner transformation
- Lookup transformation (with caching enabled)

You can configure memory size for the index and data cache in the transformation properties. By default, the PowerCenter Integration Service determines the amount of memory to allocate for caches. However, you can manually configure a cache size for the data and index caches.

By default, the DTM creates cache files in the directory configured for the `$PMCacheDir` service process variable. If the DTM requires more space than it allocates, it pages to local index and data files.

The DTM process also creates an in-memory cache to store data for the Sorter transformations and XML targets. You configure the memory size for the cache in the transformation properties. By default, the PowerCenter Integration Service determines the cache size for the Sorter transformation and XML target at run time. The PowerCenter Integration Service allocates a minimum value of 16,777,216 bytes for the Sorter transformation cache and 10,485,760 bytes for the XML target. The DTM creates cache files in the directory configured for the `$PMTempDir` service process variable. If the DTM requires more cache space than it allocates, it pages to local cache files.

When processing large amounts of data, the DTM may create multiple index and data files. The session does not fail if it runs out of cache memory and pages to the cache files. It does fail, however, if the local directory for cache files runs out of disk space.

After the session completes, the DTM releases memory used by the index and data caches and deletes any index and data files. However, if the session is configured to perform incremental aggregation or if a Lookup transformation is configured for a persistent lookup cache, the DTM saves all index and data cache information to disk for the next session run.

Code Pages and Data Movement Modes

You can configure PowerCenter to move single byte and multibyte data. The PowerCenter Integration Service can move data in either ASCII or Unicode data movement mode. These modes determine how the PowerCenter Integration Service handles character data. You choose the data movement mode in the PowerCenter Integration Service configuration settings. If you want to move multibyte data, choose Unicode data movement mode. To ensure that characters are not lost during conversion from one code page to another, you must also choose the appropriate code pages for your connections.

ASCII Data Movement Mode

Use ASCII data movement mode when all sources and targets are 7-bit ASCII or EBCDIC character sets. In ASCII mode, the PowerCenter Integration Service recognizes 7-bit ASCII and EBCDIC characters and stores each character in a single byte. When the PowerCenter Integration Service runs in ASCII mode, it does not validate session code pages. It reads all character data as ASCII characters and does not perform code page conversions. It also treats all numerics as U.S. Standard and all dates as binary data.

You can also use ASCII data movement mode when sources and targets are 8-bit ASCII.

Unicode Data Movement Mode

Use Unicode data movement mode when sources or targets use 8-bit or multibyte character sets and contain character data. In Unicode mode, the PowerCenter Integration Service recognizes multibyte character sets as defined by supported code pages.

If you configure the PowerCenter Integration Service to validate data code pages, the PowerCenter Integration Service validates source and target code page compatibility when you run a session. If you configure the PowerCenter Integration Service for relaxed data code page validation, the PowerCenter Integration Service lifts source and target compatibility restrictions.

The PowerCenter Integration Service converts data from the source character set to UCS-2 before processing, processes the data, and then converts the UCS-2 data to the target code page character set.

before loading the data. The PowerCenter Integration Service allots two bytes for each character when moving data through a mapping. It also treats all numerics as U.S. Standard and all dates as binary data.

The PowerCenter Integration Service code page must be a subset of the PowerCenter repository code page.

Output Files and Caches

The PowerCenter Integration Service process generates output files when you run workflows and sessions. By default, the PowerCenter Integration Service logs status and error messages to log event files. Log event files are binary files that the Log Manager uses to display log events. During each session, the PowerCenter Integration Service also creates a reject file. Depending on transformation cache settings and target types, the PowerCenter Integration Service may create additional files as well.

The PowerCenter Integration Service stores output files and caches based on the service process variable settings. Generate output files and caches in a specified directory by setting service process variables in the session or workflow properties, PowerCenter Integration Service properties, a parameter file, or an operating system profile.

If you define service process variables in more than one place, the PowerCenter Integration Service reviews the precedence of each setting to determine which service process variable setting to use:

1. PowerCenter Integration Service process properties. Service process variables set in the PowerCenter Integration Service process properties contain the default setting.
2. Operating system profile. Service process variables set in an operating system profile override service process variables set in the PowerCenter Integration Service properties. If you use operating system profiles, the PowerCenter Integration Service saves workflow recovery files to the \$PMStorageDir configured in the PowerCenter Integration Service process properties. The PowerCenter Integration Service saves session recovery files to the \$PMStorageDir configured in the operating system profile.
3. Parameter file. Service process variables set in parameter files override service process variables set in the PowerCenter Integration Service process properties or an operating system profile.
4. Session or workflow properties. Service process variables set in the session or workflow properties override service process variables set in the PowerCenter Integration Service properties, a parameter file, or an operating system profile.

For example, if you set the \$PMSessionLogFile in the operating system profile and in the session properties, the PowerCenter Integration Service uses the location specified in the session properties.

The PowerCenter Integration Service creates the following output files:

- Workflow log
- Session log
- Session details file
- Performance details file
- Reject files
- Row error logs
- Recovery tables and files
- Control file
- Post-session email
- Output file
- Cache files

When the PowerCenter Integration Service process on UNIX creates any file other than a recovery file, it sets the file permissions according to the umask of the shell that starts the PowerCenter Integration Service process. For example, when the umask of the shell that starts the PowerCenter Integration Service process is 022, the PowerCenter Integration Service process creates files with rw-r--r-- permissions. To change the file permissions, you must change the umask of the shell that starts the PowerCenter Integration Service process and then restart it.

The PowerCenter Integration Service process on UNIX creates recovery files with rw----- permissions.

The PowerCenter Integration Service process on Windows creates files with read and write permissions.

Workflow Log

The PowerCenter Integration Service process creates a workflow log for each workflow it runs. It writes information in the workflow log such as initialization of processes, workflow task run information, errors encountered, and workflow run summary. Workflow log error messages are categorized into severity levels. You can configure the PowerCenter Integration Service to suppress writing messages to the workflow log file. You can view workflow logs from the PowerCenter Workflow Monitor. You can also configure the workflow to write events to a log file in a specified directory.

As with PowerCenter Integration Service logs and session logs, the PowerCenter Integration Service process enters a code number into the workflow log file message along with message text.

Session Log

The PowerCenter Integration Service process creates a session log for each session it runs. It writes information in the session log such as initialization of processes, session validation, creation of SQL commands for reader and writer threads, errors encountered, and load summary. The amount of detail in the session log depends on the tracing level that you set. You can view the session log from the PowerCenter Workflow Monitor. You can also configure the session to write the log information to a log file in a specified directory.

As with PowerCenter Integration Service logs and workflow logs, the PowerCenter Integration Service process enters a code number along with message text.

Session Details

When you run a session, the PowerCenter Workflow Manager creates session details that provide load statistics for each target in the mapping. You can monitor session details during the session or after the session completes. Session details include information such as table name, number of rows written or rejected, and read and write throughput. To view session details, double-click the session in the PowerCenter Workflow Monitor.

Performance Detail File

The PowerCenter Integration Service process generates performance details for session runs. The PowerCenter Integration Service process writes the performance details to a file. The file stores performance details for the last session run.

You can review a performance details file to determine where session performance can be improved. Performance details provide transformation-by-transformation information on the flow of data through the session.

You can also view performance details in the PowerCenter Workflow Monitor if you configure the session to collect performance details.

Reject Files

By default, the PowerCenter Integration Service process creates a reject file for each target in the session. The reject file contains rows of data that the writer does not write to targets.

The writer may reject a row in the following circumstances:

- It is flagged for reject by an Update Strategy or Custom transformation.
- It violates a database constraint such as primary key constraint.
- A field in the row was truncated or overflowed, and the target database is configured to reject truncated or overflowed data.

By default, the PowerCenter Integration Service process saves the reject file in the directory entered for the service process variable `$PMBadFileDir` in the PowerCenter Workflow Manager, and names the reject file `target_table_name.bad`.

Note: If you enable row error logging, the PowerCenter Integration Service process does not create a reject file.

Row Error Logs

When you configure a session, you can choose to log row errors in a central location. When a row error occurs, the PowerCenter Integration Service process logs error information that allows you to determine the cause and source of the error. The PowerCenter Integration Service process logs information such as source name, row ID, current row data, transformation, timestamp, error code, error message, repository name, folder name, session name, and mapping information.

When you enable flat file logging, by default, the PowerCenter Integration Service process saves the file in the directory entered for the service process variable `$PMBadFileDir`.

Recovery Tables Files

The PowerCenter Integration Service process creates recovery tables on the target database system when it runs a session enabled for recovery. When you run a session in recovery mode, the PowerCenter Integration Service process uses information in the recovery tables to complete the session.

When the PowerCenter Integration Service process performs recovery, it restores the state of operations to recover the workflow from the point of interruption. The workflow state of operations includes information such as active service requests, completed and running status, workflow variable values, running workflows and sessions, and workflow schedules.

Control File

When you run a session that uses an external loader, the PowerCenter Integration Service process creates a control file and a target flat file. The control file contains information about the target flat file such as data format and loading instructions for the external loader. The control file has an extension of `.ctl`. The PowerCenter Integration Service process creates the control file and the target flat file in the PowerCenter Integration Service variable directory, `$PMTargetFileDir`, by default.

Email

You can compose and send email messages by creating an Email task in the Workflow Designer or Task Developer. You can place the Email task in a workflow, or you can associate it with a session. The Email task allows you to automatically communicate information about a workflow or session run to designated recipients.

Email tasks in the workflow send email depending on the conditional links connected to the task. For post-session email, you can create two different messages, one to be sent if the session completes successfully, the other if the session fails. You can also use variables to generate information about the session name, status, and total rows loaded.

Indicator File

If you use a flat file as a target, you can configure the PowerCenter Integration Service to create an indicator file for target row type information. For each target row, the indicator file contains a number to indicate whether the row was marked for insert, update, delete, or reject. The PowerCenter Integration Service process names this file *target_name.ind* and stores it in the PowerCenter Integration Service variable directory, *\$PMTargetFileDir*, by default.

Output File

If the session writes to a target file, the PowerCenter Integration Service process creates the target file based on a file target definition. By default, the PowerCenter Integration Service process names the target file based on the target definition name. If a mapping contains multiple instances of the same target, the PowerCenter Integration Service process names the target files based on the target instance name.

The PowerCenter Integration Service process creates this file in the PowerCenter Integration Service variable directory, *\$PMTargetFileDir*, by default.

Cache Files

When the PowerCenter Integration Service process creates memory cache, it also creates cache files. The PowerCenter Integration Service process creates cache files for the following mapping objects:

- Aggregator transformation
- Joiner transformation
- Rank transformation
- Lookup transformation
- Sorter transformation
- XML target

By default, the DTM creates the index and data files for Aggregator, Rank, Joiner, and Lookup transformations and XML targets in the directory configured for the *\$PMCacheDir* service process variable. The PowerCenter Integration Service process names the index file *PM*.idx*, and the data file *PM*.dat*. The PowerCenter Integration Service process creates the cache file for a Sorter transformation in the *\$PMTempDir* service process variable directory.

Incremental Aggregation Files

If the session performs incremental aggregation, the PowerCenter Integration Service process saves index and data cache information to disk when the session finished. The next time the session runs, the PowerCenter Integration Service process uses this historical information to perform the incremental aggregation. By default, the DTM creates the index and data files in the directory configured for the *\$PMCacheDir* service process variable. The PowerCenter Integration Service process names the index file *PMAGG*.dat* and the data file *PMAGG*.idx*.

Persistent Lookup Cache

If a session uses a Lookup transformation, you can configure the transformation to use a persistent lookup cache. With this option selected, the PowerCenter Integration Service process saves the lookup cache to disk the first time it runs the session, and then uses this lookup cache during subsequent session runs. By default, the DTM creates the index and data files in the directory configured for the \$PMCacheDir service process variable. If you do not name the files in the transformation properties, these files are named PMLKUP*.idx and PMLKUP*.dat.

CHAPTER 12

High Availability for the PowerCenter Integration Service

This chapter includes the following topics:

- [High Availability for the PowerCenter Integration Service Overview, 278](#)
- [Resilience, 278](#)
- [Restart and Failover, 280](#)
- [Recovery, 282](#)
- [PowerCenter Integration Service Failover and Recovery Configuration, 283](#)

High Availability for the PowerCenter Integration Service Overview

Configure high availability for the PowerCenter Integration Service to minimize interruptions to data integration tasks.

The PowerCenter Integration Service has the following high availability features that are available based on your license:

- **Resilience.** A PowerCenter Integration Service process is resilient to connections with PowerCenter Integration Service clients and with external components.
- **Restart and failover.** If the PowerCenter Integration Service process becomes unavailable, the Service Manager can restart the process or fail it over to another node.
- **Recovery.** When the PowerCenter Integration Service restarts or fails over a service process, it can automatically recover interrupted workflows that are configured for recovery.

Resilience

Based on your license, the PowerCenter Integration Service is resilient to the temporary unavailability of PowerCenter Integration Service clients and external components such as databases and FTP servers.

The PowerCenter Integration Service tries to reconnect to PowerCenter Integration Service clients within the PowerCenter Integration Service resilience timeout period. The PowerCenter Integration Service resilience timeout period is based on the resilience properties that you configure for the PowerCenter Integration

Service, PowerCenter Integration Service clients, and the domain. The PowerCenter Integration Service tries to reconnect to external components within the resilience timeout for the database or FTP connection object.

PowerCenter Integration Service Client Resilience

PowerCenter Integration Service clients are resilient to the temporary unavailability of the PowerCenter Integration Service.

The PowerCenter Integration Service can be unavailable because of network failure or because a PowerCenter Integration Service process fails. PowerCenter Integration Service clients include the application services, PowerCenter Client, the Service Manager, the Web Services Hub, and *pmcmd*. PowerCenter Integration Service clients also include applications developed using LMAPI.

External Component Resilience

A PowerCenter Integration Service process is resilient to temporary unavailability of external components.

External components can be temporarily unavailable because of network failure or because the component experiences a failure. If the PowerCenter Integration Service process loses the connection to an external component, it tries to reconnect to the component within the retry period for the connection object.

You can configure the following types of external resilience for the PowerCenter Integration Service:

Database and application connection resilience

The PowerCenter Integration Service depends on external database systems and applications to run sessions and workflows. It is resilient if the database or application supports resilience. The PowerCenter Integration Service is resilient to failures when it initializes the connection to the source or target and when it reads data from a source or writes data to a target. If a database or application is temporarily unavailable, the PowerCenter Integration Service tries to connect for a specified amount of time. You can configure the connection retry period for relational connection objects for some application connection objects.

PowerExchange does not support session-level runtime connection resilience for database connections other than those used for PowerExchange Express CDC for Oracle. If recovery from a dropped PowerExchange connection is required, configure the workflow for automatic recovery of terminated tasks.

Runtime resilience of connections between the PowerCenter Integration Service and PowerExchange Listener is optionally available for the initial connection attempt only. You must set the **Connection Retry Period** attribute to a value greater than 0 when you define PowerExchange Client for PowerCenter (PWXPC) relational and application connections. The Integration Service then retries the connection to the PowerExchange Listener after the initial connection attempt fails. If the Integration Service cannot connect to the PowerExchange Listener within the retry period, the session fails.

FTP connection resilience

If a connection is lost while the PowerCenter Integration Service is transferring files to or from an FTP server, the PowerCenter Integration Service tries to reconnect for the amount of time configured in the FTP connection object. The PowerCenter Integration Service is resilient to interruptions if the FTP server supports resilience.

Client connection resilience

You can configure connection resilience for PowerCenter Integration Service clients that are external applications using C/Java LMAPI. You configure this type of resilience in the Application connection object.

Example

You configure a retry period of 180 for an Oracle relational database connection object. If the PowerCenter Integration Service loses connectivity to the database during the initial connection or when it reads data from the database, it tries to reconnect for 180 seconds. If it cannot reconnect to the database, the session fails.

Restart and Failover

If a PowerCenter Integration Service process becomes unavailable, the Service Manager tries to restart it or fails it over to another node based on the shutdown mode, the service configuration, and the operating mode for the service. Restart and failover behavior is different for services that run on a single node, primary and backup nodes, or on a grid.

When the PowerCenter Integration Service fails over, the behavior of completed tasks depends on the following situations:

- If a completed task reported a completed status to the PowerCenter Integration Service process prior to the PowerCenter Integration Service failure, the task will not restart.
- If a completed task did not report a completed status to the PowerCenter Integration Service process prior to the PowerCenter Integration Service failure, the task will restart.

Running on a Single Node

When a single process is running, the failover behavior depends on the following sources of failure:

Service Process

If the service process shuts down unexpectedly, the Service Manager tries to restart the service process. If the Service Manager cannot restart the process, the process stops or fails.

When you restart the process, the PowerCenter Integration Service restores the state of operation for the service and restores workflow schedules, service requests, and workflows.

The failover and recovery behavior of the PowerCenter Integration Service after a service process fails depends on the operating mode:

- **Normal.** When you restart the process, the workflow fails over on the same node. The PowerCenter Integration Service can recover the workflow based on the workflow state and recovery strategy. If the workflow is enabled for high availability recovery, the PowerCenter Integration Service restores the state of operation for the workflow and recovers the workflow from the point of interruption. The PowerCenter Integration Service performs failover and recovers the schedules, requests, and workflows. If a scheduled workflow is not enabled for high availability recovery, the PowerCenter Integration Service removes the workflow from the schedule.
- **Safe.** When you restart the process, the workflow does not fail over and the PowerCenter Integration Service does not recover the workflow. It performs failover and recovers the schedules, requests, and workflows when you enable the service in normal mode.

Service

When the PowerCenter Integration Service becomes unavailable, you must enable the service and start the service processes. You can manually recover workflows and sessions based on the state and the configured recovery strategy.

The workflows that run after you start the service processes depend on the operating mode:

- Normal. Workflows start if they are configured to run continuously or on initialization. You must reschedule all other workflows.
- Safe. Scheduled workflows do not start. You must enable the service in normal mode for the scheduled workflows to run.

Node

When the node becomes unavailable, the restart and failover behavior is the same as restart and failover for the service process, based on the operating mode.

Running on a Primary Node

When both primary and backup services are running, the failover behavior depends on the following sources of failure:

Service Process

When you disable the service process on a primary node, the service process fails over to a backup node. When the service process on a primary node shuts down unexpectedly, the Service Manager tries to restart the service process before failing it over to a backup node.

After the service process fails over to a backup node, the PowerCenter Integration Service restores the state of operation for the service and restores workflow schedules, service requests, and workflows.

The failover and recovery behavior of the PowerCenter Integration Service after a service process fails depends on the operating mode:

- Normal. The PowerCenter Integration Service can recover the workflow based on the workflow state and recovery strategy. If the workflow was enabled for high availability recovery, the PowerCenter Integration Service restores the state of operation for the workflow and recovers the workflow from the point of interruption. The PowerCenter Integration Service performs failover and recovers the schedules, requests, and workflows. If a scheduled workflow is not enabled for high availability recovery, the PowerCenter Integration Service removes the workflow from the schedule.
- Safe. The PowerCenter Integration Service does not run scheduled workflows and it disables schedule failover, automatic workflow recovery, workflow failover, and client request recovery. It performs failover and recovers the schedules, requests, and workflows when you enable the service in normal mode.

Service

When the PowerCenter Integration Service becomes unavailable, you must enable the service and start the service processes. You can manually recover workflows and sessions based on the state and the configured recovery strategy. Workflows start if they are configured to run continuously or on initialization. You must reschedule all other workflows.

The workflows that run after you start the service processes depend on the operating mode:

- Normal. Workflows start if they are configured to run continuously or on initialization. You must reschedule all other workflows.
- Safe. Scheduled workflows do not start. You must enable the service in normal mode to run the scheduled workflows.

Node

When the node becomes unavailable, the failover behavior is the same as the failover for the service process, based on the operating mode.

Running on a Grid

When a service is running on a grid, the failover behavior depends on the following sources of failure:

Master Service Process

If you disable the master service process, the Service Manager elects another node to run the master service process. If the master service process shuts down unexpectedly, the Service Manager tries to restart the process before electing another node to run the master service process.

The master service process then reconfigures the grid to run on one less node. The PowerCenter Integration Service restores the state of operation, and the workflow fails over to the newly elected master service process.

The PowerCenter Integration Service can recover the workflow based on the workflow state and recovery strategy. If the workflow was enabled for high availability recovery, the PowerCenter Integration Service restores the state of operation for the workflow and recovers the workflow from the point of interruption. When the PowerCenter Integration Service restores the state of operation for the service, it restores workflow schedules, service requests, and workflows. The PowerCenter Integration Service performs failover and recovers the schedules, requests, and workflows.

If a scheduled workflow is not enabled for high availability recovery, the PowerCenter Integration Service removes the workflow from the schedule.

Worker Service Process

If you disable a worker service process, the master service process reconfigures the grid to run on one less node. If the worker service process shuts down unexpectedly, the Service Manager tries to restart the process before the master service process reconfigures the grid.

After the master service process reconfigures the grid, it can recover tasks based on task state and recovery strategy.

Because workflows do not run on the worker service process, workflow failover is not applicable.

Service

When the PowerCenter Integration Service becomes unavailable, you must enable the service and start the service processes. You can manually recover workflows and sessions based on the state and the configured recovery strategy. Workflows start if they are configured to run continuously or on initialization. You must reschedule all other workflows.

Node

When the node running the master service process becomes unavailable, the failover behavior is the same as the failover for the master service process. When the node running the worker service process becomes unavailable, the failover behavior is the same as the failover for the worker service process.

Note: You cannot configure a PowerCenter Integration Service to fail over in safe mode when it runs on a grid.

Recovery

Based on your license, the PowerCenter Integration Service can automatically recover workflows and tasks based on the recovery strategy, the state of the workflows and tasks, and the PowerCenter Integration Service operating mode.

Stopped, Aborted, or Terminated Workflows

When the PowerCenter Integration Service restarts or fails over a service process, it can automatically recover interrupted workflows that are configured for recovery, based on the operating mode. When you run a workflow that is enabled for HA recovery, the PowerCenter Integration Service stores the state of operation in the `$PMStorageDir` directory. When the PowerCenter Integration Service recovers a workflow, it restores the state of operation and begins recovery from the point of interruption. The PowerCenter Integration Service can recover a workflow with a stopped, aborted, or terminated status.

In normal mode, the PowerCenter Integration Service can automatically recover the workflow. In safe mode, the PowerCenter Integration Service does not recover the workflow until you enable the service in normal mode.

When the PowerCenter Integration Service recovers a workflow that failed over, it begins recovery at the point of interruption. The PowerCenter Integration Service can recover a task with a stopped, aborted, or terminated status according to the recovery strategy for the task. The PowerCenter Integration Service behavior for task recovery does not depend on the operating mode.

Note: The PowerCenter Integration Service does not automatically recover a workflow or task that you stop or abort through the PowerCenter Workflow Monitor or `pmcmd`.

Running Workflows

You can configure automatic task recovery in the workflow properties. When you configure automatic task recovery, the PowerCenter Integration Service can recover terminated tasks while the workflow is running. You can also configure the number of times that the PowerCenter Integration Service tries to recover the task. If the PowerCenter Integration Service cannot recover the task in the configured number of times for recovery, the task and the workflow are terminated.

The PowerCenter Integration Service behavior for task recovery does not depend on the operating mode.

Suspended Workflows

The PowerCenter Integration Service can restore the workflow state after a suspended workflow fails over to another node if you enable recovery in the workflow properties.

If a service process shuts down while a workflow is suspended, the PowerCenter Integration Service marks the workflow as terminated. It fails the workflow over to another node, and changes the workflow state to terminated. The PowerCenter Integration Service does not recover any workflow task. You can fix the errors that caused the workflow to suspend, and manually recover the workflow.

PowerCenter Integration Service Failover and Recovery Configuration

During failover and recovery, the PowerCenter Integration Service needs to access state of operation files and process state information.

The state of operation files store the state of each workflow and session operation. The PowerCenter Integration Service always stores the state of each workflow and session operation in files in the `$PMStorageDir` directory of the PowerCenter Integration Service process.

Process state information includes information about which node was running the master PowerCenter Integration Service process and which node was running each session. You can configure the PowerCenter Integration Service to store process state information on a cluster file system or in the PowerCenter repository database.

Store High Availability Persistence on a Cluster File System

By default, the PowerCenter Integration Service stores process state information along with the state of operation files in the \$PMStorageDir directory of the Integration Service process. You must configure the \$PMStorageDir directory for each PowerCenter Integration Service process to use the same directory on a cluster file system.

Nodes that run the PowerCenter Integration Service must be on the same cluster file system so that they can share resources. Also, nodes within a cluster must be on the cluster file system's heartbeat network. Use a highly available cluster file system that is configured for I/O fencing. The hardware requirements and configuration of an I/O fencing solution are different for each file system.

The following cluster file systems are certified by Informatica for use for PowerCenter Integration Service failover and session recovery:

Storage Array Network

- Veritas Cluster Files System (VxFS)

- IBM General Parallel File System (GPFS)

Network Attached Storage using NFS v3 protocol

- EMC UxFS hosted on an EMV Celerra NAS appliance

- NetApp WAFL hosted on a NetApp NAS appliance

Contact the file system vendors directly to evaluate which file system matches your requirements.

Store High Availability Persistence in a Database

You can configure the PowerCenter Integration Service to store process state information in database tables. When you configure the PowerCenter Integration Service to store process state information in a database, the service still stores the state of each workflow and session operation in files in the \$PMStorageDir directory. You can configure the \$PMStorageDir directory to use a POSIX compliant shared file system. You do not need to use a cluster file system.

Configure the PowerCenter Integration Service to store process state information in database tables in the advanced properties. The PowerCenter Integration Service stores process state information in persistent database tables in the associated PowerCenter repository database.

During failover, automatic recovery of workflows resume when the service process can access the database tables.

CHAPTER 13

PowerCenter Repository Service

This chapter includes the following topics:

- [PowerCenter Repository Service Overview, 285](#)
- [Creating a Database for the PowerCenter Repository, 286](#)
- [Creating the PowerCenter Repository Service, 286](#)
- [PowerCenter Repository Service Properties, 289](#)
- [PowerCenter Repository Service Process Properties, 294](#)
- [High Availability for the PowerCenter Repository Service, 295](#)

PowerCenter Repository Service Overview

A PowerCenter repository is a collection of database tables that contains metadata. A PowerCenter Repository Service manages the PowerCenter repository. It performs all metadata transactions between the PowerCenter repository database and PowerCenter repository clients.

Create a PowerCenter Repository Service to manage the metadata in repository database tables. Each PowerCenter Repository Service manages a single repository. You need to create a unique PowerCenter Repository Service for each PowerCenter repository in a Informatica domain.

Creating and configuring a PowerCenter Repository Service involves the following tasks:

- Create a database for the repository tables. Before you can create the repository tables, you need to create a database to store the tables. If you create a PowerCenter Repository Service for an existing repository, you do not need to create a new database. You can use the existing database, as long as it meets the minimum requirements for a repository database.
- Create the PowerCenter Repository Service. Create the PowerCenter Repository Service to manage the repository. When you create a PowerCenter Repository Service, you can choose to create the repository tables. If you do not create the repository tables, you can create them later or you can associate the PowerCenter Repository Service with an existing repository.
- Configure the PowerCenter Repository Service. After you create a PowerCenter Repository Service, you can configure its properties. You can configure properties such as the error severity level or maximum user connections.

Based on your license, the PowerCenter Repository Service can be highly available.

Creating a Database for the PowerCenter Repository

Before you can manage a repository with a PowerCenter Repository Service, you need a database to hold the repository database tables. You can create the repository on any supported database system.

Use the database management system client to create the database. The repository database name must be unique. If you create a repository in a database with an existing repository, the create operation fails. You must delete the existing repository in the target database before creating the new repository.

To protect the repository and improve performance, do not create the repository on an overloaded machine. The machine running the repository database system must have a network connection to the node that runs the PowerCenter Repository Service.

Tip: You can optimize repository performance on IBM DB2 EEE databases when you store a PowerCenter repository in a single-node tablespace. When setting up an IBM DB2 EEE database, the database administrator must define the database on a single node.

Creating the PowerCenter Repository Service

Use the Administrator tool to create a PowerCenter Repository Service.

Before You Begin

Before you create a PowerCenter Repository Service, complete the following tasks:

- Determine repository requirements. Determine whether the repository needs to be version-enabled and whether it is a local, global, or standalone repository.
- Verify license. Verify that you have a valid license to run application services. Although you can create a PowerCenter Repository Service without a license, you need a license to run the service. In addition, you need a license to configure some options related to version control and high availability.
- Determine code page. Determine the code page to use for the PowerCenter repository. The PowerCenter Repository Service uses the character set encoded in the repository code page when writing data to the repository. The repository code page must be compatible with the code pages for the PowerCenter Client and all application services in the Informatica domain.

Tip: After you create the PowerCenter Repository Service, you cannot change the code page in the PowerCenter Repository Service properties. To change the repository code page after you create the PowerCenter Repository Service, back up the repository and restore it to a new PowerCenter Repository Service. When you create the new PowerCenter Repository Service, you can specify a compatible code page.

Creating a PowerCenter Repository Service

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the folder where you want to create the PowerCenter Repository Service.
Note: If you do not select a folder, you can move the PowerCenter Repository Service into a folder after you create it.
3. In the Domain Actions menu, click New > PowerCenter Repository Service.
The Create New Repository Service dialog box appears.

4. Enter values for the following PowerCenter Repository Service options.

The following table describes the PowerCenter Repository Service options:

Property	Description
Name	<p>Name of the PowerCenter Repository Service. The characters must be compatible with the code page of the repository. The name is not case sensitive and must be unique within the domain. It cannot exceed 128 characters or begin with @. It also cannot contain spaces or the following special characters:</p> <p>` ~ % ^ * + = { } \ ; : ' " / ? . , < > ! () []</p> <p>The PowerCenter Repository Service and the repository have the same name.</p>
Description	Description of PowerCenter Repository Service. The description cannot exceed 765 characters.
Location	Domain and folder where the service is created. Click Select Folder to choose a different folder. You can also move the PowerCenter Repository Service to a different folder after you create it.
License	License that allows use of the service. If you do not select a license when you create the service, you can assign a license later. The options included in the license determine the selections you can make for the repository. For example, you must have the team-based development option to create a versioned repository. Also, you need the high availability option to run the PowerCenter Repository Service on more than one node.
Node	Node on which the service process runs. Required if you do not select a license with the high availability option. If you select a license with the high availability option, this property does not appear.
Primary Node	Node on which the service process runs by default. Required if you select a license with the high availability option. This property appears if you select a license with the high availability option.
Backup Nodes	Nodes on which the service process can run if the primary node is unavailable. Optional if you select a license with the high availability option. This property appears if you select a license with the high availability option.
Database Type	Type of database storing the repository.
Code Page	Repository code page. The PowerCenter Repository Service uses the character set encoded in the repository code page when writing data to the repository. You cannot change the code page in the PowerCenter Repository Service properties after you create the PowerCenter Repository Service.
Connect String	Native connection string the PowerCenter Repository Service uses to access the repository database. For example, use <i>servername@dbname</i> for Microsoft SQL Server and <i>dbname.world</i> for Oracle.
Username	Account for the repository database. Set up this account using the appropriate database client tools.
Password	Repository database password corresponding to the database user. Must be in 7-bit ASCII.

Property	Description
Use DSN	Enables the PowerCenter Integration Service to use the Data Source Name from the Microsoft ODBC Administrator for the connection to a Microsoft SQL Server database. If you select the Use DSN option, the PowerCenter Integration Service retrieves the database and server names from the DSN. If you do not select the Use DSN option, you must provide the database and server names.
DataSource Name	Name of the datasource in the DSN.
TablespaceName	Tablespace name for IBM DB2 and Sybase repositories. When you specify the tablespace name, the PowerCenter Repository Service creates all repository tables in the same tablespace. You cannot use spaces in the tablespace name. To improve repository performance on IBM DB2 EEE repositories, specify a tablespace name with one node.
Creation Mode	Creates or omits new repository content. Select one of the following options: <ul style="list-style-type: none"> - Create repository content. Select if no content exists in the database. Optionally, choose to create a global repository, enable version control, or both. If you do not select these options during service creation, you can select them later. However, if you select the options during service creation, you cannot later convert the repository to a local repository or to a non-versioned repository. The option to enable version control appears if you select a license with the team-based development option. - Do not create repository content. Select if content exists in the database or if you plan to create the repository content later.
Enable the Repository Service	Enables the service. When you select this option, the service starts running when it is created. Otherwise, you need to click the Enable button to run the service. You need a valid license to run a PowerCenter Repository Service.

- If you create a PowerCenter Repository Service for a repository with existing content and the repository existed in a different Informatica domain, verify that users and groups with privileges for the PowerCenter Repository Service exist in the current domain.

The Service Manager periodically synchronizes the list of users and groups in the repository with the users and groups in the domain configuration database. During synchronization, users and groups that do not exist in the current domain are deleted from the repository. You can use *infacmd* to export users and groups from the source domain and import them into the target domain.

- Click OK.

Database Connect Strings

When you create a database connection, specify a connect string for that connection. The PowerCenter Repository Service uses native connectivity to communicate with the repository database.

The following table lists the native connect string syntax for each supported database:

Database	Connect String Syntax	Example
IBM DB2	<database name>	mydatabase
Microsoft SQL Server	<server name>@<database name>	sqlserver@mydatabase

Database	Connect String Syntax	Example
Oracle	<database name>.world (same as TNSNAMES entry)	oracle.world
Sybase	<server name>@<database name>	sybaseserver@mydatabase

PowerCenter Repository Service Properties

You can configure repository, node assignment, database, advanced, and custom properties for the PowerCenter Repository Service.

Use the Administrator tool to configure the following PowerCenter Repository Service properties:

- Repository properties. Configure repository properties, such as the Operating Mode.
- Node assignments. If you have the high availability option, configure the primary and backup nodes to run the service.
- Database properties. Configure repository database properties, such as the database user name, password, and connection string.
- Advanced properties. Configure advanced repository properties, such as the maximum connections and locks on the repository.
- Custom properties. Configure custom properties that are unique to specific environments.

To view and update properties, select the PowerCenter Repository Service in the Navigator. The Properties tab for the service appears.

Node Assignments

If you have the high availability option, you can designate primary and backup nodes to run the service. By default, the service runs on the primary node. If the node becomes unavailable, the service fails over to a backup node.

General Properties

To edit the general properties, select the PowerCenter Repository Service in the Navigator, select the **Properties** view, and then click **Edit** in the General Properties section.

The following table describes the general properties for the service:

Property	Description
Name	Name of the service. The name is not case sensitive and must be unique within the domain. It cannot exceed 128 characters or begin with @. It also cannot contain spaces or the following special characters: ` ~ % ^ * + = { } \ ; : ' " / ? . , < > ! () [] You cannot change the name of the service after you create it.
Description	Description of the service. The description cannot exceed 765 characters.

Property	Description
License	License object that allows use of the service.
Primary Node	Node on which the service runs. To assign the PowerCenter Repository Service to a different node, you must first disable the service.

Repository Properties

You can configure some of the repository properties when you create the service.

The following table describes the repository properties:

Property	Description
Operating Mode	Mode in which the PowerCenter Repository Service is running. Values are Normal and Exclusive. Run the PowerCenter Repository Service in exclusive mode to perform some administrative tasks, such as promoting a local repository to a global repository or enabling version control. To apply changes, restart the PowerCenter Repository Service.
Security Audit Trail	Tracks changes made to users, groups, privileges, permissions, and captures audit messages under the user activity logs on the imported .xml file. The Log Manager tracks the changes. To apply changes, restart the PowerCenter Repository Service.
Global Repository	Creates a global repository. If the repository is a global repository, you cannot revert back to a local repository. To promote a local repository to a global repository, the PowerCenter Repository Service must be running in exclusive mode.
Version Control	Creates a versioned repository. After you enable a repository for version control, you cannot disable the version control. To enable a repository for version control, you must run the PowerCenter Repository Service in exclusive mode. This property appears if you have the team-based development option.

Database Properties

Database properties provide information about the database that stores the repository metadata. You specify the database properties when you create the PowerCenter Repository Service. After you create a repository, you may need to modify some of these properties. For example, you might need to change the database user name and password, or you might want to adjust the database connection timeout.

The following table describes the database properties:

Property	Description
Database Type	Type of database storing the repository. To apply changes, restart the PowerCenter Repository Service.
Code Page	Repository code page. The PowerCenter Repository Service uses the character set encoded in the repository code page when writing data to the repository. You cannot change the code page in the PowerCenter Repository Service properties after you create the PowerCenter Repository Service. This is a read-only field.

Property	Description
Connect String	<p>Native connection string the PowerCenter Repository Service uses to access the database containing the repository. For example, use <i>servername@dbname</i> for Microsoft SQL Server and <i>dbname.world</i> for Oracle.</p> <p>To apply changes, restart the PowerCenter Repository Service.</p>
Table Space Name	<p>Tablespace name for IBM DB2 and Sybase repositories. When you specify the tablespace name, the PowerCenter Repository Service creates all repository tables in the same tablespace. You cannot use spaces in the tablespace name.</p> <p>You cannot change the tablespace name in the repository database properties after you create the service. If you create a PowerCenter Repository Service with the wrong tablespace name, delete the PowerCenter Repository Service and create a new one with the correct tablespace name.</p> <p>To improve repository performance on IBM DB2 EEE repositories, specify a tablespace name with one node.</p> <p>To apply changes, restart the PowerCenter Repository Service.</p>
Optimize Database Schema	<p>Enables optimization of repository database schema when you create repository contents or back up and restore an IBM DB2 or Microsoft SQL Server repository. When you enable this option, the Repository Service creates repository tables using Varchar(2000) columns instead of CLOB columns wherever possible. Using Varchar columns improves repository performance because it reduces disk input and output and because the database buffer cache can cache Varchar columns.</p> <p>To use this option, the repository database must meet the following page size requirements:</p> <ul style="list-style-type: none"> - IBM DB2: Database page size 4 KB or greater. At least one temporary tablespace with page size 16 KB or greater. - Microsoft SQL Server: Database page size 8 KB or greater. <p>Default is disabled.</p>
Database Username	<p>Account for the database containing the repository. Set up this account using the appropriate database client tools. To apply changes, restart the PowerCenter Repository Service.</p>
Database Password	<p>Repository database password corresponding to the database user. Must be in 7-bit ASCII. To apply changes, restart the PowerCenter Repository Service.</p>
Database Connection Timeout	<p>Period of time that the PowerCenter Repository Service tries to establish or reestablish a connection to the database system. Default is 180 seconds.</p>
Database Array Operation Size	<p>Number of rows to fetch each time an array database operation is issued, such as insert or fetch. Default is 100.</p> <p>To apply changes, restart the PowerCenter Repository Service.</p>
Database Pool Size	<p>Maximum number of connections to the repository database that the PowerCenter Repository Service can establish. If the PowerCenter Repository Service tries to establish more connections than specified for DatabasePoolSize, it times out the connection after the number of seconds specified for DatabaseConnectionTimeout. Default is 500. Minimum is 20.</p>
Table Owner Name	<p>Name of the owner of the repository tables for a DB2 repository.</p> <p>Note: You can use this option for DB2 databases only.</p>

Advanced Properties

Advanced properties control the performance of the PowerCenter Repository Service and the repository database.

The following table describes the advanced properties:

Property	Description
Authenticate MS-SQL User	Uses Windows authentication to access the Microsoft SQL Server database. The user name that starts the PowerCenter Repository Service must be a valid Windows user with access to the Microsoft SQL Server database. To apply changes, restart the PowerCenter Repository Service.
Required Comments for Checkin	Requires users to add comments when checking in repository objects. To apply changes, restart the PowerCenter Repository Service.
Minimum Severity for Log Entries	<p>Level of error messages written to the PowerCenter Repository Service log. Specify one of the following message levels:</p> <ul style="list-style-type: none">- Fatal- Error- Warning- Info- Trace- Debug <p>When you specify a severity level, the log includes all errors at that level and above. For example, if the severity level is Warning, fatal, error, and warning messages are logged. Use Trace or Debug if Informatica Global Customer Support instructs you to use that logging level for troubleshooting purposes. Default is INFO.</p>
Resilience Timeout	Period of time that the service tries to establish or reestablish a connection to another service. If blank, the service uses the domain resilience timeout. Default is 180 seconds.
Limit on Resilience Timeout	<p>Maximum amount of time that the service holds on to resources to accommodate resilience timeouts. This property limits the resilience timeouts for client applications connecting to the service. If a resilience timeout exceeds the limit, the limit takes precedence. If blank, the service uses the domain limit on resilience timeouts. Default is 180 seconds.</p> <p>To apply changes, restart the PowerCenter Repository Service.</p>
Repository Agent Caching	Enables repository agent caching. Repository agent caching provides optimal performance of the repository when you run workflows. When you enable repository agent caching, the PowerCenter Repository Service process caches metadata requested by the PowerCenter Integration Service. Default is Yes.
Agent Cache Capacity	Number of objects that the cache can contain when repository agent caching is enabled. You can increase the number of objects if there is available memory on the machine where the PowerCenter Repository Service process runs. The value must not be less than 100. Default is 10,000.

Property	Description
Allow Writes With Agent Caching	Allows you to modify metadata in the repository when repository agent caching is enabled. When you allow writes, the PowerCenter Repository Service process flushes the cache each time you save metadata through the PowerCenter Client tools. You might want to disable writes to improve performance in a production environment where the PowerCenter Integration Service makes all changes to repository metadata. Default is Yes.
Heart Beat Interval	Interval at which the PowerCenter Repository Service verifies its connections with clients of the service. Default is 60 seconds.
Maximum Active Users	Maximum number of connections the repository accepts from repository clients. Default is 200.
Maximum Object Locks	Maximum number of locks the repository places on metadata objects. Default is 50,000.
Database Pool Expiration Threshold	Minimum number of idle database connections allowed by the PowerCenter Repository Service. For example, if there are 20 idle connections, and you set this threshold to 5, the PowerCenter Repository Service closes no more than 15 connections. Minimum is 3. Default is 5.
Database Pool Expiration Timeout	Interval, in seconds, at which the PowerCenter Repository Service checks for idle database connections. If a connection is idle for a period of time greater than this value, the PowerCenter Repository Service can close the connection. Minimum is 300. Maximum is 2,592,000 (30 days). Default is 3,600 (1 hour).
Preserve MX Data for Old Mappings	Preserves MX data for old versions of mappings. When disabled, the PowerCenter Repository Service deletes MX data for old versions of mappings when you check in a new version. Default is disabled.

If you update the following properties, restart the PowerCenter Repository Service for the modifications to take effect:

- Minimum severity for log entries
- Maximum active users
- Maximum object locks

Metadata Manager Service Properties

You can access data lineage analysis for a PowerCenter repository from the PowerCenter Designer. To access data lineage from the Designer, you configure the Metadata Manager Service properties for the PowerCenter Repository Service.

Before you configure data lineage for a PowerCenter repository, complete the following tasks:

- Make sure Metadata Manager is running. Create a Metadata Manager Service in the Administrator tool or verify that an enabled Metadata Manager Service exists in the domain that contains the PowerCenter Repository Service for the PowerCenter repository.
- Load the PowerCenter repository metadata. Create a resource for the PowerCenter repository in Metadata Manager and load the PowerCenter repository metadata into the Metadata Manager warehouse.

The following table describes the Metadata Manager Service properties:

Property	Description
Metadata Manager Service	Name of the Metadata Manager Service used to run data lineage. Select from the available Metadata Manager Services in the domain.
Resource Name	Name of the PowerCenter resource in Metadata Manager.

Custom Properties for the PowerCenter Repository Service

Configure custom properties that are unique to specific environments.

You might need to apply custom properties in special cases. When you define a custom property, enter the property name and an initial value. Define custom properties only at the request of Informatica Global Customer Support.

PowerCenter Repository Service Process Properties

You can configure custom and environment variable properties for the PowerCenter Repository Service process.

Use the Administrator tool to configure the following PowerCenter Repository Service process properties:

- Custom properties. Configure custom properties that are unique to specific environments.
- Environment variables. Configure environment variables for each PowerCenter Repository Service process.

To view and update properties, select a PowerCenter Repository Service in the Navigator and click the Processes view.

Custom Properties for the PowerCenter Repository Service Process

Configure custom properties that are unique to specific environments.

You might need to apply custom properties in special cases. When you define a custom property, enter the property name and an initial value. Define custom properties only at the request of Informatica Global Customer Support.

Environment Variables

The database client path on a node is controlled by an environment variable.

Set the database client path environment variable for the PowerCenter Repository Service process if the PowerCenter Repository Service process requires a different database client than another PowerCenter Repository Service process that is running on the same node.

The database client code page on a node is usually controlled by an environment variable. For example, Oracle uses NLS_LANG, and IBM DB2 uses DB2CODEPAGE. All PowerCenter Integration Services and PowerCenter Repository Services that run on this node use the same environment variable. You can configure

a PowerCenter Repository Service process to use a different value for the database client code page environment variable than the value set for the node.

You can configure the code page environment variable for a PowerCenter Repository Service process when the PowerCenter Repository Service process requires a different database client code page than the PowerCenter Integration Service process running on the same node.

For example, the PowerCenter Integration Service reads from and writes to databases using the UTF-8 code page. The PowerCenter Integration Service requires that the code page environment variable be set to UTF-8. However, you have a Shift-JIS repository that requires that the code page environment variable be set to Shift-JIS. Set the environment variable on the node to UTF-8. Then add the environment variable to the PowerCenter Repository Service process properties and set the value to Shift-JIS.

High Availability for the PowerCenter Repository Service

Configure high availability for the PowerCenter Repository Service to minimize interruptions to data integration tasks.

The PowerCenter Repository Service has the following high availability features that are available based on your license:

- **Resilience.** The PowerCenter Repository Service is resilient to the temporary unavailability of other services and the repository database. PowerCenter Repository Service clients are resilient to connections with the PowerCenter Repository Service.
- **Restart and failover.** If the PowerCenter Repository Service fails, the Service Manager can restart the service or fail it over to another node, based on node availability.
- **Recovery.** After restart or failover, the PowerCenter Repository Service can recover operations from the point of interruption.

Resilience

The PowerCenter Repository Service is resilient to temporary unavailability of PowerCenter Repository Service clients and the PowerCenter Repository database.

An application service can be unavailable because of network failure or because a service process fails. You can configure the resilience timeout for the connection between the PowerCenter Repository Service and the following components:

PowerCenter Repository Service Clients

A PowerCenter Repository Service client can be a PowerCenter Client or a PowerCenter service that depends on the PowerCenter Repository Service. For example, the PowerCenter Integration Service is a PowerCenter Repository Service client because it depends on the PowerCenter Repository Service for a connection to the repository.

The PowerCenter Repository Service resilience timeout period is based on the resilience properties that you configure for the PowerCenter Repository Service, PowerCenter Repository Service clients, and the domain.

Note: The Web Services Hub is not resilient to the PowerCenter Repository Service.

PowerCenter Repository Database

The PowerCenter repository database might become unavailable because of network failure or because the repository database system becomes unavailable. If the repository database becomes unavailable, the PowerCenter Repository Service tries to reconnect to the repository database within the period specified by the database connection timeout configured in the PowerCenter Repository Service properties.

Tip: If the repository database system has high availability features, set the database connection timeout to allow the repository database system enough time to become available before the PowerCenter Repository Service tries to reconnect to it. Test the database system features that you plan to use to determine the optimum database connection timeout.

Restart and Failover

If the PowerCenter Repository Service process fails, the Service Manager can restart the process on the same node. If the node is not available, the PowerCenter Repository Service process fails over to the backup node.

The PowerCenter Repository Service process fails over to a backup node in the following situations:

- The PowerCenter Repository Service process fails and the primary node is not available.
- The PowerCenter Repository Service process is running on a node that fails.
- You disable the PowerCenter Repository Service process.

After failover, PowerCenter Repository Service clients synchronize and connect to the PowerCenter Repository Service process without loss of service.

You can disable a PowerCenter Repository Service process to shut down a node for maintenance. If you disable a PowerCenter Repository Service process in complete or abort mode, the PowerCenter Repository Service process fails over to another node.

Recovery

After a PowerCenter Repository Service restarts or fails over, it restores the state of operation from the repository and recovers operations from the point of interruption.

The PowerCenter Repository Service maintains the state of operation in the repository. The state of operations includes information about repository locks, requests in progress, and connected clients.

The PowerCenter Repository Service performs the following tasks to recover operations:

- Gets locks on repository objects, such as mappings and sessions
- Reconnects to clients, such as the PowerCenter Designer and the PowerCenter Integration Service
- Completes requests in progress, such as saving a mapping
- Sends outstanding notifications about metadata changes, such as workflow schedule changes

CHAPTER 14

PowerCenter Repository Management

This chapter includes the following topics:

- [PowerCenter Repository Management Overview, 297](#)
- [PowerCenter Repository Service and Service Processes, 298](#)
- [Operating Mode, 300](#)
- [PowerCenter Repository Content, 301](#)
- [Enabling Version Control, 303](#)
- [Managing a Repository Domain, 303](#)
- [Managing User Connections and Locks, 307](#)
- [Sending Repository Notifications, 309](#)
- [Backing Up and Restoring the PowerCenter Repository, 310](#)
- [Copying Content from Another Repository, 312](#)
- [Repository Plug-in Registration, 313](#)
- [Audit Trails, 314](#)
- [Repository Performance Tuning, 314](#)

PowerCenter Repository Management Overview

You use the Administrator tool to manage PowerCenter Repository Services and repository content. A PowerCenter Repository Service manages a single repository.

You can use the Administrator tool to complete the following repository tasks:

- Enable and disable a PowerCenter Repository Service or service process.
- Change the operating mode of a PowerCenter Repository Service.
- Create and delete repository content.
- Back up, copy, restore, and delete a repository.
- Promote a local repository to a global repository.
- Register and unregister a local repository.
- Manage user connections and locks.

- Send repository notification messages.
- Manage repository plug-ins.
- Configure permissions on the PowerCenter Repository Service.
- Upgrade a repository.
- Upgrade a PowerCenter Repository Service and its dependent services to the latest service version.

PowerCenter Repository Service and Service Processes

When you enable a PowerCenter Repository Service, a service process starts on a node designated to run the service. The service is available to perform repository transactions. If you have the high availability option, the service can fail over to another node if the current node becomes unavailable. If you disable the PowerCenter Repository Service, the service cannot run on any node until you reenables the service.

When you enable a service process, the service process is available to run, but it may not start. For example, if you have the high availability option and you configure a PowerCenter Repository Service to run on a primary node and two backup nodes, you enable PowerCenter Repository Service processes on all three nodes. A single process runs at any given time, and the other processes maintain standby status. If you disable a PowerCenter Repository Service process, the PowerCenter Repository Service cannot run on the particular node of the service process. The PowerCenter Repository Service continues to run on another node that is designated to run the service, as long as the node is available.

Enabling and Disabling a PowerCenter Repository Service

You can enable the PowerCenter Repository Service when you create it or after you create it. You need to enable the PowerCenter Repository Service to perform the following tasks in the Administrator tool:

- Assign privileges and roles to users and groups for the PowerCenter Repository Service.
- Create or delete content.
- Back up or restore content.
- Upgrade content.
- Copy content from another PowerCenter repository.
- Register or unregister a local repository with a global repository.
- Promote a local repository to a global repository.
- Register plug-ins.
- Manage user connections and locks.
- Send repository notifications.

You must disable the PowerCenter Repository Service to run it in its exclusive mode.

Note: Before you disable a PowerCenter Repository Service, verify that all users are disconnected from the repository. You can send a repository notification to inform users that you are disabling the service.

Enabling a PowerCenter Repository Service

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.

2. In the Domain Navigator, select the PowerCenter Repository Service.
3. In the **Manage** tab **Actions** menu, click **Enable**

The status indicator at the top of the contents panel indicates when the service is available.

Disabling a PowerCenter Repository Service

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the PowerCenter Repository Service.
3. On the **Manage** tab **Actions** menu, select **Disable Service**.
4. In the Disable Repository Service, select to abort all service processes immediately or allow services processes to complete.
5. Click **OK**.

Enabling and Disabling PowerCenter Repository Service Processes

A service process is the physical representation of a service running on a node. The process for a PowerCenter Repository Service is the *pmrepagent* process. At any given time, only one service process is running for the service in the domain.

When you create a PowerCenter Repository Service, service processes are enabled by default on the designated nodes, even if you do not enable the service. You disable and enable service processes on the Processes view. You may want to disable a service process to perform maintenance on the node or to tune performance.

If you have the high availability option, you can configure the service to run on multiple nodes. At any given time, a single process is running for the PowerCenter Repository Service. The service continues to be available as long as one of the designated nodes for the service is available. With the high availability option, disabling a service process does not disable the service if the service is configured to run on multiple nodes. Disabling a service process that is running causes the service to fail over to another node.

Enabling a PowerCenter Repository Service Process

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the PowerCenter Repository Service associated with the service process you want to enable.
3. In the contents panel, click the **Processes** view.
4. Select the process you want to enable.
5. In the **Manage** tab **Actions** menu, click **Enable Process** to enable the service process on the node.

Disabling a PowerCenter Repository Service Process

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the PowerCenter Repository Service associated with the service process you want to disable.
3. In the contents panel, click the **Processes** view.
4. Select the process you want to disable.
5. On the **Manage** tab **Actions** menu, select **Disable Process**.

6. In the dialog box that appears, select to abort service processes immediately or allow service processes to complete.
7. Click **OK**.

Operating Mode

You can run the PowerCenter Repository Service in normal or exclusive operating mode. When you run the PowerCenter Repository Service in normal mode, you allow multiple users to access the repository to update content. When you run the PowerCenter Repository Service in exclusive mode, you allow only one user to access the repository. Set the operating mode to exclusive to perform administrative tasks that require a single user to access the repository and update the configuration. If a PowerCenter Repository Service has no content associated with it or if a PowerCenter Repository Service has content that has not been upgraded, the PowerCenter Repository Service runs in exclusive mode only.

When the PowerCenter Repository Service runs in exclusive mode, it accepts connection requests from the Administrator tool and *pmrep*.

Run a PowerCenter Repository Service in exclusive mode to perform the following administrative tasks:

- Delete repository content. Delete the repository database tables for the PowerCenter repository.
- Enable version control. If you have the team-based development option, you can enable version control for the repository. A versioned repository can store multiple versions of an object.
- Promote a PowerCenter repository. Promote a local repository to a global repository to build a repository domain.
- Register a local repository. Register a local repository with a global repository to create a repository domain.
- Register a plug-in. Register or unregister a repository plug-in that extends PowerCenter functionality.
- Upgrade the PowerCenter repository. Upgrade the repository metadata.

Before running a PowerCenter Repository Service in exclusive mode, verify that all users are disconnected from the repository. You must stop and restart the PowerCenter Repository Service to change the operating mode.

When you run a PowerCenter Repository Service in exclusive mode, repository agent caching is disabled, and you cannot assign privileges and roles to users and groups for the PowerCenter Repository Service.

Note: You cannot use *pmrep* to log in to a new PowerCenter Repository Service running in exclusive mode if the Service Manager has not synchronized the list of users and groups in the repository with the list in the domain configuration database. To synchronize the list of users and groups, restart the PowerCenter Repository Service.

Running a PowerCenter Repository Service in Exclusive Mode

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the PowerCenter Repository Service.
3. In the Properties view, click Edit in the repository properties section.
4. Set the operating mode to Exclusive.
5. Click OK.

The Administrator tool prompts you to restart the PowerCenter Repository Service.

6. Verify that you have notified users to disconnect from the repository, and click Yes if you want to log out users who are still connected.

A warning message appears.

7. Choose to allow processes to complete or abort all processes, and then click OK.

The PowerCenter Repository Service stops and then restarts. The service status at the top of the right pane indicates when the service has restarted. The Disable button for the service appears when the service is enabled and running.

Note: PowerCenter does not provide resilience for a repository client when the PowerCenter Repository Service runs in exclusive mode.

Running a PowerCenter Repository Service in Normal Mode

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the PowerCenter Repository Service.
3. In the Properties view, click Edit in the repository properties section.
4. Select Normal as the operating mode.
5. Click OK.

The Administrator tool prompts you to restart the PowerCenter Repository Service.

Note: You can also use the *infacmd* UpdateRepositoryService command to change the operating mode.

PowerCenter Repository Content

Repository content are repository tables in the database. You can create or delete repository content for a PowerCenter Repository Service.

Creating PowerCenter Repository Content

You can create repository content for a PowerCenter Repository Service if you did not create content when you created the service or if you deleted the repository content. You cannot create content for a PowerCenter Repository Service that already has content.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select a PowerCenter Repository Service that has no content associated with it.
3. On the **Manage** tab **Actions** menu, select Repository Content > Create.

The page displays the options to create content.

4. Optionally, choose to create a global repository.

Select this option if you are certain you want to create a global repository. You can promote a local repository to a global repository at any time, but you cannot convert a global repository to a local repository.

5. Optionally, enable version control.

You must have the team-based development option to enable version control. Enable version control if you are certain you want to use a versioned repository. You can convert a non-versioned repository to a

versioned repository at any time, but you cannot convert a versioned repository to a non-versioned repository.

6. Click OK.

Deleting PowerCenter Repository Content

Delete repository content when you want to delete all metadata and repository database tables from the repository. When you delete repository content, you also delete all privileges and roles assigned to users for the PowerCenter Repository Service.

You might delete the repository content if the metadata is obsolete. Deleting repository content is an irreversible action. If the repository contains information that you might need later, back up the repository before you delete it.

To delete a global repository, you must unregister all local repositories. Also, you must run the PowerCenter Repository Service in exclusive mode to delete repository content.

Note: You can also use the *pmrep* Delete command to delete repository content.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the PowerCenter Repository Service from which you want to delete the content.
3. Change the operating mode of the PowerCenter Repository Service to exclusive.
4. On the **Manage** tab **Actions** menu, click Repository Content > Delete.
5. Enter your user name, password, and security domain.

The Security Domain field appears when the Informatica domain contains an LDAP security domain.

6. If the repository is a global repository, choose to unregister local repositories when you delete the content.

The delete operation does not proceed if it cannot unregister the local repositories. For example, if a Repository Service for one of the local repositories is running in exclusive mode, you may need to unregister that repository before you delete the global repository.

7. Click OK.

The activity log displays the results of the delete operation.

Upgrading PowerCenter Repository Content

To upgrade the PowerCenter repository content, you must have permission on the PowerCenter Repository Service.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the PowerCenter Repository Service for the repository you want to upgrade.
3. On the **Manage** tab **Actions** menu, click **Repository Contents > Upgrade**.
4. Enter the repository administrator user name and password.
5. Click **OK**.

The activity log displays the results of the upgrade operation.

Enabling Version Control

If you have the team-based development option, you can enable version control for a new or existing repository. A versioned repository can store multiple versions of objects. If you enable version control, you can maintain multiple versions of an object, control development of the object, and track changes. You can also use labels and deployment groups to associate groups of objects and copy them from one repository to another. After you enable version control for a repository, you cannot disable it.

When you enable version control for a repository, the repository assigns all versioned objects version number 1, and each object has an active status.

You must run the PowerCenter Repository Service in exclusive mode to enable version control for the repository.

1. Ensure that all users disconnect from the PowerCenter repository.
2. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
3. Change the operating mode of the PowerCenter Repository Service to exclusive.
4. Enable the PowerCenter Repository Service.
5. In the Domain Navigator, select the PowerCenter Repository Service.
6. In the repository properties section of the Properties view, click Edit.
7. Select Version Control.
8. Click OK.

The Repository Authentication dialog box appears.

9. Enter your user name, password, and security domain.

The Security Domain field appears when the Informatica domain contains an LDAP security domain.

10. Change the operating mode of the PowerCenter Repository Service to normal.

The repository is now versioned.

Managing a Repository Domain

A repository domain is a group of linked PowerCenter repositories that consists of one global repository and one or more local repositories. You group repositories in a repository domain to share data and metadata between repositories. When working in a repository domain, you can perform the following tasks:

- Promote metadata from a local repository to a global repository, making it accessible to all local repositories in the repository domain.
- Copy objects from or create shortcuts to metadata in the global repository.
- Copy objects from the local repository to the global repository.

Prerequisites for a PowerCenter Repository Domain

Before building a repository domain, verify that you have the following required elements:

- A licensed copy of Informatica to create the global repository.
- A license for each local repository you want to create.
- A database created and configured for each repository.

- A PowerCenter Repository Service created and configured to manage each repository.

A PowerCenter Repository Service accesses the repository faster if the PowerCenter Repository Service process runs on the machine where the repository database resides.

- Network connections between the PowerCenter Repository Services and PowerCenter Integration Services.
- Compatible repository code pages.

To register a local repository, the code page of the global repository must be a subset of each local repository code page in the repository domain. To copy objects from the local repository to the global repository, the code pages of the local and global repository must be compatible.

Building a PowerCenter Repository Domain

Use the following steps as a guideline to connect separate PowerCenter repositories into a repository domain:

1. Create a repository and configure it as a global repository. You can specify that a repository is the global repository when you create the PowerCenter Repository Service. Alternatively, you can promote an existing local repository to a global repository.
2. Register local repositories with the global repository. After a local repository is registered, you can connect to the global repository from the local repository and you can connect to the local repository from the global repository.

3. Create user accounts for users performing cross-repository work. A user who needs to connect to multiple repositories must have privileges for each PowerCenter Repository Service.

When the global and local repositories exist in different Informatica domains, the user must have an identical user name, password, and security domain in each Informatica domain. Although the user name, password, and security domain must be the same, the user can be a member of different user groups and can have a different set of privileges for each PowerCenter Repository Service.

4. Configure the user account used to access the repository associated with the PowerCenter Integration Service. To run a session that uses a global shortcut, the PowerCenter Integration Service must access the repository in which the mapping is saved and the global repository with the shortcut information. You enable this behavior by configuring the user account used to access the repository associated with the PowerCenter Integration Service. This user account must have privileges for the following services:
 - The local PowerCenter Repository Service associated with the PowerCenter Integration Service
 - The global PowerCenter Repository Service in the domain

Promoting a Local Repository to a Global Repository

You can promote an existing repository to a global repository. After you promote a repository to a global repository, you cannot change it to a local or standalone repository. After you promote a repository, you can register local repositories to create a repository domain.

When registering local repositories with a global repository, the global and local repository code pages must be compatible. Before promoting a repository to a global repository, make sure the repository code page is compatible with each local repository you plan to register.

To promote a repository to a global repository, you need to change the operating mode of the PowerCenter Repository Service to exclusive. If users are connected to the repository, have them disconnect before you run the repository in exclusive mode.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.

2. In the Domain Navigator, select the PowerCenter Repository Service for the repository you want to promote.
3. If the PowerCenter Repository Service is running in normal mode, change the operating mode to exclusive.
4. If the PowerCenter Repository Service is not enabled, click Enable.
5. In the repository properties section for the service, click Edit.
6. Select Global Repository, and click OK.
The Repository Authentication dialog box appears.
7. Enter your user name, password, and security domain.
The Security Domain field appears when the Informatica Domain contains an LDAP security domain.
8. Click OK.

After you promote a local repository, the value of the GlobalRepository property is true in the general properties for the PowerCenter Repository Service.

Registering a Local Repository

You can register local repositories with a global repository to create a repository domain. When you register a local repository, the code pages of the local and global repositories must be compatible. You can copy objects from the local repository to the global repository and create shortcuts. You can also copy objects from the global repository to the local repository.

If you unregister a repository from the global repository and register it again, the PowerCenter Repository Service re-establishes global shortcuts. For example, if you create a copy of the global repository and delete the original, you can register all local repositories with the copy of the global repository. The PowerCenter Repository Service reestablishes all global shortcuts unless you delete objects from the copied repository.

A separate PowerCenter Repository Service manages each repository. For example, if a repository domain has three local repositories and one global repository, it must have four PowerCenter Repository Services. The PowerCenter Repository Services and repository databases do not need to run on the same machine. However, you improve performance for repository transactions if the PowerCenter Repository Service process runs on the same machine where the repository database resides.

You can move a registered local or global repository to a different PowerCenter Repository Service in the repository domain or to a different Informatica domain.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the PowerCenter Repository Service associated with the local repository.
3. If the PowerCenter Repository Service is running in normal mode, change the operating mode to exclusive.
4. If the PowerCenter Repository Service is not enabled, click Enable.
5. To register a local repository, on the **Manage** tab **Actions** menu, click Repository Domain > Register Local Repository. Continue to the next step. To unregister a local repository, on the **Manage** tab **Actions** menu, click Repository Domain > Unregister Local Repository. Skip to step [11](#).
6. Select the Informatica domain of the PowerCenter Repository Service for the global repository.
If the PowerCenter Repository Service is in a domain that does not appear in the list of Informatica domains, click Manage Domain List to update the list.
The Manage List of Domains dialog box appears.

7. To add a domain to the list, enter the following information:

Field	Description
Domain Name	Name of a Informatica Domain that you want to link to.
Host Name	Machine hosting the master gateway node for the linked domain. The machine hosting the master gateway for the local Informatica Domain must have a network connection to this machine.
Host Port	Gateway port number for the linked domain.

8. Click Add to add more than one domain to the list, and repeat step [7](#) for each domain.
To edit the connection information for a linked domain, go to the section for the domain you want to update and click Edit.
To remove a linked domain from the list, go to the section for the domain you want to remove and click Delete.
9. Click Done to save the list of domains.
10. Select the PowerCenter Repository Service for the global repository.
11. Enter the user name, password, and security domain for the user who manages the global PowerCenter Repository Service.
The Security Domain field appears when the Informatica Domain contains an LDAP security domain.
12. Enter the user name, password, and security domain for the user who manages the local PowerCenter Repository Service.
13. Click OK.

Viewing Registered Local and Global Repositories

For a global repository, you can view a list of all the registered local repositories. Likewise, if a local repository is registered with a global repository, you can view the name of the global repository and the Informatica domain where it resides.

A PowerCenter Repository Service manages a single repository. The name of a repository is the same as the name of the PowerCenter Repository Service that manages it.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the PowerCenter Repository Service that manages the local or global repository.
3. On the **Manage** tab **Actions** menu, click Repository Domain > View Registered Repositories.

For a global repository, a list of local repositories appears.

For a local repository, the name of the global repository appears.

Note: The Administrator tool displays a message if a local repository is not registered with a global repository or if a global repository has no registered local repositories.

Moving Local and Global Repositories

If you need to move a local or global repository to another Informatica domain, complete the following steps:

1. Unregister the local repositories. For each local repository, follow the procedure to unregister a local repository from a global repository. To move a global repository to another Informatica domain, unregister all local repositories associated with the global repository.
2. Create the PowerCenter Repository Services using existing content. For each repository in the target domain, follow the procedure to create a PowerCenter Repository Service using the existing repository content in the source Informatica domain.

Verify that users and groups with privileges for the source PowerCenter Repository Service exist in the target domain. The Service Manager periodically synchronizes the list of users and groups in the repository with the users and groups in the domain configuration database. During synchronization, users and groups that do not exist in the target domain are deleted from the repository.

You can use *infacmd* to export users and groups from the source domain and import them into the target domain.

3. Register the local repositories. For each local repository in the target Informatica domain, follow the procedure to register a local repository with a global repository.

Managing User Connections and Locks

You can use the Administrator tool to manage user connections and locks and perform the following tasks:

- View locks. View object locks and lock type. The PowerCenter repository locks repository objects and folders by user. The repository uses locks to prevent users from duplicating or overwriting work. The repository creates different types of locks depending on the task.
- View user connections. View all user connections to the repository.
- Close connections and release locks. Terminate residual connections and locks. When you close a connection, you release all locks associated with that connection.

Viewing Locks

You can view locks and identify residual locks in the Administrator tool.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the PowerCenter Repository Service with the locks that you want to view.
3. In the contents panel, click the **Connections & Locks** view.
4. In the details panel, click the **Locks** view.

The following table describes the object lock information:

Column Name	Description
Server Thread ID	Identification number assigned to the repository connection.
Folder	Folder in which the locked object is saved.

Column Name	Description
Object Type	Type of object, such as folder, version, mapping, or source.
Object Name	Name of the locked object.
Lock Type	Type of lock: in-use, write-intent, or execute.
Lock Name	Name assigned to the lock.

Viewing User Connections

You can view user connection details in the Administrator tool. You might want to view user connections to verify all users are disconnected before you disable the PowerCenter Repository Service.

To view user connection details:

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the PowerCenter Repository Service with the locks that you want to view.
3. In the contents panel, click the **Connections & Locks** view.
4. In the details panel, click the **Properties** view.

The following table describes the user connection information:

Property	Description
Connection ID	Identification number assigned to the repository connection.
Status	Connection status.
Username	User name associated with the connection.
Security Domain	Security domain of the user.
Application	Repository client associated with the connection.
Service	Service that connects to the PowerCenter Repository Service.
Host Name	Name of the machine running the application.
Host Address	IP address for the host machine.
Host Port	Port number of the machine hosting the repository client used to communicate with the repository.
Process ID	Identifier assigned to the PowerCenter Repository Service process.
Login Time	Time when the user connected to the repository.
Last Active Time	Time of the last metadata transaction between the repository client and the repository.

Closing User Connections and Releasing Locks

Sometimes, the PowerCenter Repository Service does not immediately disconnect a user from the repository. The repository has a residual connection when the repository client or machine is shut down but the connection remains in the repository. This can happen in the following situations:

- Network problems occur.
- A PowerCenter Client, PowerCenter Integration Service, PowerCenter Repository Service, or database machine shuts down improperly.

A residual repository connection also retains all repository locks associated with the connection. If an object or folder is locked when one of these events occurs, the repository does not release the lock. This lock is called a residual lock.

If a system or network problem causes a repository client to lose connectivity to the repository, the PowerCenter Repository Service detects and closes the residual connection. When the PowerCenter Repository Service closes the connection, it also releases all repository locks associated with the connection.

A PowerCenter Integration Service may have multiple connections open to the repository. If you close one PowerCenter Integration Service connection to the repository, you close all connections for that service.

Important: Closing an active connection can cause repository inconsistencies. Close residual connections only.

To close a connection and release locks:

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the PowerCenter Repository Service with the connection you want to close.
3. In the contents panel, click the **Connections & Locks** view.
4. In the contents panel, select a connection.
The details panel displays connection properties in the properties view and locks in the locks view.
5. In the **Manage** tab **Actions** menu, select **Delete User Connection**.
The **Delete Selected Connection** dialog box appears.
6. Enter a user name, password, and security domain.
You can enter the login information associated with a particular connection, or you can enter the login information for the user who manages the PowerCenter Repository Service.
The **Security Domain** field appears when the Informatica domain contains an LAP security domain.
7. Click **OK**.

The PowerCenter Repository Service closes connections and releases all locks associated with the connections.

Sending Repository Notifications

You create and send notification messages to all users connected to a repository.

You might want to send a message to notify users of scheduled repository maintenance or other tasks that require you to disable a PowerCenter Repository Service or run it in exclusive mode. For example, you might

send a notification message to ask users to disconnect before you promote a local repository to a global repository.

1. Select the PowerCenter Repository Service in the Navigator.
2. In the **Manage** tab **Actions** menu, select **Notify Users**.
The **Notify Users** window appears.
3. Enter the message text.
4. Click **OK**.

The PowerCenter Repository Service sends the notification message to the PowerCenter Client users. A message box informs users that the notification was received. The message text appears on the Notifications tab of the PowerCenter Client Output window.

Backing Up and Restoring the PowerCenter Repository

Regularly back up repositories to prevent data loss due to hardware or software problems. When you back up a repository, the PowerCenter Repository Service saves the repository in a binary file, including the repository objects, connection information, and code page information. If you need to recover the repository, you can restore the content of the repository from this binary file.

If you back up a repository that has operating system profiles assigned to folders, the PowerCenter Repository Service does not back up the folder assignments. After you restore the repository, you must assign the operating system profiles to the folders.

Before you back up a repository and restore it in a different domain, verify that users and groups with privileges for the source PowerCenter Repository Service exist in the target domain. The Service Manager periodically synchronizes the list of users and groups in the repository with the users and groups in the domain configuration database. During synchronization, users and groups that do not exist in the target domain are deleted from the repository.

You can use *infacmd* to export users and groups from the source domain and import them into the target domain.

Backing Up a PowerCenter Repository

When you back up a repository, the PowerCenter Repository Service stores the file in the backup location you specify for the node. You specify the backup location when you set up the node. View the general properties of the node to determine the path of the backup directory. The PowerCenter Repository Service uses the extension *.rep* for all repository backup files.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the PowerCenter Repository Service for the repository you want to back up.
3. On the **Manage** tab **Actions** menu, select **Repository Contents** > **Back Up**.
4. Enter your user name, password, and security domain.
The Security Domain field appears when the Informatica domain contains an LDAP security domain.
5. Enter a file name and description for the repository backup file.

Use an easily distinguishable name for the file. For example, if the name of the repository is DEVELOPMENT, and the backup occurs on May 7, you might name the file DEVELOPMENTMay07.rep. If you do not include the .rep extension, the PowerCenter Repository Service appends that extension to the file name.

6. If you use the same file name that you used for a previous backup file, select whether or not to replace the existing file with the new backup file.

To overwrite an existing repository backup file, select Replace Existing File. If you specify a file name that already exists in the repository backup directory and you do not choose to replace the existing file, the PowerCenter Repository Service does not back up the repository.

7. Choose to skip or back up workflow and session logs, deployment group history, and MX data. You might want to skip these operations to increase performance when you restore the repository.
8. Click OK.

The results of the backup operation appear in the activity log.

Viewing a List of Backup Files

You can view the backup files you create for a repository in the backup directory where they are saved. You can also view a list of existing backup files in the Administrator tool. If you back up a repository through *pmrep*, you must provide a file extension of .rep to view it in the Administrator tool.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the PowerCenter Repository Service for a repository that has been backed up.
3. On the **Manage** tab **Actions** menu, select Repository Contents > View Backup Files.

The list of the backup files shows the repository version and the options skipped during the backup.

Restoring a PowerCenter Repository

You can restore metadata from a repository binary backup file. When you restore a repository, you must have a database available for the repository. You can restore the repository in a database that has a compatible code page with the original database.

If a repository exists at the target database location, you must delete it before you restore a repository backup file.

Informatica restores repositories from the current product version. If you have a backup file from an earlier product version, you must use the earlier product version to restore the repository.

Verify that the repository license includes the license keys necessary to restore the repository backup file. For example, you must have the team-based development option to restore a versioned repository.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the PowerCenter Repository Service that manages the repository content you want to restore.
3. On the **Manage** tab **Actions** menu, click Repository Contents > Restore.
The Restore Repository Contents options appear.
4. Select a backup file to restore.
5. Select whether or not to restore the repository as new.

When you restore a repository as new, the PowerCenter Repository Service restores the repository with a new repository ID and deletes the log event files.

Note: When you copy repository content, you create the repository as new.

6. Optionally, choose to skip restoring the workflow and session logs, deployment group history, and Metadata Exchange (MX) data to improve performance.
7. Click **OK**.

The activity log indicates whether the restore operation succeeded or failed.

Note: When you restore a global repository, the repository becomes a standalone repository. After restoring the repository, you need to promote it to a global repository.

Copying Content from Another Repository

Copy content into a repository when no content exists for the repository and you want to use the content from a different repository. Copying repository content provides a quick way to copy the metadata that you want to use as a basis for a new repository. You can copy repository content to preserve the original repository before upgrading. You can also copy repository content when you need to move a repository from development into production.

To copy repository content, you must create the PowerCenter Repository Service for the target repository. When you create the PowerCenter Repository Service, set the creation mode to create the PowerCenter Repository Service without content. Also, you must select a code page that is compatible with the original repository. Alternatively, you can delete the content from a PowerCenter Repository Service that already has content associated with it.

You must copy content into an empty repository. If repository in the target database already has content, the copy operation fails. You must back up the repository the target database and delete its content before copying the repository content.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the PowerCenter Repository Service to which you want to add copied content.

You cannot copy content to a repository that has content. If necessary, back up and delete existing repository content before copying in the new content.

3. On the **Manage** tab **Actions** menu, click Repository Contents > Copy From.

The dialog box displays the options for the Copy From operation.

4. Select the name of the PowerCenter Repository Service.

The source PowerCenter Repository Service and the PowerCenter Repository Service to which you want to add copied content must be in the same domain and it must be of the same service version.

5. Enter a user name, password, and security domain for the user who manages the repository from which you want to copy content.

The Security Domain field appears when the Informatica domain contains an LDAP security domain.

6. To skip copying the workflow and session logs, deployment group history, and Metadata Exchange (MX) data, select the check boxes in the advanced options. Skipping this data can increase performance.
7. Click **OK**.

The activity log displays the results of the copy operation.

Repository Plug-in Registration

Use the Administrator tool to register and remove repository plug-ins. Repository plug-ins are third-party or other Informatica applications that extend PowerCenter functionality by introducing new repository metadata.

For installation issues specific to the plug-in, consult the plug-in documentation.

Registering a Repository Plug-in

Register a repository plug-in to add its functionality to the repository. You can also update an existing repository plug-in.

1. Run the PowerCenter Repository Service in exclusive mode.
2. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
3. In the Domain Navigator, select the PowerCenter Repository Service to which you want to add the plug-in.
4. In the contents panel, click the Plug-ins view.
5. In the **Manage** tab **Actions** menu, select Register Plug-in.
6. On the Register Plug-in page, click the Browse button to locate the plug-in file.
7. If the plug-in was registered previously and you want to overwrite the registration, select the check box to update the existing plug-in registration. For example, you can select this option when you upgrade a plug-in to the latest version.
8. Enter your user name, password, and security domain.
The Security Domain field appears when the Informatica Domain contains an LDAP security domain.
9. Click OK.
The PowerCenter Repository Service registers the plug-in with the repository. The results of the registration operation appear in the activity log.
10. Run the PowerCenter Repository Service in normal mode.

Unregistering a Repository Plug-in

To unregister a repository plug-in, the PowerCenter Repository Service must be running in exclusive mode. Verify that all users are disconnected from the repository before you unregister a plug-in.

The list of registered plug-ins for a PowerCenter Repository Service appears on the Plug-ins tab.

If the PowerCenter Repository Service is not running in exclusive mode, the Remove buttons for plug-ins are disabled.

1. Run the PowerCenter Repository Service in exclusive mode.
2. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
3. In the Domain Navigator, select the PowerCenter Repository Service from which you want to remove the plug-in.
4. Click the Plug-ins view.
The list of registered plug-ins appears.
5. Select a plug-in and click the unregister Plug-in button.
6. Enter your user name, password, and security domain.

The Security Domain field appears when the Informatica Domain contains an LDAP security domain.

7. Click OK.
8. Run the PowerCenter Repository Service in normal mode.

Audit Trails

You can track changes to users, groups, and permissions on repository objects by selecting the SecurityAuditTrail configuration option in the PowerCenter Repository Service properties in the Administrator tool. When you enable the audit trail, the PowerCenter Repository Service logs security changes to the PowerCenter Repository Service log. The audit trail logs the following operations:

- Changing the owner or permissions for a folder or connection object.
- Adding or removing a user or group.

The audit trail does not log the following operations:

- Changing your own password.
- Changing the owner or permissions for a deployment group, label, or query.

Repository Performance Tuning

You can use the Informatica features to improve the performance of the repository. You can update statistics and skip information when you copy, back up, or restore the repository.

Repository Statistics

Almost all PowerCenter repository tables use at least one index to speed up queries. Most databases keep and use column distribution statistics to determine which index to use to execute SQL queries optimally. Database servers do not update these statistics continuously.

In frequently used repositories, these statistics can quickly become outdated, and SQL query optimizers might not choose the best query plan. In large repositories, choosing a sub-optimal query plan can have a negative impact on performance. Over time, repository operations gradually become slower.

Informatica identifies and updates the statistics of all repository tables and indexes when you copy, upgrade, and restore repositories. You can also update statistics using the pmrep UpdateStatistics command.

Repository Copy, Back Up, and Restore Processes

Large repositories can contain a large volume of log and historical information that slows down repository service performance. This information is not essential to repository service operation. When you back up, restore, or copy a repository, you can choose to skip the following types of information:

- Workflow and session logs
- Deployment group history
- Metadata Exchange (MX) data

By skipping this information, you reduce the time it takes to copy, back up, or restore a repository.

You can also skip this information when you use the *pmrep* commands.

CHAPTER 15

PowerExchange Listener Service

This chapter includes the following topics:

- [PowerExchange Listener Service Overview, 316](#)
- [DBMOVER Statements for the Listener Service, 317](#)
- [Creating a Listener Service, 318](#)
- [Listener Service Properties, 318](#)
- [Editing Listener Service Properties, 320](#)
- [Enabling, Disabling, and Restarting the Listener Service, 321](#)
- [Listener Service Logs, 322](#)
- [Listener Service Restart and Failover, 322](#)

PowerExchange Listener Service Overview

The PowerExchange Listener Service is an application service that manages the PowerExchange Listener.

The PowerExchange Listener manages communication between PowerExchange and a data source for bulk data movement or change data capture. You can define a PowerExchange Listener service so that when you run a workflow, PowerExchange on the PowerCenter Integration Service or Data Integration Service node connects to the PowerExchange Listener through the Listener Service. Use the Administrator tool to manage the service and view service logs.

When managed by the Listener Service, the PowerExchange Listener is also called the Listener Service process.

The Service Manager, Listener Service, and Listener Service process must reside on the same node in the Informatica domain.

On a Linux, UNIX, or Windows machine, you can use the Listener Service to manage the Listener process instead of issuing PowerExchange commands such as DTLLST to start the Listener process or CLOSE to stop the Listener process.

Note: If the PowerExchange Listener is running on i5/OS or z/OS, you cannot manage it with a PowerExchange Listener Service. Instead, manage the PowerExchange Listener by issuing z/OS or i5/OS commands or by issuing pwxcmd commands. For more information, see the *PowerExchange Command Reference*.

You can use the Administrator tool to perform the following Listener Service tasks:

- Create a service.

- View or edit service properties.
- View logs of service events.
- Enable, disable, or restart a service.

You can also use the `infacmd pwx` commands to perform many of these tasks.

Before you create a Listener Service, install PowerExchange and configure a PowerExchange Listener on the node where you want to create the Listener Service. When you create a Listener Service, the Service Manager associates it with the PowerExchange Listener on the node. When you start or stop the Listener Service, the PowerExchange Listener also starts or stops.

DBMOVER Statements for the Listener Service

Before you create a Listener Service, define `LISTENER` and `SVCNODE` statements in the `DBMOVER` file on each node in the Informatica domain where a PowerExchange Listener runs. Also, define a `NODE` statement in the `DBMOVER` file on each node where an Informatica client tool or integration service that connects to the Listener runs.

A client tool is the Developer tool or PowerCenter Client. An integration service is the PowerCenter Integration Service or Data Integration Service.

Define the following `DBMOVER` statements on all nodes where a PowerExchange Listener runs:

LISTENER

Required. Defines the TCP/IP port on which a named PowerExchange Listener process listens for work requests.

The node name in the `LISTENER` statement must match the name that you provide in the `Start Parameters` configuration property when you define the Listener Service.

SVCNODE

Optional. On Linux, UNIX, and Windows, use the `SVCNODE` statement to specify the TCP/IP port on which a PowerExchange Listener listens for `infacmd pwx` or `pwxcmd` commands.

This name must match the node name specified in the `LISTENER` statement in the `DBMOVER` configuration file.

Also, to issue `infacmd pwx` commands to connect to the Listener through the Listener application service, this name must match one of the following values:

- If you created the application service through Informatica Administrator, the node name value that you specified in the **Start Parameters** property.
- If you created the application service through the `infacmd pwx CreateListenerService` command, the node name value that you specified for the `-StartParameters` option on the command.

Use the same port number that you specify for the `SVCNODE` Port Number configuration property for the service.

Define the following `DBMOVER` statement on each node where an Informatica client tool or integration service that connects to the Listener runs:

NODE

Configures the Informatica client tool or integration service to connect to the PowerExchange Listener at the specified IP address or host name or to locate the Listener Service in the domain.

To configure the client tool or integration service to locate the Listener Service in the domain, include the optional *service_name* parameter in the NODE statement. The *service_name* parameter identifies the node, and the *port* parameter in the NODE statement identifies the port number.

Note: If the NODE statement does not include the *service_name* parameter, the Informatica client tool or integration service connects directly to the Listener at the specified IP address or host name. It does not locate the Listener Service in the domain.

For more information about customizing the DBMOVER configuration file for bulk data movement or CDC sessions, see the following guides:

- *PowerExchange Bulk Data Movement Guide*
- *PowerExchange CDC Guide for Linux, UNIX, and Windows*

Creating a Listener Service

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. Click **Actions > New > PowerExchange Listener Service**.
The **New PowerExchange Listener Service** dialog box appears.
3. Enter the general properties for the service, and click **Next**.
For more information, see [“PowerExchange Listener Service General Properties” on page 319](#).
4. Enter the configuration properties for the service.
For more information, see [“PowerExchange Listener Service Configuration Properties” on page 320](#).
5. Click **OK**.
6. To enable the Listener Service, select the service in the Domain Navigator and click **Enable the Service**.

Listener Service Properties

To view the properties of a Listener Service, select the service in the Domain Navigator and click the **Properties** tab.

You can change the properties while the service is running, but you must restart the service for the properties to take effect.

PowerExchange Listener Service General Properties

The following table describes the general properties for the service:

Property	Description
Name	Name of the service. The name is not case sensitive and must be unique within the domain. It cannot exceed 128 characters or begin with @. It also cannot contain spaces or the following special characters: ` ~ % ^ * + = { } \ ; : ' " / ? . , < > ! () [] You cannot change the name of the service after you create it.
Description	Description of the service. The description cannot exceed 765 characters.
Location	Domain and folder where the service is created. Click Browse to choose a different folder. You can move the service after you create it.
Node	Node on which the service runs.
License	License object that allows use of the service.
Backup Nodes	If your license includes high availability, nodes on which the service can run if the primary node is unavailable.

PowerExchange Listener Service Configuration Properties

The following table describes the configuration properties of a Listener Service:

Configuration Property	Description
Service Process	Read only. Type of PowerExchange process that the service manages. For the Listener Service, the service process is named Listener.
Start Parameters	<p>Parameters to include when you start the Listener Service. Separate the parameters with the space character.</p> <p>You can include the following parameters:</p> <ul style="list-style-type: none">- <i>service_name</i> Required. Name that identifies the Listener Service. This name must match the name in the LISTENER statement in the DBMOVER configuration file on the machine where the PowerExchange Listener runs.- <i>config=directory</i> Optional. Specifies the full path and file name for a DBMOVER configuration file that overrides the default dbmover.cfg file in the installation directory. This override file takes precedence over any other override configuration file that you optionally specify with the PWX_CONFIG environment variable.- <i>license=directory/license_key_file</i> Optional. Specifies the full path and file name for any license key file that you want to use instead of the default license.key file in the installation directory. This override license key file must have a file name or path that is different from that of the default file. This override file takes precedence over any other override license key file that you optionally specify with the PWX_LICENSE environment variable. Note: In the config and license parameters, you must provide the full path only if the file does <i>not</i> reside in the installation directory. Include double quotation marks around any path and file name that contains spaces.
SVC NODE Port Number	<p>Specifies the port on which the Listener Service connects to the PowerExchange Listener.</p> <p>Use the same port number that is specified in the SVCNODE statement of the DBMOVER file.</p>

Environment Variables for the Listener Service Process

You can edit environment variables for a Listener Service process on the **Processes** tab.

The following table describes the environment variables that are defined for the Listener Service process:

Property	Description
Environment Variables	Environment variables that are defined for the Listener Service process.

Editing Listener Service Properties

You can edit general and configuration properties for the Listener Service in the Administrator tool.

Editing Listener Service General Properties

Use the **Properties** tab in the Administrator tool to edit Listener Service general properties.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the PowerExchange Listener Service.
The **PowerExchange Listener Service Properties** window appears.
3. In the **General Properties** area of the **Properties** tab, click **Edit**.
The **Edit PowerExchange Listener Service** dialog box appears.
4. Edit the general properties of the service.
5. Click **OK**.

Editing Listener Service Configuration Properties

Use the **Properties** tab in the Administrator tool to configure Listener Service configuration properties.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the PowerExchange Listener Service.
3. In the **Configuration Properties** area of the **Properties** tab, click **Edit**.
The **Edit PowerExchange Listener Service** dialog box appears.
4. Edit the configuration properties.

Enabling, Disabling, and Restarting the Listener Service

You can enable, disable, or restart a Listener Service from the Administrator tool. You might disable the Listener Service if you need to temporarily restrict users from using the service. You might restart a service if you modified a property.

Enabling the Listener Service

To enable the Listener Service, select the service in the Domain Navigator and click **Enable the Service**.

Disabling the Listener Service

If you need to temporarily restrict users from using a Listener Service, you can disable it.

1. Select the service in the Domain Navigator, and click **Disable the Service**.
2. Select one of the following options:
 - **Complete**. Allows all Listener subtasks to run to completion before shutting down the service and the Listener Service process. Corresponds to the PowerExchange Listener CLOSE command.
 - **Stop**. Waits up to 30 seconds for subtasks to complete, and then shuts down the service and the Listener Service process. Corresponds to the PowerExchange Listener CLOSE FORCE command.
 - **Abort**. Stops all processes immediately and shuts down the service.
3. Click **OK**.

For more information about the CLOSE and CLOSE FORCE commands, see the *PowerExchange Command Reference*.

Note: After you select an option and click **OK**, the Administrator tool displays a busy icon until the service stops. If you select the **Complete** option but then want to disable the service more quickly with the **Stop** or **Abort** option, you must issue the `infacmd isp disableService` command.

Restarting the Listener Service

You can restart a Listener Service that you previously disabled.

To restart the Listener Service, select the service in the Navigator and click **Restart**.

Listener Service Logs

The Listener Service generates operational and error log events that the Log Manager collects in the domain.

You can view Listener Service logs by performing one of the following actions in the Administrator tool:

- In the **Logs** tab, select the **Domain** view. You can filter on any of the columns.
- In the **Logs** tab, click the **Service** view. In the **Service Type** column, select **PowerExchange Listener Service**. In the **Service Name** list, optionally select the name of the service.
- On the **Manage** tab, click the **Domain** view. Click the **Listener Service Actions** menu, and then select **View Logs**.

Messages appear by default in time stamp order, with the most recent messages on top.

Listener Service Restart and Failover

If you have the PowerCenter high availability option, the Listener Service provides restart and failover capabilities.

If the Listener Service or the Listener Service process fails on the primary node, the Service Manager restarts the service on the primary node.

If the primary node fails, the Listener Service fails over to the backup node, if one is defined. After failover, the Service Manager synchronizes and connects to the PowerExchange Listener on the backup node.

For the PowerExchange service to fail over successfully, the backup node must be able to connect to the data source or target. Configure the PowerExchange Listener and, if applicable, the PowerExchange Logger for Linux, UNIX, and Windows on the backup node as you do on the primary node.

If the PowerExchange Listener fails during a PowerCenter session, the session fails, and you must restart it. For CDC sessions, PWXPC performs warm start processing. For more information, see the *PowerExchange Interfaces Guide for PowerCenter*.

CHAPTER 16

PowerExchange Logger Service

This chapter includes the following topics:

- [PowerExchange Logger Service Overview, 323](#)
- [Configuration Statements for the Logger Service, 324](#)
- [Creating a Logger Service, 324](#)
- [Properties of the PowerExchange Logger Service, 325](#)
- [Logger Service Management, 327](#)
- [Enabling, Disabling, and Restarting the Logger Service, 328](#)
- [Logger Service Logs, 329](#)
- [Logger Service Restart and Failover, 329](#)

PowerExchange Logger Service Overview

The Logger Service is an application service that manages the PowerExchange Logger for Linux, UNIX, and Windows. The PowerExchange Logger captures change data from a data source and writes the data to PowerExchange Logger log files. Use the Administrator tool to manage the service and view service logs.

When managed by the Logger Service, the PowerExchange Logger is also called the Logger Service process.

The Service Manager, Logger Service, and PowerExchange Logger must reside on the same node in the Informatica domain.

On a Linux, UNIX, or Windows machine, you can use the Logger Service to manage the PowerExchange Logger process instead of issuing PowerExchange commands such as PWXCCL to start the Logger process or SHUTDOWN to stop the Logger process.

You can run multiple Logger Services on the same node. Create a Logger Service for each PowerExchange Logger process that you want to manage on the node. You must run one PowerExchange Logger process for each source type and instance, as defined in a PowerExchange registration group.

Perform the following tasks to manage the Logger Service:

- Create a service.
- View the service properties.
- View service logs
- Enable, disable, and restart the service.

You can use the Administrator tool or the *infacmd* command line program to administer the Logger Service.

Before you create a Logger Service, install PowerExchange and configure a PowerExchange Logger on the node where you want to create the Logger Service. When you create a Logger Service, the Service Manager associates it with the PowerExchange Logger that you specify. When you start or stop the Logger Service, you also start or stop the Logger Service process.

Configuration Statements for the Logger Service

The Logger Service reads configuration information from the DBMOVER and PowerExchange Logger Configuration (pwxcl.cfg) files.

Optionally, define the following statement in the DBMOVER file on each node that you configure to run the Logger Service:

SVCNODE

Optional. On Linux, UNIX, and Windows, use the SVCNODE statement to specify the TCP/IP port on which a PowerExchange Logger listens for infacmd pwx or pwxcmd commands.

The service name must match the service name that you specify in the associated CONDENSENAME statement in the pwxcl.cfg file. The port number must match the port number that you specify for the SVCNODE Port Number configuration property for the service.

Define the following statement in the PowerExchange Logger configuration file on each node that you configure to run the Logger Service:

CONDENSENAME

Name for the command-handling service for a PowerExchange Logger process to which commands are issued from the Logger Service.

Enter a service name up to 64 characters in length. No default is available.

The service name must match the service name that is specified in the associated SVCNODE statement in the dbmover.cfg file.

For more information about customizing the DBMOVER and PowerExchange Logger Configuration files for CDC sessions, see the *PowerExchange CDC Guide for Linux, UNIX, and Windows*.

Creating a Logger Service

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. Click **Actions > New > PowerExchange Logger Service**.
The New PowerExchange Logger Service dialog box appears.
3. Enter the service properties.
For more information, see the following topics:
 - [“PowerExchange Logger Service General Properties” on page 325](#)
 - [“PowerExchange Logger Service Configuration Properties” on page 325](#)
4. Click **OK**.
5. To enable the Logger Service, select the service in the Navigator and click **Enable the Service**.

Properties of the PowerExchange Logger Service

To view the properties of a PowerExchange Logger Service, select the service in the Domain Navigator and click the Properties tab.

You can change the properties while the service is running, but you must restart the service for the properties to take effect.

PowerExchange Logger Service General Properties

The following table describes the general properties for the service:

Property	Description
Name	Name of the service. The name is not case sensitive and must be unique within the domain. It cannot exceed 128 characters or begin with @. It also cannot contain spaces or the following special characters: ` ~ % ^ * + = { } \ ; : ' " / ? . , < > ! () [] You cannot change the name of the service after you create it.
Description	Description of the service. The description cannot exceed 765 characters.
Location	Domain and folder where the service is created. Click Browse to choose a different folder. You can move the service after you create it.
Node	Node on which the service runs.
License	License object that allows use of the service.
Backup Nodes	If your license includes high availability, nodes on which the service can run if the primary node is unavailable.

PowerExchange Logger Service Configuration Properties

The following table describes the configuration properties of a Logger Service:

Service Process

Read only. The type of PowerExchange process that the service manages. For a Logger Service, this value must be Logger.

Start Parameters

Optional. Parameters that you can specify when you start the Logger Service. If you specify more than one parameter, separate them with a space character.

Parameter descriptions:

- coldstart={Y|N}

Indicates whether to cold start or warm start the Logger Service. Enter Y to cold start the Logger Service. If the CDCT file contains log entries, the Logger Service deletes these entries. Enter N to warm start the Logger Service from the restart point that is specified in the CDCT file. If no restart information exists in the CDCT file, the Logger Service ends with an error.

Default is N.

- `config=directory/pwx_config_file`

Specifies the full path and file name for a dbmover configuration file that overrides the default dbmover.cfg file. The override file must have a path or file name that is different from that of the default file. This override file takes precedence over any configuration file that you optionally specify in the PWX_CONFIG environment variable.

- `cs=directory/pwxlogger_config_file`

Specifies the full path and file name for a Logger Service configuration file that overrides the default pwxcl.cfg configuration file. The override file must have a path or file name that is different from that of the default file.

- `encryptepwd=encrypted_password`

A password in encrypted format for enabling the encryption of PowerExchange Logger log files. With this password, the PowerExchange Logger can generate a unique encryption key for each Logger log file. The password is stored in the CDCT file in encrypted format. For security purposes, the password is not stored in CDCT backup files and is not displayed in the CDCT reports that you can generate with the PowerExchange PWXUCDCT utility.

If you specify this parameter, you must also specify `coldstart=Y`.

If you specify this parameter and also specify the ENCRYPTPWD parameter in the PowerExchange Logger configuration file, pwxcl.cfg, the parameter in the configuration file takes precedence. If you specify this parameter and also specify the ENCRYPTPWD parameter in the PowerExchange Logger configuration file, an error occurs.

You can set the AES algorithm to use for log file encryption in the ENCRYPTOPT parameter of the pwxcl.cfg file. The default is AES128.

Tip: For optimal security, Informatica recommends that you specify the encryption password when cold starting the PowerExchange Logger rather than in the pwxcl.cfg configuration file. This practice can reduce the risk of malicious access to the encryption password for the following reasons: 1) The encryption password is not stored in the pwxcl.cfg file, and 2) You can remove the password from the command line after a successful cold start. If you specify the encryption password for a cold start and then need to restore the CDCT file later, you must enter the same encryption password in the RESTORE_CDCT command of the PWXUCDCT utility.

To *not* encrypt PowerExchange Logger log files, do not enter an encryption password.

- `license=directory/license_key_file`

Specifies the full path and file name for a license key file that overrides the default license.key file. The override file must have a path or file name that is different from that of the default file. This override file takes precedence over any license key file that you optionally specify in the PWX_LICENSE environment variable.

- `specialstart={Y|N}`

Indicates whether to perform a special start of the PowerExchange Logger. A special start begins PowerExchange capture processing from the point in the change stream that you specify in the pwxcl.cfg file. This start point overrides the restart point from the CDCT file for the PowerExchange Logger run. A special start does not delete any content from the CDCT file.

Use this parameter to skip beyond problematic parts in the source logs without losing captured data. For example, use a special start in the following situations:

- You do not want the PowerExchange Logger to capture an upgrade of an Oracle catalog. In this case, stop the PowerExchange Logger before the upgrade. After the upgrade is complete, generate new sequence and restart tokens for the PowerExchange Logger based on the post-upgrade SCN. Enter these token values in the SEQUENCE_TOKEN and RESTART_TOKEN parameters in the pwxcl.cfg, and then special start the PowerExchange Logger.
- You do not want the PowerExchange Logger to reprocess old, unavailable logs that were caused by outstanding UOWs that are not of CDC interest. In this case, stop the PowerExchange Logger. Edit the RESTART_TOKEN value to reflect the SCN of the earliest available log, and then perform a special start. If any of the outstanding UOWs that started before this restart point are of CDC interest, data might be lost.

Valid values:

- Y. Perform a special start of the PowerExchange Logger from the point in the change stream that is defined by the SEQUENCE_TOKEN and RESTART_TOKEN parameter values in the pwxcl.cfg configuration file. You must specify valid token values in the pwxcl.cfg file to perform a special start. These token values override the token values from the CDCT file. Ensure that the SEQUENCE_TOKEN value in the pwxcl.cfg is greater than or equal to the current sequence token from the CDCT file.

Do not also specify the coldstart=Y parameter. If you do, the coldstart=Y parameter takes precedence.

- N. Do not perform a special start. Perform a cold start or warm start as indicated by the coldstart parameter.

Default is N.

Note: In the config, cs, and license parameters, provide the full path only if the file does *not* reside in the PowerExchange installation directory. Include quotes around any path and file name that contains spaces.

SVC NODE Port Number

Specifies the port on which the Logger Service connects to the PowerExchange Logger.

Use the same port number that is in the SVCNODE statement of the DBMOVER file.

Logger Service Management

Use the Properties tab in the Administrator tool to configure general or configuration properties for the Logger Service.

Configuring Logger Service General Properties

Use the Properties tab in the Administrator tool to configure Logger Service general properties.

1. In the Navigator, select the PowerExchange Logger Service.
The PowerExchange Logger Service properties window appears.
2. In the General Properties area of the Properties tab, click **Edit**.
The Edit PowerExchange Logger Service dialog box appears.

3. Edit the general properties of the service.
4. Click OK.

Configuring Logger Service Configuration Properties

Use the Properties tab in the Administrator tool to configure Logger Service configuration properties.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the PowerExchange Logger Service.
The PowerExchange Logger Service properties window appears.
3. In the Configuration Properties area of the Properties tab, click **Edit**.
The Edit PowerExchange Logger Service dialog box appears.
4. Edit the configuration properties for the service.

Configuring the Logger Service Process Properties

Use the Processes tab in the Administrator tool to configure the environment variables for each service process.

Environment Variables for the Logger Service Process

You can edit environment variables for a Logger Service process.

The following table describes the environment variables for the Logger Service process:

Property	Description
Environment Variables	Environment variables defined for the Logger Service process.

Enabling, Disabling, and Restarting the Logger Service

You can enable, disable, or restart a PowerExchange Logger Service by using the Administrator tool. You can disable a PowerExchange service if you need to temporarily restrict users from using the service. You might restart a service if you modified a property.

Enabling the Logger Service

To enable the Logger Service, select the service in the Navigator and click **Enable the Service**.

Disabling the Logger Service

If you need to temporarily restrict users from using the Logger Service, you can disable it.

1. Select the service in the Domain Navigator, and click **Disable the Service**.

2. Select one of the following options:
 - **Complete.** Initiates a controlled shutdown of all processes and shuts down the service. Corresponds to the PowerExchange SHUTDOWN command.
 - **Abort.** Stops all processes immediately and shuts down the service.
3. Click **OK**.

Restarting the Logger Service

You can restart a Logger Service that you previously disabled.

To restart the Logger Service, select the service in the Navigator and click **Restart**.

Logger Service Logs

The Logger Service generates operational and error log events that the Log Manager in the domain collects.

To view Logger Service logs, perform one of the following actions in the Administrator tool:

- In the Logs tab, select the **Domain** view. You can filter on any of the columns.
- In the Logs tab, click the **Service** view. In the **Service Type** column, select **PowerExchange Logger Service**. In the **Service Name** list, optionally select the name of the service.
- On the **Manage** tab, click the **Domain** view. Click the **Logger Service Actions** menu, and then select **View Logs**.

Messages appear by default in time stamp order, with the most recent messages on top.

Logger Service Restart and Failover

If you have the PowerCenter high availability option, the Logger Service provides restart and failover capabilities.

If the Logger Service or the Logger Service process fails on the primary node, the Service Manager restarts the service on the primary node.

If the primary node fails, the Logger Service fails over to the backup node, if one is defined. After failover, the Service Manager synchronizes and connects to the Logger Service process on the backup node.

For the Logger Service to fail over successfully, the Logger Service process on the backup node must be able to connect to the data source. Include the same statements in the DBMOVER and PowerExchange Logger configuration files on each node.

CHAPTER 17

SAP BW Service

This chapter includes the following topics:

- [SAP BW Service Overview, 330](#)
- [Creating the SAP BW Service, 331](#)
- [Enabling and Disabling the SAP BW Service, 333](#)
- [Configuring the SAP BW Service Properties, 334](#)
- [Configuring the Associated Integration Service, 336](#)
- [Configuring the SAP BW Service Processes, 336](#)
- [Load Balancing for the SAP BW System and the SAP BW Service, 337](#)
- [Viewing Log Events, 337](#)

SAP BW Service Overview

Create an SAP BW Service when you want to read data from or write data to SAP BW. Use the Administrator tool to create and manage the SAP BW Service.

The SAP BW Service is an application service that performs the following tasks:

- Listens for RFC requests from SAP BW.
- Initiates workflows to extract from or load to SAP BW.
- Sends log events to the Log Manager.

Use the Administrator tool to complete the following SAP BW Service tasks:

- Create the SAP BW Service.
- Enable and disable the SAP BW Service.
- Configure the SAP BW Service properties.
- Configure the associated PowerCenter Integration Service or Data Integration Service.
- Configure the SAP BW Service processes.
- Configure permissions for the SAP BW Service.
- View messages that the SAP BW Service sends to the Log Manager.

Creating the SAP BW Service

Create an SAP BW Service when you want to read data from or write data to SAP BW. Use the Administrator tool to create the SAP BW Service.

1. Log in to the Administrator tool.
2. In the Domain Navigator, select the domain.
3. Perform one of the following steps:
 - To create an SAP BW Service for PowerCenter, click **Actions > New > PowerCenter SAP BW Service**. The **New PowerCenter SAP BW Service** window appears.
 - To create an SAP BW Service for the Developer tool, click **Actions > New > SAP BW Service**. The **New SAP BW Service** window appears.
4. Configure the SAP BW Service properties.

The following table describes the information that you must enter when you create an SAP BW Service for PowerCenter:

Property	Description
Name	Name of the SAP BW Service. The characters must be compatible with the code page of the associated repository. The name is not case sensitive and must be unique within the domain. It cannot exceed 128 characters or begin with @. It also cannot contain spaces or the following special characters: ` ~ % ^ * + = { } \ ; : ' " / ? . , < > ! ()] [
Description	Description of the SAP BW Service. The description cannot exceed 765 characters.
Location	Name of the domain and folder in which the Administrator tool must create the SAP BW Service. By default, the Administrator tool creates the SAP BW Service in the domain where you are connected. Click Browse to select a new folder in the domain.
License	License file.
Node	Node on which the SAP BW Service must run.
SAP Destination R Type	DEST entry defined in the <code>sapnwrfc.ini</code> file to connect to the SAP BW Service.
Associated Integration Service	The PowerCenter Integration Service that you want to associate with the SAP BW Service.
Repository User Name	Account used to access the repository.

Property	Description
Repository Password	Password for the user. Note: If secure communication is enabled for the domain, you do not need to specify the repository password.
Security Domain	Security domain for the user. Appears when the Informatica domain contains an LDAP security domain.

The following table describes the information that you must enter when you create an SAP BW Service for the Developer tool:

Property	Description
Name	Name of the SAP BW Service. The characters must be compatible with the code page of the associated repository. The name is not case sensitive and must be unique within the domain. It cannot exceed 128 characters or begin with @. It also cannot contain spaces or the following special characters: ` ~ % ^ * + = { } \ ; : ' " / ? . , < > ! ()] [
Description	Description of the SAP BW Service. The description cannot exceed 765 characters.
Location	Name of the domain and folder in which the Administrator tool must create the SAP BW Service. By default, the Administrator tool creates the SAP BW Service in the domain where you are connected. Click Browse to select a new folder in the domain.
License	License file.
Node	Node on which the SAP BW Service must run.
Program ID	Program ID for the logical system that you create in SAP BW for the SAP BW Service. The Program ID in SAP BW must match this parameter, including case.
Gateway Host	Host name of the SAP gateway.
Gateway Server	Server name of the SAP gateway.
SAP Connection	SAP connection that you want to use. Specify a connection to a specific SAP application server or an SAP load balancing connection.
Trace	Use this option to track the JCo calls that the SAP system makes. SAP stores the information about the JCo calls in a trace file. Specify one of the following values: - 0. Off - 1. Full Default is 0. You can access the trace files from the following directory on the machine where you installed the Informatica services: <code><Informatica installation directory>/tomcat/bin</code>

Property	Description
Other Connection Parameters	Enter any other connection parameter that you want to use. Use the following format: <parameter name>=<value>
Associated Data Integration Service	The Data Integration Service that you want to associate with the SAP BW Service.
Repository User Name	Account used to access the repository.
Repository Password	Password for the user. Note: If secure communication is enabled for the domain, you do not need to specify the repository password.

5. Click **OK**.

The SAP BW Service is created.

Enabling and Disabling the SAP BW Service

Use the Administrator tool to enable and disable the SAP BW Service. You might disable the SAP BW Service if you need to perform maintenance on the machine where the SAP BW Service runs. Enable the disabled SAP BW Service to make it available again.

Before you enable the SAP BW Service, you must define Informatica as a logical system in SAP BW.

When you enable the SAP BW Service, the service starts. If the service cannot start, the domain tries to restart the service based on the restart options configured in the domain properties.

If the service is enabled but fails to start after reaching the maximum number of attempts, the following message appears:

```
The SAP BW Service <service name> is enabled.
The service did not start. Please check the logs for more information.
```

You can review the logs to determine the reason for failure and fix the problem. After you fix the problem, disable and re-enable the SAP BW Service to start it.

When you enable the SAP BW Service, it tries to connect to the associated Integration Service. If the Integration Service is not enabled and the SAP BW Service cannot connect to it, the SAP BW Service still starts successfully. When the SAP BW Service receives a request from SAP BW to start a workflow, the service tries to connect to the associated Integration Service again. If it cannot connect, the SAP BW Service returns the following message to the SAP BW system:

```
The SAP BW Service could not find Integration Service <service name> in domain <domain name>.
```

To resolve this problem, verify that the Integration Service is enabled, and that the domain name and Integration Service name that you entered under the third-party details of the InfoPackage are valid. Then, restart the process chain in the SAP BW system.

When you disable the SAP BW Service, select one of the following options:

- Complete. Disables the SAP BW Service after all service processes complete.

- **Abort.** Aborts all processes immediately and then disables the SAP BW Service. You might choose abort if a service process stops responding.

Enabling the SAP BW Service

1. In the Domain Navigator of the Administrator tool, select the SAP BW Service.
2. Click **Actions > Enable Service**.

Disabling the SAP BW Service

1. In the Domain Navigator of the Administrator tool, select the SAP BW Service.
2. Click **Actions > Disable Service**.
The **Disable SAP BW Service** window appears.
3. Select the disable mode and click **OK**.

Configuring the SAP BW Service Properties

Use the **Properties** tab in the Administrator tool to configure general properties for the SAP BW Service and to configure the node on which the service runs.

1. In the Domain Navigator, select the SAP BW Service.
The **SAP BW Service Properties** window appears.
2. In the **Properties** tab, click **Edit** corresponding to the category of properties that you want to update.
3. Update the property values and restart the SAP BW Service for the changes to take effect.

General Properties

The following table describes the general properties for the service:

Property	Description
Name	Name of the service. The name is not case sensitive and must be unique within the domain. It cannot exceed 128 characters or begin with @. It also cannot contain spaces or the following special characters: ` ~ % ^ * + = { } \ ; : ' " / ? . , < > ! () [] You cannot change the name of the service after you create it.
Description	Description of the service.
License	License object that allows use of the service.
Node	Node on which the service runs.

SAP BW Service Properties

The following table describes the SAP BW Service properties for PowerCenter:

Property	Description
SAP Destination R Type	DEST entry defined in the <code>sapnwrfc.ini</code> file for a connection to an RFC server program. Edit this property if you have created a different DEST entry in the <code>sapnwrfc.ini</code> file for the SAP BW Service.
Retry Period	Number of seconds the SAP BW Service waits before trying to connect to the SAP BW system if a previous connection failed. The SAP BW Service tries to connect five times. Between connection attempts, it waits the number of seconds you specify. After five unsuccessful attempts, the SAP BW Service shuts down. Default is 5 seconds.

The following table describes the SAP BW Service properties for the Developer tool:

Property	Description
Program ID	Program ID for the logical system you create in SAP BW for the SAP BW Service. The Program ID in SAP BW must match this parameter, including case.
Gateway Host	Host name of the SAP gateway.
Gateway Server	Server name of the SAP gateway.
SAP Connection	SAP connection. Specify a connection to a specific SAP application server or an SAP load balancing connection.
Trace	Use this option to track the JCo calls that the SAP system makes. SAP stores the information about the JCo calls in a trace file. Specify one of the following values: - 0. Off - 1. Full Default is 0. You can access the trace files from the following directory on the machine where you installed the Informatica services: <code><Informatica installation directory>/tomcat/bin</code>
Other Connection Parameters	Enter any other connection parameter that you want to use. Use the following format: <code><parameter name>=<value></code>
Retry Period	Number of seconds the SAP BW Service waits before trying to connect to the SAP BW system if a previous connection failed. The SAP BW Service tries to connect five times. Between connection attempts, it waits the number of seconds you specify. After five unsuccessful attempts, the SAP BW Service shuts down. Default is 5 seconds.

Configuring the Associated Integration Service

Use the Administrator tool to configure the associated Integration Service and connection information for the repository database. To read data from or write data to SAP BW, you must also configure a Workflow Orchestration Service for the Integration Service that is associated with the SAP BW Service.

1. Log in to the Administrator tool.
2. In the Domain Navigator, select the SAP BW Service.
3. Perform one of the following steps:
 - To configure an SAP BW Service for PowerCenter, click **Associated Integration Service**.
 - To configure an SAP BW Service for the Developer tool, click **Associated Data Integration Service**.
4. Click **Edit** and edit the following properties:

Property	Description
Associated Integration Service or Associated Data Integration Service	Name of the PowerCenter Integration Service or the Data Integration Service to which you want to associate the SAP BW Service.
Repository User Name	Account used to access the repository.
Repository Password	Password for the user. Note: If secure communication is enabled for the domain, you need not specify the repository password.
Security Domain	Security domain for the user. Appears when the Informatica domain contains an LDAP security domain.

5. Click **OK** to save the changes.

Configuring the SAP BW Service Processes

When you use PowerCenter to filter and load data to SAP BW, you can configure the temporary parameter file directory that the SAP BW Service must use.

1. Log in to the Administrator tool.
2. In the Domain Navigator, select the SAP BW Service.
3. Click **Processes**.
4. Click **Edit**.

5. Edit the following property:

Property	Description
ParamFileDir	<p>Temporary parameter file directory. The SAP BW Service stores SAP BW data selection entries in the parameter file when you filter data to load into SAP BW.</p> <p>The directory must exist on the node where the SAP BW Service runs. Verify that the directory you specify has read and write permissions enabled.</p> <p>The default directory is <Informatica installation directory>/services/shared/BWParam.</p>

Load Balancing for the SAP BW System and the SAP BW Service

You can configure the SAP BW system to use load balancing. To support an SAP BW system configured for load balancing, the SAP BW Service records the host name and system number of the SAP BW server requesting data from PowerCenter. The SAP BW Service passes this information to the PowerCenter Integration Service. The PowerCenter Integration Service uses this information to load data to the same SAP BW server that made the request. For more information about configuring the SAP BW system to use load balancing, see the SAP documentation.

You can also configure the SAP BW Service in PowerCenter to use load balancing. When you create the SAP BW Service, define an SAP load balancing connection. If the load on the SAP BW Service becomes too high, you can create multiple instances of the SAP BW Service to balance the load. To run multiple SAP BW Services configured for load balancing, create each service with a unique name but use the same values for all other parameters. The services can run on the same node or on different nodes. The SAP BW server distributes data to the multiple SAP BW Services in a round-robin fashion.

Viewing Log Events

The SAP BW Service sends log events to the Log Manager. The SAP BW Service captures log events that track interactions between PowerCenter and SAP BW. You can view SAP BW Service log events in the following locations:

- Administrator tool. On the **Logs** tab, enter search criteria to find log events that the SAP BW Service captures when extracting from or loading into SAP NetWeaver BI.
- SAP BW Monitor. In the Monitor - Administrator Workbench window, you can view log events that the SAP BW Service captures for an InfoPackage that is included in a process chain to load data into SAP BW. SAP BW pulls the messages from the SAP BW Service and displays them in the monitor. The SAP BW Service must be running to view the messages in the SAP BW Monitor.

To view log events about how the Integration Service processes an SAP BW workflow, view the session log or workflow log.

CHAPTER 18

Search Service

This chapter includes the following topics:

- [Search Service Overview, 338](#)
- [Search Service Architecture, 339](#)
- [Search Index, 339](#)
- [Search Request Process, 340](#)
- [Search Service Properties, 341](#)
- [Search Service Process Properties, 343](#)
- [Creating a Search Service, 344](#)
- [Enabling the Search Service, 344](#)
- [Recycling and Disabling the Search Service, 345](#)

Search Service Overview

The Search Service manages search in the Analyst tool and Business Glossary Desktop. By default, the Search Service returns search results from a Model repository, such as data objects, mapping specifications, profiles, reference tables, rules, and scorecards.

The Search Service can also return additional results. The results can include related assets, business terms, and policies. The results can include column profile results and domain discovery results from a profiling warehouse. In addition, you can perform a search based on patterns, datatypes, unique values, or null values.

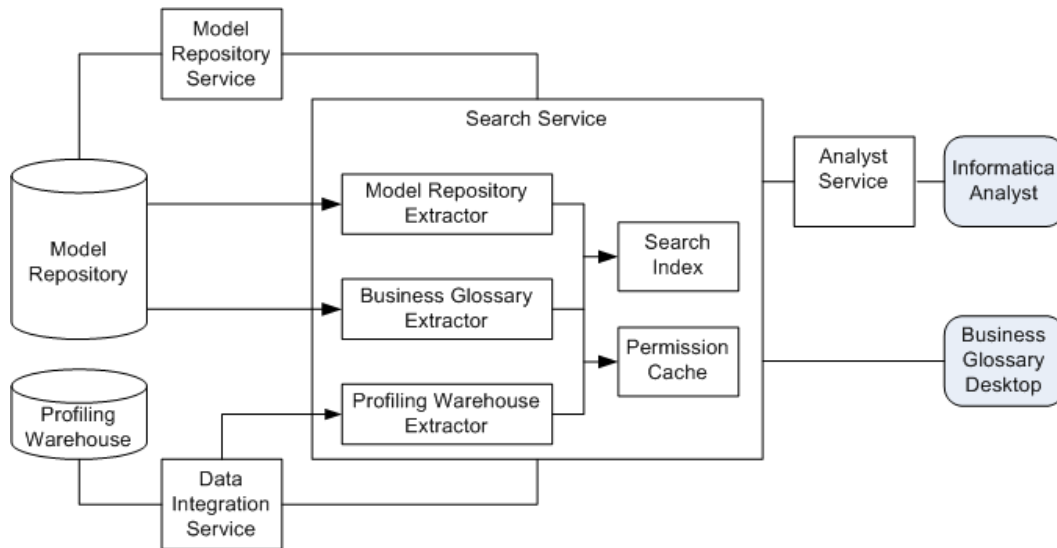
You can associate each Search Service with one Model repository and one profiling warehouse. To perform searches on multiple Model repositories or profiling warehouses, you must create multiple Search Services.

The Search Service performs each search on a search index instead of the Model repository or profiling warehouse. To create the search index, the Search Service extracts information about content from the Model repository and profiling warehouse. You can configure the interval at which the Search Service extracts this information. To enable faster searches, the Search Service indexes all extracted content.

Search Service Architecture

The Search Service interacts with different components in the Informatica domain when it builds the search index and returns search results. The Search Service can build a search index based on content in a Model repository and a profiling warehouse.

The following diagram shows the Informatica domain components with which the Search Service interacts:



When you create the Search Service, you specify the associated Model Repository Service. The Search Service determines the associated Data Integration Service based on the Model Repository Service.

To enable search across multiple repositories, the Search Service builds a search index based on content in one Model repository and one profiling warehouse. To enable search on multiple Model repositories or multiple profiling warehouses, create multiple Search Services.

The Search Service extracts content, including business glossary terms, from the Model repository associated with the Model Repository Service. The Search Service extracts column profile results and domain discovery results from the profiling warehouse associated with the Data Integration Service. The Search Service also extracts permission information to ensure that the user who submits a search request has permission to view each object returned in the search results. The Search Service stores the permission information in a permission cache.

Users can perform a search in the Analyst tool or Business Glossary Desktop. When a user performs a search in the Analyst tool, the Analyst Service submits the request to the Search Service. When a user performs a search in Business Glossary Desktop, Business Glossary Desktop submits the request to the Search Service. The Search Service returns results from the search index based on permissions in the permission cache.

Search Index

The Search Service performs each search on a search index instead of the Model repository or profiling warehouse. The search index enables faster searches and searches on content from the Model repository and profiling warehouse.

The Search Service generates the search index based on content in the Model repository and profiling warehouse. The Search Service contains extractors to extract content from each repository.

The Search Service contains the following extractors:

Model Repository extractor

Extracts content from a Model repository.

Business Glossary extractor

Extracts business glossary terms from the Model repository.

Profiling Warehouse extractor

Extracts column profiling results and domain discovery results from a profiling warehouse.

The Search Service indexes all content that it extracts. The Search Service maintains one search index for all extracted content. If a search index does not exist when the Search Service starts, the Search Service generates the search index.

During the initial extraction, the Search Service extracts and indexes all content. After the first extraction, the Search Service updates the search index based on content that has been added to, changed in, or removed from the Model repository and profiling warehouse since the previous extraction. You can configure the interval at which the Search Service generates the search index.

The Search Service extracts and indexes batches of objects. If it fails to extract or index an object, it retries again. After the third attempt, the Search Service ignores the object, writes an error message to the Search Service log, and then processes the next object.

The Search Service stores the search index in files in the extraction directory that you specify when you create the Search Service.

Extraction Interval

The Search Service extracts content based on the interval that you configure. You can configure the interval when you create the Search Service or update the service properties.

The extraction interval is the number of seconds between each extraction.

The Search Service returns search results from the search index. The search results depend on the extraction interval. For example, if you set the extraction interval to 360 seconds, a user may have to wait up to 360 seconds before an object appears in the search results.

Search Request Process

The Search Service processes search requests differently based on whether the request comes from the Analyst tool or Business Glossary Desktop.

The following steps describe the search request process:

1. A user enters search criteria in the Analyst tool or Business Glossary Desktop.
2. For a search in the Analyst tool, the corresponding Analyst Service sends the search request to the Search Service. For a search in Business Glossary Desktop, Business Glossary Desktop sends the search request to the Search Service.
3. The Search Service retrieves the search results from the search index based on the search criteria.
4. The Search Service verifies permissions on each search result and returns objects on which the user has read permission.

Note: The domain administrator must start the Search Service before the Search Service can return any search results. If the Search Service is not running when a user performs a search, an error appears.

Search Service Properties

When you create a Search Service, you configure the Search Service properties. You can edit the Search Service properties on the **Properties** tab in the Administrator tool.

You can configure the following types of Search Service properties:

- General properties
- Logging options
- Search options
- Custom properties

If you update any of the properties, recycle the Search Service for the modifications to take effect.

General Properties for the Search Service

General properties for the Search Service include the name and description of the Search Service, the node on which the Search Service runs, and the license associated with the Search Service.

You can configure the following general properties for the service:

Name

Name of the service. The name is not case sensitive and must be unique within the domain. It cannot exceed 128 characters or begin with @. It also cannot contain spaces or the following special characters:

` ~ % ^ * + = { } \ ; : ' " / ? . , < > | ! () []

You cannot change the name of the service after you create it.

Description

Description of the service. The description cannot exceed 765 characters.

License

License object that allows use of the service.

Node

Node on which the service runs.

Logging Options for the Search Service

The logging options include properties for the severity level for Search Service logs.

Configure the **Log Level** property to configure the level of error messages written to the Search Service log.

Choose one of the following message levels:

- Error. Writes ERROR code messages to the log. ERROR messages include connection failures, failures to save or retrieve metadata, service errors.
- Warning. Writes WARNING and ERROR code messages to the log. WARNING errors include recoverable system failures or warnings.
- Info. Writes INFO, WARNING, and ERROR code messages to the log. INFO messages include system and service change messages.

- Tracing. Writes TRACE, INFO, WARNING, and ERROR code messages to the log. TRACE messages log user request failures such as SQL request failures, mapping run request failures, and deployment failures.
- Debug. Writes DEBUG, TRACE, INFO, WARNING, and ERROR code messages to the log. DEBUG messages are user request logs.

Default is INFO.

Search Options for the Search Service

Search options for the Search Service include the port number, index location, extraction interval, and the Model repository details.

You can configure the following search options for the Search Service:

Port Number

Port on which the Search Service runs. Default is 8084.

Index Location

Directory that contains the search index files. Enter a directory on the machine that runs the Search Service. If the directory does not exist, Informatica creates the directory when it creates the Search Service.

Extraction Interval

Interval in seconds at which the Search Service updates the search index. Set to 60 seconds or more to enable the Search Service to complete an extraction and index before starting the next extraction. Default is 60 seconds. Minimum is 20 seconds.

Model Repository Service

Model Repository Service associated with the Model repository from which the Search Service extracts assets. A Model Repository Service appears only if it is not associated with a Search Service.

User Name

User name to access the Model repository. The Model repository user must have the Administrator role for the Model Repository Service. Not available for a domain with Kerberos authentication.

Password

An encrypted version of the user password to access the Model repository. Not available for a domain with Kerberos authentication.

Modify Password

Select to specify a different password than the one associated with the Model repository user. Select this option if the password changes for a user. Not available for a domain with Kerberos authentication.

Security Domain

LDAP security domain for the Model repository user. The field appears when the Informatica domain contains an LDAP security domain. Not available for a domain with Kerberos authentication.

Custom Properties for the Search Service

Configure custom properties that are unique to specific environments.

You might need to apply custom properties in special cases. When you define a custom property, enter the property name and an initial value. Define custom properties only at the request of Informatica Global Customer Support.

Search Service Process Properties

When you create a Search Service, you configure the Search Service process properties. You can edit the Search Service process properties on the **Processes** tab in the Administrator tool.

The Search Service runs the Search Service process on a node. When you select the Search Service in the Administrator tool, you can view the service processes for the Search Service on the **Processes** tab. You can view the node properties for the service process in the **Service** panel. You can view the service process properties in the **Service Process Properties** panel.

Note: You must select the node to view the service process properties in the **Service Process Properties** panel.

You can configure the following types of Search Service process properties:

- Advanced properties
- Environment variables
- Custom properties

If you update any of the process properties, restart the Search Service for the modifications to take effect.

Advanced Properties of the Search Service Process

Advanced properties include properties for the maximum heap size and the Java Virtual Manager (JVM) memory settings.

You can configure the following advanced properties for the Search Service process:

Maximum Heap Size

Amount of RAM allocated to the Java Virtual Machine (JVM) that runs the Search Service. Use this property to increase the performance. Append one of the following letters to the value to specify the units:

- b for bytes.
- k for kilobytes.
- m for megabytes.
- g for gigabytes.

Default is 768 megabytes. Specify 1 gigabyte if you run the Search Service on a 64-bit machine.

JVM Command Line Options

Java Virtual Machine (JVM) command line options to run Java-based programs.

You must set the following JVM command line options:

- -Dfile.encoding. File encoding. Default is UTF-8.
- -Xms. Minimum heap size. Default value is 256 m.
- -XX:MaxPermSize. Maximum permanent generation size. Default is 128 m.
- -XX:+HeapDumpOutOfMemoryError. Include option to write heap memory to a file if a java.lang.OutOfMemoryError error occurs.

Environment Variables for the Search Service Process

You can edit environment variables for the Search Service process.

You can define environment variables for the Search Service in the **Environment Variables** property.

Custom Properties for the Search Service Process

Configure custom properties that are unique to specific environments.

You might need to apply custom properties in special cases. When you define a custom property, enter the property name and an initial value. Define custom properties only at the request of Informatica Global Customer Support.

Creating a Search Service

Create the Search Service in the domain to enable searching in the Analyst tool and Business Glossary Desktop.

Before you create the Search Service, create the associated Model Repository Service, and Analyst Service. To enable search on objects in a profiling warehouse, create the Data Integration Service also.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. On the Domain Actions menu, click **New** > **Search Service**.
The **New Search Service - Step 1 of 2** window appears.
3. Enter the general properties for the service.
4. Optionally, click **Browse** in the **Location** field to select the location in the Navigator where you want to the service to appear.
The **Select Folder** dialog box appears.
5. Optionally, click **Create Folder** to create another folder.
6. Click **OK**.
The **Select Folder** dialog box closes.
7. Click **Next**.
The **New Search Service - Step 2 of 2** window appears.
8. Enter the search options for the service.
9. Click **Finish**.

Enabling the Search Service

Enable the Search Service to enable search in the Analyst tool and Business Glossary Desktop.

Before you enable the Search Service, verify that you enabled the Model Repository Service, Data Integration Service, and the Analyst Service.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator of the Administrator tool, select the Search Service.
3. Click the **Enable Service** button.
The Search Service starts.

Recycling and Disabling the Search Service

Disable the Search Service to perform maintenance or temporarily restrict users from performing searches in the associated Analyst tool or Business Glossary Desktop. Recycle the Search Service to restart the Search Service and apply the latest service and service process properties.

Before you recycle the Search Service, verify that you enabled the Model Repository Service, Data Integration Service, and the Analyst Service.

You must recycle the Search Service when you change the user name or password of the Model Repository Service or associate a different Model Repository Service with the Search Service. You must also recycle the Search Service when you update any of the Search Service properties or Search Service process properties.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator of the Administrator tool, select the Search Service.
3. Click the **Disable the Service** button or the **Recycle the Service** button.

The **Disable Service** or **Recycle Service** dialog box appears.

4. Select the shut down mode for the Search Service.

Select one of the following modes:

- **Complete.** Runs jobs to completion before disabling or recycling the service.
- **Stop.** Waits up to 30 seconds to complete jobs that are running before disabling or recycling the service.
- **Abort.** Tries to stop all jobs before aborting them and disabling or recycling the service.

CHAPTER 19

System Services

This chapter includes the following topics:

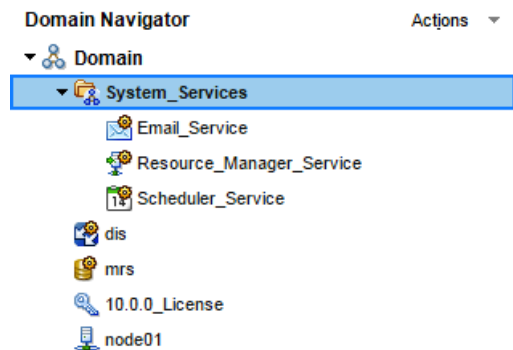
- [System Services Overview, 346](#)
- [Email Service, 347](#)
- [Resource Manager Service, 350](#)
- [Scheduler Service, 353](#)

System Services Overview

A system service is an application service that can have a single instance in the domain. When you create the domain, the system services are created for you. You can enable, disable, and configure system services.

System services are created in the System Services folder. Expand the System Services folder in the Domain Navigator to view and configure the system services. You cannot delete, move, or edit the properties or contents of the System Services folder.

The following image shows the System Services folder in the Domain Navigator:



By default, system services are disabled and are assigned to run on the master gateway node. You can change the node assignment and enable the service to use the functionality that the service provides.

The domain includes the following system services:

Email Service

The Email Service sends email notifications for business glossaries, scorecards, and workflows. Enable the Email Service to allow users to configure email notifications.

Resource Manager Service

The Resource Manager Service manages computing resources in the domain and dispatches jobs to achieve optimal performance and scalability. The Resource Manager Service collects information about nodes with the compute role. The service matches job requirements with resource availability to identify the best compute node to run the job.

The Resource Manager Service communicates with compute nodes in a Data Integration Service grid. Enable the Resource Manager Service when you configure a Data Integration Service grid to run jobs in separate remote processes.

Scheduler Service

The Scheduler Service manages schedules for profiles, scorecards, deployed mappings, and deployed workflows.

Email Service

The Email Service sends email notifications for business glossaries, scorecards, and workflows. Enable the Email Service to allow users to configure email notifications.

The Email Service sends emails for the following notifications:

- Business glossary notifications.
- Scorecard notifications.
- Workflow notifications. Workflow notifications include emails sent from Human tasks and Notification tasks in workflows that the Data Integration Service runs.

The Email Service is associated with a Model Repository Service. The Model repository stores metadata for the email notifications that users configure. Both the Model Repository Service and the Email Service must be available for the Email Service to send email notifications.

The Email Service is highly available. High availability enables the Service Manager and the Email Service to react to network failures and failures of the Email Service. The Email Service has the restart and failover high availability feature. If a Email Service becomes unavailable, the Service Manager can restart the service on the same node or on a back-up node.

Before You Enable the Email Service

Before you enable the Email Service, complete the prerequisite tasks for the service.

Perform the following tasks before you enable the Email Service:

- If the domain uses Kerberos authentication and you set the service principal level at the process level, create a keytab file for the service. For more information about creating the service principal names and keytab files, see the *Informatica Security Guide*.
- Configure the Model repository options for the service.
- Configure the email server properties.

Email Service Properties

You can configure general properties, Model Repository Service options, and email server properties for the Email Service. To configure the Email Service properties, select the service in the Domain Navigator and click

Edit in the **Properties** view. You can change the properties while the service is running, but you must recycle the service for the changed properties to take effect.

General Properties

The following table describes the general properties for the service:

Property	Description
Name	Name of the service. You cannot change the name of the Email Service.
Description	Description of the service. The description cannot exceed 765 characters.
Node	Node on which the service runs.
Backup Nodes	Nodes on which the service can run if the primary node is unavailable.

Model Repository Service Options

Configure a Model repository to store metadata for the email notifications that users configure. The Model Repository Service must be available for the Email Service to send email notifications.

If the Model repository is integrated with a version control system, then you must synchronize the repository before you associate it with the Email Service.

The following table describes the Model Repository options for the service:

Property	Description
Model Repository Service	Model Repository Service associated with the Email Service.
Username	User name of an administrator user in the Informatica domain. Not available for a domain with Kerberos authentication.
Password	Password of the administrator user in the Informatica domain. Not available for a domain with Kerberos authentication.

Email Server Properties

Configure the email server properties so Business Glossary and Data Quality users can configure email notifications.

The Email Service uses the email server configuration to send the following notifications:

- Business glossary notifications.
- Scorecard notifications.
- Workflow notifications. Workflow notifications include emails sent from Human tasks and Notification tasks in workflows that the Data Integration Service runs.

The following table describes the email server properties for the service:

Property	Description
SMTP Server Host Name	The SMTP outbound mail server host name. For example, enter the Microsoft Exchange Server for Microsoft Outlook. Default is localhost.
SMTP Server Port	Port number used by the outbound SMTP mail server. Valid values are from 1 to 65535. Default is 25.
SMTP Server User Name	User name for authentication upon sending, if required by the outbound SMTP mail server.
SMTP Server Password	Password for authentication upon sending, if required by the outbound SMTP mail server.
SMTP Authentication Enabled	Indicates that the SMTP server is enabled for authentication. If true, the outbound mail server requires a user name and password. Default is false.
Use TLS Security	Indicates that the SMTP server uses the TLS protocol. If true, enter the TLS port number for the SMTP server port property. Default is false.
Use SSL Security	Indicates that the SMTP server uses the SLL protocol. If true, enter the SSL port number for the SMTP server port property. Default is false.
Sender Email Address	Email address that the Email Service uses in the From field when sending notification emails from a workflow. Default is <code>admin@example.com</code> .

Email Service Process Properties

When the Email Service is configured to run on primary and back-up nodes, a service process is enabled on each node. Only a single process runs at any given time, and the other processes maintain standby status. You can view the state of the service process on each node on the **Processes** view.

You can view the following information about the Email Service process:

- **Process Configuration.** The state of the process configured to run on the node. The state can be Enabled or Disabled.
- **Process State.** The state of the service process running on the node. The state can be Enabled or Disabled.
- **Node.** The node that the service process runs on.
- **Node Role.** Indicates whether the node has the service role, the compute role, or both roles.
- **Node Status.** The state of the node that the process is running on. The state can be Enabled or Disabled.

Enabling, Disabling, and Recycling the Email Service

You can enable, disable, and recycle the Email Service from the Administrator tool.

By default, the Email Service is disabled. Enable the Email Service when you need to allow users to generate emails based on Human tasks in a workflow or changes to the Business Glossary. When you enable the Email

Service, a service process starts on the node designated to run the service. The service is available to send emails based on the notification properties that users configure.

You might disable the Email Service if you need to perform maintenance. You might recycle the Email Service if you connect to a different Model Repository Service.

When you recycle or disable an Email Service, you must choose a mode to recycle or disable it in. You can choose one of the following options:

- Complete. Wait for all subtasks to complete.
- Stop. Wait up to 30 seconds for all subtasks to complete.
- Abort. Stop all processes immediately.

Optionally, you can choose to specify whether the action was planned or unplanned, and enter comments about the action. If you complete these options, the information appears in the **Events** and **History** panels in the **Domain** view on the **Manage** tab.

To enable the service, select the service in the Domain Navigator and click **Enable the Service**.

To disable the service, select the service in the Domain Navigator and click **Disable the Service**.

To recycle the service, select the service in the Domain Navigator and click **Recycle the Service**. When you recycle the service, the Service Manager restarts the service. You must recycle the Email Service whenever you change a property for the service.

Resource Manager Service

The Resource Manager Service manages computing resources in the domain and dispatches jobs to achieve optimal performance and scalability. The Resource Manager Service collects information about nodes with the compute role. The service matches job requirements with resource availability to identify the best compute node to run the job.

The Resource Manager Service communicates with compute nodes in a Data Integration Service grid. Enable the Resource Manager Service when you configure a Data Integration Service grid to run jobs in separate remote processes. The Resource Manager Service does not require a license object before you enable the service.

The Resource Manager Service is highly available. High availability enables the Service Manager and the Resource Manager Service to react to network failures and failures of the Resource Manager Service. The Resource Manager Service has the restart and failover high availability feature. If a Resource Manager Service becomes unavailable, the Service Manager can restart the service on the same node or on a back-up node.

Resource Manager Service Architecture

The Resource Manager Service connects to nodes with the compute role in a Data Integration Service grid that is configured to run jobs in separate remote processes.

When the Service Manager on a node with the compute role starts, the Service Manager registers the node with the Resource Manager Service. Compute nodes use a heartbeat protocol to send periodic signals to the Resource Manager Service. The Resource Manager Service stores compute node details in memory. If the node stops sending heartbeat signals, the Resource Manager Service marks the node as unavailable and does not dispatch jobs to the node.

When you enable a Data Integration Service that runs on the grid, the Data Integration Service designates one node with the compute role as the master compute node. The Service Manager on the master compute node communicates with the Resource Manager Service to find an available worker compute node to run job requests.

Before You Enable the Resource Manager Service

Before you enable the Resource Manager Service, complete the prerequisite tasks for the service.

Before you enable the Resource Manager Service, configure a Data Integration Service grid to run jobs in separate remote processes. The designated master compute node in the grid communicates with the Resource Manager Service to find an available compute node to run jobs remotely.

Resource Manager Service Properties

To configure the Resource Manager Service properties, select the service in the Domain Navigator and click the **Properties** view. You can change the properties while the service is running, but you must recycle the service for the changed properties to take effect.

General Properties

In the general properties, configure the primary and back-up nodes for the Resource Manager Service.

The following table describes the general properties for the service:

Property	Description
Name	Name of the service. You cannot change the name of the Resource Manager Service.
Description	Description of the service. The description cannot exceed 765 characters.
Node	Node on which the service runs.
Backup Nodes	Nodes on which the service can run if the primary node is unavailable.

Logging Options

The following table describes the log level property for the Resource Manager Service:

Property	Description
Log Level	Determines the default severity level for the service logs. Choose one of the following options: <ul style="list-style-type: none">- Fatal. Writes FATAL messages to the log. FATAL messages include nonrecoverable system failures that cause the service to shut down or become unavailable.- Error. Writes FATAL and ERROR code messages to the log. ERROR messages include connection failures, failures to save or retrieve metadata, service errors.- Warning. Writes FATAL, WARNING, and ERROR messages to the log. WARNING errors include recoverable system failures or warnings.- Info. Writes FATAL, INFO, WARNING, and ERROR messages to the log. INFO messages include system and service change messages.- Trace. Write FATAL, TRACE, INFO, WARNING, and ERROR code messages to the log. TRACE messages log user request failures.- Debug. Write FATAL, DEBUG, TRACE, INFO, WARNING, and ERROR messages to the log. DEBUG messages are user request logs.

Resource Manager Service Process Properties

When the Resource Manager Service is configured to run on primary and back-up nodes, a service process is enabled on each node. Only a single process runs at any given time, and the other processes maintain standby status. You can configure the service process properties differently for each node.

To configure the Resource Manager Service process properties, select the service in the Domain Navigator and click the **Processes** view. You can change the properties while the service is running, but you must restart the service process for the changed properties to take effect.

Environment Variables

You can configure environment variables for the Resource Manager Service process.

The following table describes the environment variables:

Property	Description
Environment Variable	Enter a name and a value for the environment variable.

Advanced Options

The following table describes the advanced options:

Property	Description
Maximum Heap Size	Amount of RAM allocated to the Java Virtual Machine (JVM) that runs the service process. Use this property to increase the performance. Append one of the following letters to the value to specify the units: <ul style="list-style-type: none">- b for bytes.- k for kilobytes.- m for megabytes.- g for gigabytes.
JVM Command Line Options	Java Virtual Machine (JVM) command line options to run Java-based programs. When you configure the JVM options, you must set the Java SDK classpath, Java SDK minimum memory, and Java SDK maximum memory properties. You must set the following JVM command line options: <ul style="list-style-type: none">- Xms. Minimum heap size. Default value is 256 m.- MaxPermSize. Maximum permanent generation size. Default is 128 m.- Dfile.encoding. File encoding. Default is UTF-8.

Enabling, Disabling, and Recycling the Resource Manager Service

You can enable, disable, and recycle the Resource Manager Service from the Administrator tool.

By default, the Resource Manager Service is disabled. Enable the Resource Manager Service when you configure a Data Integration Service grid to run jobs on remote nodes with the compute role. When you enable the Resource Manager Service, a service process starts on the node designated to run the service. The service is available to manage computing resources in the domain.

You might disable the Resource Manager Service if you need to perform maintenance or you need to temporarily prevent Data Integration Service jobs from remotely running on nodes with the compute role. You might recycle the Resource Manager Service if you changed a property. When you recycle the service, the Service Manager restarts the service.

When you disable a Resource Manager Service, you must choose the mode to disable it in. You can choose one of the following options:

- Complete. Wait for all processes to complete.
- Abort. Stop all processes immediately.

Optionally, you can choose to specify whether the action was planned or unplanned, and enter comments about the action. If you complete these options, the information appears in the **Events** and **Command History** panels in the **Domain** view on the **Manage** tab.

To enable the service, select the service in the Domain Navigator and click **Enable the Service**.

To disable the service, select the service in the Domain Navigator and click **Disable the Service**.

To recycle the service, select the service in the Domain Navigator and click **Recycle the Service**.

Note: If the Resource Manager Service is configured to run on primary and back-up nodes, you can enable or disable a Resource Manager Service process on the **Processes** view. Disabling a service process does not disable the service. Disabling a service process that is running causes the service to fail over to another node.

Scheduler Service

The Scheduler Service manages schedules for profiles, scorecards, deployed mappings, and deployed workflows.

Use schedules to run deployed mappings and workflows at a specified time. You can schedule the objects to run one time, or on an interval. Enable the Scheduler Service to create, manage, and run schedules.

The Scheduler Service is associated with a Model Repository Service. The Model repository stores metadata for the schedules that users configure. Both the Model Repository Service and the Scheduler Service must be available for scheduled objects to run.

The Scheduler Service is highly available. High availability enables the Service Manager and the Scheduler Service to react to network failures and failures of the Scheduler Service. The Scheduler Service has the restart and failover high availability feature. If a Scheduler Service becomes unavailable, the Service Manager can restart the service on the same node or on a back-up node.

Before You Enable the Scheduler Service

Before you enable the Scheduler Service, complete the prerequisite tasks for the service.

Before you enable the Scheduler Service, complete the following tasks:

- If the domain uses Kerberos authentication and you set the service principal level at the process level, create a keytab file for the service. For more information about creating the service principal names and keytab files, see the *Informatica Security Guide*.
- Configure a Model repository for the service.

Scheduler Service Properties

You can configure general properties, logging options, and a Model Repository Service for the Scheduler Service. To configure the Scheduler Service properties, select the service in the Domain Navigator and click

Edit in the **Properties** view. You can change the properties while the service is running, but you must recycle the service for the modifications to take effect.

General Properties

The following table describes the general properties for the service:

Property	Description
Name	Name of the service. You cannot change the name of the Scheduler Service.
Description	Description of the service. The description cannot exceed 765 characters.
Node	Node on which the service runs.
Backup Nodes	Nodes on which the service can run if the primary node is unavailable.

Logging Options

Configure the Logging Level property to determine the level of error messages that are written to the Scheduler Service log.

The following table describes the logging level properties for the service:

Property	Description
Logging Level	Determines the default severity level for the service logs. Choose one of the following options: <ul style="list-style-type: none">- Fatal. Writes FATAL messages to the log. FATAL messages include nonrecoverable system failures that cause the service to shut down or become unavailable.- Error. Writes FATAL and ERROR code messages to the log. ERROR messages include connection failures, failures to save or retrieve metadata, service errors.- Warning. Writes FATAL, WARNING, and ERROR messages to the log. WARNING errors include recoverable system failures or warnings.- Info. Writes FATAL, INFO, WARNING, and ERROR messages to the log. INFO messages include system and service change messages.- Trace. Write FATAL, TRACE, INFO, WARNING, and ERROR code messages to the log. TRACE messages log user request failures.- Debug. Write FATAL, DEBUG, TRACE, INFO, WARNING, and ERROR messages to the log. DEBUG messages are user request logs.

Model Repository Service Options

Configure a Model repository to store information about the schedules. The Model Repository Service must be available for the Scheduler Service to run scheduled objects.

If the Model repository is integrated with a version control system, synchronize the Model repository before you associate it with the Scheduler Service.

The following table describes the Model repository options for the service:

Property	Description
Model Repository Service	Model Repository Service associated with the Scheduler Service.
Username	User name of an administrator user in the Informatica domain. Not available for a domain with Kerberos authentication.
Password	Password of the administrator user in the Informatica domain. Not available for a domain with Kerberos authentication.
Security Domain	LDAP security domain for the user who manages the Scheduler Service. The security domain field does not appear for users with Native or Kerberos authentication.

Storage Properties

Configure a temporary file location when you configure the Scheduler Service to run on multiple nodes. Use the temporary file location to store parameter files for deployed mappings and workflows. The file location must be a directory that all of the nodes can access.

The following table describes the Temporary File Location property:

Property	Description
Temporary File Location	Path to the directory where parameter files are read from and written to.

Scheduler Service Process Properties

When the Scheduler Service is configured to run on primary and back-up nodes, a service process is enabled on each node. Only a single process runs at any given time, and the other processes maintain standby status. You can configure the service process properties differently for each node.

To configure the Scheduler Service process properties, select the service in the Domain Navigator and click the **Processes** view. You can change the properties while the service is running, but you must restart the service process for the changed properties to take effect.

Security Properties

When you set the HTTP protocol type for the Scheduler Service to HTTPS or both, you enable the Transport Layer Security (TLS) protocol for the service. Depending on the HTTP protocol type of the service, you define the HTTP port, the HTTPS port, or both ports for the service process.

The following table describes the Scheduler Service security properties:

Property	Description
HTTP Port	Unique HTTP port number for the Scheduler Service process when the service uses the HTTP protocol. Default is 6211.
HTTPS Port	Unique HTTPS port number for the Scheduler Service process when the service uses the HTTPS protocol. When you set an HTTPS port number, you must also define the keystore file that contains the required keys and certificates.

HTTP Configuration Options

Configure the HTTP options when the Scheduler Service uses the HTTPS protocol.

The following table describes the HTTP configuration options:

Property	Description
Keystore File	Path and file name of the keystore file that contains the keys and certificates. Required if you use HTTPS connections for the service. You can create a keystore file with a keytool. Keytool is a utility that generates and stores private or public key pairs and associated certificates in a keystore file. You can use the self-signed certificate or use a certificate signed by a certificate authority.
Keystore Password	Password for the keystore file.
Truststore File	Path and file name of the truststore file that contains authentication certificates trusted by the service.
Truststore Password	Password for the keystore file.
SSL Protocol	Secure Sockets Layer protocol to use. Default is TLS.

Advanced Options

You can configure maximum heap size and JVM command line options for the Scheduler Service.

The following table describes the advanced options:

Property	Description
Maximum Heap Size	Amount of RAM allocated to the Java Virtual Machine (JVM) that runs the service process. Use this property to increase the performance. Append one of the following letters to the value to specify the units: <ul style="list-style-type: none">- b for bytes.- k for kilobytes.- m for megabytes.- g for gigabytes.
JVM Command Line Options	Java Virtual Machine (JVM) command line options to run Java-based programs. When you configure the JVM options, you must set the Java SDK classpath, Java SDK minimum memory, and Java SDK maximum memory properties. You must set the following JVM command line options: <ul style="list-style-type: none">- Xmx. Maximum heap size. Default value is 640 m.- Xms. Minimum heap size. Default value is 256 m.- MaxPermSize. Maximum permanent generation size. Default is 192 m.- Dfile.encoding. File encoding. Default is UTF-8.

Environment Variables

You can configure environment variables for the Scheduler Service process.

The following table describes the environment variables:

Property	Description
Environment Variable	Enter a name and a value for the environment variable.

Enabling, Disabling, and Recycling the Scheduler Service

You can enable, disable, and recycle the Scheduler Service from the Administrator tool.

By default, the Scheduler Service is disabled. Enable the Scheduler Service when you want to manage schedules or run scheduled objects. When you enable the Scheduler Service, a service process starts on the node designated to run the service. The service is available to schedule and run objects.

You might disable the Scheduler Service for maintenance, or recycle the service if you change a property.

When you recycle or disable a Scheduler Service, you must choose a mode to recycle or disable it in. You can choose one of the following modes:

- Complete. Wait for all subtasks to complete.
- Stop. Wait up to 30 seconds for all subtasks to complete.
- Abort. Stop all processes immediately.

Optionally, you can choose to specify whether the action is planned or unplanned, and enter comments about the action. If you complete these options, then the information appears in the service **Events** and **Command History** panels in the **Domain** view on the **Manage** tab.

To enable the service, select the service in the Domain Navigator and click **Enable the Service**.

To disable the service, select the service in the Domain Navigator and click **Disable the Service**.

To recycle the service, select the service in the Domain Navigator and click **Recycle the Service**. When you recycle the service, the Service Manager restarts the service. You must recycle the Scheduler Service whenever you change a property for the service.

CHAPTER 20

Test Data Manager Service

This chapter includes the following topics:

- [Test Data Manager Service Overview , 359](#)
- [Test Data Manager Service Dependencies, 360](#)
- [Test Data Manager Service Properties, 360](#)
- [Database Connection Strings, 364](#)
- [Configuring the Test Data Manager Service, 364](#)
- [Creating the Test Data Manager Service, 365](#)
- [Enabling and Disabling the Test Data Manager Service, 365](#)
- [Editing the Test Data Manager Service, 366](#)
- [Deleting the Test Data Manager Service, 367](#)

Test Data Manager Service Overview

The Test Data Manager Service is an application service in the Informatica domain. Test Data Manager uses the Test Data Manager Service to perform data masking, data discovery, data subset, and test data generation tasks. Test Data Manager connects to the Test Data Manager Service and uses the database content from the TDM repository associated with the service. The TDM repository is a relational database that contains tables that TDM requires to run and the tables that store metadata about data sources.

Create a Test Data Manager Service in the Informatica domain to use Test Data Manager. Use the Administrator tool or the infacmd command line program to administer the Test Data Manager Service.

Test Data Manager Service Dependencies

The Test Data Manager Service depends on PowerCenter services and requires other application services to perform tasks. Before you create the Test Data Manager Service, you must create the services that it depends on.

PowerCenter Services

Create the PowerCenter services that the Test Data Manager Service depends on in the following order:

1. PowerCenter Repository Service
Test Data Manager requires this service to access metadata stored in the PowerCenter repository.
2. PowerCenter Integration Service
Test Data Manager requires this service to run workflows and sessions.

Application Services

Create the application services that the Test Data Manager Service depends on in the following order:

1. Model Repository Service
Test Data Manager requires this service to perform data discovery.
2. Data Integration Service
Test Data Manager requires this service to perform data discovery.
3. Analyst Service
Test Data Manager requires this service to link TDM objects with terms in the Business Glossary.
4. Test Data Warehouse Service
Test Data Manager requires this service to create and store data sets in the test data warehouse.

Create the services before you create the Test Data Manager Service.

Test Data Manager Service Properties

To view the Test Data Manager Service properties, select the service in the Domain Navigator and click the Properties view. You can configure the following Test Data Manager Service properties:

- General properties
- Service properties
- TDM repository configuration properties
- TDM server configuration properties
- Advanced properties

If you update a property, restart the Test Data Manager Service to apply the update.

General Properties

The following table describes the general properties for the service:

Property	Description
Name	Name of the service. The name is not case sensitive and must be unique within the domain. It cannot exceed 128 characters or begin with @. It also cannot contain spaces or the following special characters: ` ~ % ^ * + = { } \ ; : ' " / ? . , < > ! () [] You cannot change the name of the service after you create it.
Description	Description of the service. The description cannot exceed 765 characters.
Location	Domain and folder where the service is created. Click Browse to choose a different folder. You can move the service after you create it.
License	License object that allows use of the service.
Node	Node on which the service runs.

Service Properties

The following table describes the service properties that you configure for the Test Data Manager Service:

Property	Description
PowerCenter Repository Service	The PowerCenter Repository Service that the Test Data Manager Service uses to load metadata into the TDM repository.
PowerCenter Integration Service	The PowerCenter Integration Service that runs the workflows that you generate in Test Data Manager for TDM operations.
Model Repository Service	The Model Repository Service associated with the Test Data Manager Service.
User Name	The user name that the Test Data Manager Service uses to connect to the Model Repository Service.
Password	The password that the Test Data Manager Service uses to connect to the Model Repository Service.
Security Domain	The name of the security domain that the user belongs to. Select the security domain from the list.
Data Integration Service	The Data Integration Service associated with the Test Data Manager Service. The Data Integration Service runs the workflows that you generate when you perform data discovery operations in Test Data Manager. If you have enabled profiling, or if you use Hadoop connections, you must select the Data Integration Service in the domain.

Property	Description
Analyst Service	The Analyst Service associated with the Test Data Manager Service. The Analyst Service connects to the Analyst tool, a flat file cache directory to store uploaded flat files and a business glossary export file directory. Required if you want to link TDM global objects to Business Glossary assets.
Test Data Warehouse Service	The Test Data Warehouse Service associated with the Test Data Manager Service. The Test Data Warehouse Service manages the test data warehouse repository. Required if you want to create and store data sets in the test data warehouse.

TDM Repository Configuration Properties

The following table describes the TDM repository configuration properties that you configure for the Test Data Manager Service:

Property	Description
Database Type	Type of database for the TDM repository. <ul style="list-style-type: none"> - Oracle - Microsoft SQL Server - DB2 Note: If you use a Microsoft SQL Server database, you must set the collation to <i>case insensitive</i> on the database.
Use Trusted Connection	Available for Microsoft SQL Server. Select this if you want to log in using Windows login credentials.
Custom Driver Class	Custom JDBC parameters. Required if you select Custom database type. Enter the custom JDBC driver parameters.
Username	User account for the TDM repository database. Set up this account using the appropriate database client tools. To apply changes, restart the Test Data Manager Service.
Password	Password for the TDM repository database. Must be in 7-bit ASCII. To apply changes, restart the Test Data Manager Service.
JDBC URL	JDBC connection URL used to access the TDM repository database. Enter the JDBC URL in the following format: <ul style="list-style-type: none"> - Oracle: jdbc:informatica:oracle://<host name>:<port>;ServiceName=<service name> - IBM DB2: jdbc:informatica:db2://<host name>:<port>;DatabaseName=<database name> - Microsoft SQL Server: jdbc:informatica:sqlserver://<host name>:<port>;DatabaseName=<database name>
Connection String	Native connect string to the TDM repository database. The Test Data Manager Service uses the connect string to create a connection object to the TDM repository and the PowerCenter repository. To apply changes, restart the Test Data Manager Service.
Schema Name	Available for Microsoft SQL Server. Name of the schema for the database. If not selected, the service creates the tables in the default schema.

Property	Description
Tablespace Name	Available for DB2. Name of the tablespace in which to create the tables. You must define the tablespace on a single node and the page size must be 32 KB. In a multipartition database, you must select this option. In a single-partition database, if you do not select this option, the installer creates the tables in the default tablespace.
Creation options for the New Test Data Manager Service	<p>Options to create content, or use existing content, and upgrade existing content.</p> <ul style="list-style-type: none"> - Do not create new content. Creates the repository without creating content. Select this option if the database content exists. If the content is of a previous version, the service prompts you to upgrade the content to the current version. - Previous Test Data Manager Service Name: Enter the name of the previous Test Data Manager Service. Required if you create the service with a different name. <p>Note: If you create the Test Data Manager Service with a different name, the source and target connections do not appear in Test Data Manager. Import the connections again if the connections do not appear in Test Data Manager.</p> <ul style="list-style-type: none"> - Upgrade TDM Repository Contents. Upgrades the content to the current version. - Create new content. Creates repository content.

TDM Server Configuration Properties

The following table describes the TDM Server configuration properties that you configure for the Test Data Manager Service:

Property	Description
HTTP Port	Port number that TDM runs on. The default is 6605.
Enable Transport Layer Security (TLS)	Secures communication between the Test Data Manager Service and the domain.
HTTPS Port	Port number for the HTTPS connection. The default is 6643.
Keystore File	Path and file name of the keystore file. The keystore file contains the keys and certificates required if you use the SSL security protocol with the Test Data Manager application. Required if you select Enable Transport Layer Security.
Keystore Password	Password for the keystore file. Required if you select Enable Secured Socket Layer.
SSL Protocol	Secure Sockets Layer protocol to use. Default is TLS.

Advanced Properties

The following table describes the advanced properties that you can configure for the Test Data Manager Service:

Property	Description
JVM Params	The heap size allocated for Test Data Manager. - Xms512m - Xmx1024m -XX:MaxPermSize=512m The time after which database connections are renewed if the Test Data Manager remains idle. Required if you modified the database configuration settings to values less than the TDM defaults. Configure the following values in TDM to be less than the database values. - IDLE_TIME. -DIDLE_TIME=<seconds>. Default is 300 seconds. - CONNECT_TIME. -DCONNECT_TIME=<seconds>. Default is 5000 seconds.
Connection Pool Size	The JDBC connection pool size.
JMX Port	Port number for the JMX/RMI connections to TDM. Default is 6675.
Shutdown Port	Port number that controls the server shutdown for TDM. The TDM Server listens for shutdown commands on this port. Default is 6607.

Database Connection Strings

When you create a database connection, specify a connection string for that connection. The Test Data Manager Service uses the connection string to create a connection object to the Test Data Manager repository.

The following table lists the native connect string syntax for each supported database:

Database	Connection String Syntax	Example
IBM DB2	<i>dbname</i>	mydatabase
Microsoft SQL Server	<i>servername@dbname</i>	sqlserver@mydatabase
Oracle	<i>dbname.world</i> (same as TNSNAMES entry)	oracle.world

Configuring the Test Data Manager Service

You can create and configure a Test Data Manager Service in the Administrator tool.

1. Set up the TDM repository database. You enter the database information when you create the Test Data Manager Service.
2. Create a PowerCenter Repository Service, PowerCenter Integration Service, and Model Repository Service.

3. Optional. Create a Data Integration Service. Required if you use the data profiling feature or if you use Hadoop connections in TDM.
4. Optional. Create an Analyst Service. Required if you use the asset linking feature. The Analyst Service license must support Business Glossary.
5. Create the Test Data Manager Service and configure the service properties.
6. Enable the Test Data Manager Service in the Informatica domain.

Creating the Test Data Manager Service

Log in to the Administrator tool to create the Test Data Manager Service. You can also create the Test Data Manager Service using the TDM command line program.

1. In the Administrator tool, click the **Domain** tab.
2. Click the **Services and Nodes** view.
3. Click **Actions > New > Test Data Manager Service**.
The **New Test Data Manager Service** dialog box appears.
4. Enter values for the general properties, and click **Next**.
5. Enter values for the service properties, and click **Next**.
6. Enter the repository configuration properties and test the connection. The repository connection information must be valid for the service to work.
 - a. If no content exists, select **Create new content**. You cannot select this option if the database has content.
 - b. If the database content exists, select **Do not create new content**. If you entered a different name for the Test Data Manager Service, you are prompted to enter the name of the previous Test Data Manager Service. The application checks the version of the content. If the content is of a previous version, an option to upgrade the repository content appears. Upgrade the repository content. Creating the service without upgrading the content to the current version generates a warning.
7. Choose to enable the Test Data Manager Service, and click **Next**.
8. Enter values for the server configuration properties, and click **Next**.
9. Enter values for the advanced properties, and click **Finish**.

Enabling and Disabling the Test Data Manager Service

You can enable, disable, and recycle the Test Data Manager Service from the service **Actions** menu in the Administrator tool. You can also use the tdm command line program to enable and disable the service.

Disable a Test Data Manager Service to perform maintenance or to temporarily restrict users from accessing Test Data Manager. When you disable the Test Data Manager Service, you also stop Test Data Manager. You might recycle the service if you update a property. When you recycle the service, the Service Manager disables and enables the service.

When you enable the Test Data Manager Service, the Service Manager starts TDM on the node where the service runs.

Editing the Test Data Manager Service

You can edit the Test Data Manager Service from the Administrator tool or using the `tdm` command line program.

Edit the Test Data Manager Service to create or upgrade content and to edit or update the service properties.

Create or Upgrade TDM Repository Content

You can edit the TDM Service to create repository content after saving the service. If the TDM repository content is of an older version, you can edit the TDM Service to upgrade the content.

1. Log in to the Informatica Administrator as an Administrator.
2. Select the TDM Service in the Domain Navigator to open the service properties.
Warning messages appear if the repository content is of an older version or if the content does not exist.
3. Click **Actions > Create Contents** to create content, or click **Actions > Upgrade Contents** to upgrade repository content.

Assigning the Test Data Manager Service to a Different Node

You can assign the Test Data Manager Service to a different node in the domain. The new node that uses the Test Data Manager Service must have TDM installed.

1. Disable the Test Data Manager Service.
2. Click **Edit** in the **General Properties** section.
3. Select a different node for the Node property, and then click **OK**.
4. If the Test Data Manager Service is running in HTTPS security mode, change the Keystore File Location to the path on the new node. Click **Edit** in the **Server Configuration Properties** section and update the Keystore File location, and click **OK**.
5. Enable the Test Data Manager Service.

Assigning a New License to the Test Data Manager Service

If you buy additional licenses, you can assign a different license to the Test Data Manager Service. Unassign the Test Data Manager Service from the existing license and then assign the service to the new license. You must add the license to the domain before you can assign it to the Test Data Manager Service.

Add the new license to the domain from the Domain **Actions > New > License** option.

To assign a new license to the Test Data Manager Service, perform the following steps in the Administrator tool:

1. Disable the Test Data Manager Service.
2. Select the assigned license in the Domain Navigator.
3. Click **Assigned Services**.

4. Click **Edit Assigned Services**.
5. Select the Test Data Manager Service from the **Assigned Services** list and click **Remove** to unassign it.
6. Select the new license in the Domain Navigator.
7. Click **Assigned Services**.
8. Click **Edit Assigned Services**.
9. Select the Test Data Manager Service from the **Unassigned Services** list and click **Add** to assign it.
10. Click **OK**.
11. Enable the Test Data Manager Service.

Deleting the Test Data Manager Service

1. In the Domain Navigator, select the Test Data Manager Service.
2. Click **Actions > Disable Service** to disable the service.
3. Click **Actions > Delete**.

You cannot access Test Data Manager if you delete the service.

CHAPTER 21

Test Data Warehouse Service

This chapter includes the following topics:

- [Test Data Warehouse Service Overview, 368](#)
- [Test Data Warehouse Services Dependencies, 368](#)
- [Test Data Warehouse Service Properties, 369](#)
- [Creating the Test Data Warehouse Service, 371](#)
- [Process Properties for the Test Data Warehouse Service, 372](#)

Test Data Warehouse Service Overview

Configure a Test Data Warehouse Service if you want to create a test data warehouse in TDM.

The Test Data Warehouse Service manages the test data warehouse repository and the test data warehouse.

The test data warehouse repository is a relational database that stores the metadata created when you run operations to store data in the test data warehouse. The test data warehouse is a relational database that stores the source data that you include in data sets.

Use the Administrator tool or the `infacmd` command line program to administer the Test Data Warehouse Service.

When you create a Test Data Warehouse Service you can create a test data warehouse repository or use an existing test data warehouse repository. You can run multiple Test Data Warehouse services on the same node. Manage the service users, groups, privileges, and roles from the **Security** tab of the Administrator tool. Manage permissions for test data warehouse repository objects in Test Data Manager.

Test Data Warehouse Services Dependencies

The Test Data Warehouse Service depends on other application services to perform tasks.

Before you create the Test Data Warehouse Service, you must create the services that it depends on.

PowerCenter Services

Create the PowerCenter services that the Test Data Warehouse Service depends on in the following order:

1. PowerCenter Repository Service
Test Data Manager requires this service to access metadata stored in the PowerCenter repository.
2. PowerCenter Integration Service
Test Data Manager requires this service to run workflows and sessions.

Test Data Manager Service

To work with the test data warehouse, you require the Test Data Manager web client. Create a Test Data Manager Service and associate the Test Data Warehouse Service with it. Alternatively, update the Test Data Manager Service to associate the Test Data Warehouse Service with it.

Test Data Warehouse Service Properties

To view the Test Data Manager Service properties, select the service in the Domain Navigator and click the **Properties** view. You can configure the following Test Data Warehouse Service properties:

- General properties
- Test data warehouse repository configuration properties
- Test data warehouse properties
- Server configuration properties

General Properties

The following table describes the general properties for the service:

Property	Description
Name	Name of the service. The name is not case sensitive and must be unique within the domain. It cannot exceed 128 characters or begin with @. It also cannot contain spaces or the following special characters: ` ~ % ^ * + = { } \ ; : ' " / ? . , < > ! () [] You cannot change the name of the service after you create it.
Description	Description of the service. The description cannot exceed 765 characters.
Location	Domain and folder where the service is created. Click Browse to choose a different folder. You can move the service after you create it.
License	License object that allows use of the service.
Node	Node on which the service runs.

Test Data Warehouse Repository Configuration Properties

The following table describes the test data warehouse repository configuration properties for the service:

Property	Description
Repository Name	Name of the test data warehouse repository.
Database Type	Type of database for the test data warehouse repository. <ul style="list-style-type: none">- Oracle- Microsoft SQL Server- DB2
User Name	User account for the test data warehouse repository database. Set up this account using the appropriate database client tools.
Password	Password for the test data warehouse repository database user. Must be in 7-bit ASCII.
JDBC URL	JDBC connection URL used to access the test data warehouse repository database. Enter the JDBC URL in one of the following formats: <ul style="list-style-type: none">- Oracle: jdbc:informatica:oracle://<host name>:<port>;SID=<database name>- IBM DB2: jdbc:informatica:db2://<host name>:<port>;DatabaseName=<database name>- Microsoft SQL Server: jdbc:informatica:sqlserver://<host name>:<port>;SelectMethod=cursor;DatabaseName=<database name>
Schema Name	Available for Microsoft SQL Server. Optional. Name of the schema for the database. If not selected, the service creates the tables in the default schema.
Tablespace	Available for DB2. Name of the tablespace in which to create the tables. You must define the tablespace on a single node and the page size must be 32 KB. In a multipartition database, you must select this option. In a single-partition database, if you do not select this option, the installer creates the tables in the default tablespace.
Content creation options for the new Test Data Warehouse Service	Options to create content, or use existing content, and upgrade existing content. <ul style="list-style-type: none">- Do not create new content. Creates the repository without creating content. Select this option if the database content exists. If the content is of a previous version, the service prompts you to upgrade the content to the current version.- Create new content. Creates repository content.

Test Data Warehouse Properties

The following table describes the test data warehouse properties for the service:

Property	Description
Test Data Warehouse Name	Name of the test data warehouse.
Description	Description of the test data warehouse. The description cannot exceed 765 characters.
Connection Type	The type of connection to use as the test data warehouse.
Target Connection	The database connection to use as the test data warehouse.

Test Data Warehouse Server Configuration Properties

The following table describes the test data warehouse server configuration properties for the service:

Property	Description
HTTP Port	Port number of the Test Data Warehouse Service. The default is 7705.
Enable Transport Layer Security (TLS)	Secures communication between the Test Data Warehouse Service and the domain.
HTTPS Port	Port number for the HTTPS connection.
Keystore File	Path and file name of the keystore file. The keystore file contains the keys and certificates required if you use the SSL security protocol with the test data warehouse. Required if you select Enable Transport Layer Security.
Keystore Password	Password for the keystore file. Required if you select Enable Secured Socket Layer.
SSL Protocol	Secure Sockets Layer protocol to use. Default is TLS.
JVM Params	<p>The heap size allocated for the Test Data Warehouse Service processes.</p> <p>- Xms256m -Xmx512m -XX:MaxMetaspaceSize=256m</p> <p>The time after which database connections are renewed if the Test Data Warehouse Service remains idle. Required if you modified the database configuration settings to values less than the test data warehouse defaults.</p> <p>Configure the following test data warehouse values to be less than the database values:</p> <p>- IDLE_TIME. -DIDLE_TIME=<seconds>. Default is 300 seconds.</p> <p>- CONNECT_TIME. DCONNECT_TIME=<seconds>. Default is 5000 seconds.</p>

Creating the Test Data Warehouse Service

Use the service creation wizard in the Administrator tool to create the service.

1. In the Administrator tool, click the **Manage** tab.
2. Click the **Services and Nodes** view.
3. Click **Actions > New > Test Data Warehouse Service**.
The **New Test Data Warehouse Service** dialog box appears.
4. On the **New Test Data Warehouse Service - Step 1 of 4** page, enter the general properties and click **Next**.
5. On the **New Test Data Warehouse Service - Step 2 of 4** page, enter the test data warehouse repository properties and click **Next**.
6. On the **New Test Data Warehouse Service - Step 3 of 4** page, enter the test data warehouse properties and click **Next**.
7. On the **New Test Data Warehouse Service - Step 4 of 4** page, enter the server configuration properties and the advanced properties.

8. Click **Finish**.

The domain creates the Test Data Warehouse Service, starts the service, and creates content for the test data warehouse repository.

Process Properties for the Test Data Warehouse Service

The Test Data Warehouse Service process has the following node properties:

Node

Node that the service process runs on.

Node Role

The purpose of the node. Role can be service role or compute role.

Node Status

Status of the node. Status can be enabled or disabled.

Process Configuration

Status of the process configured to run on the node.

Process State

State of the service process running on the node. The state can be enabled or disabled.

CHAPTER 22

Web Services Hub

This chapter includes the following topics:

- [Web Services Hub Overview, 373](#)
- [Creating a Web Services Hub, 374](#)
- [Enabling and Disabling the Web Services Hub, 375](#)
- [Web Services Hub Properties, 376](#)
- [Configuring the Associated Repository, 380](#)

Web Services Hub Overview

The Web Services Hub Service is an application service in the Informatica domain that exposes PowerCenter functionality to external clients through web services. It receives requests from web service clients and passes them to the PowerCenter Integration Service or PowerCenter Repository Service. The PowerCenter Integration Service or PowerCenter Repository Service processes the requests and sends a response to the Web Services Hub. The Web Services Hub sends the response back to the web service client.

The Web Services Hub Console does not require authentication. You do not need to log in when you start the Web Services Hub Console. On the Web Services Hub Console, you can view the properties and the WSDL of any web service. You can test any web service running on the Web Services Hub. However, when you test a protected service you must run the login operation before you run the web service.

You can use the Administrator tool to complete the following tasks related to the Web Services Hub:

- Create a Web Services Hub. You can create multiple Web Services Hub Services in a domain.
- Enable or disable the Web Services Hub. You must enable the Web Services Hub to run web service workflows. You can disable the Web Services Hub to prevent external clients from accessing the web services while performing maintenance on the machine or modifying the repository.
- Configure the Web Services Hub properties. You can configure Web Services Hub properties such as the length of time a session can remain idle before time out and the character encoding to use for the service.
- Configure the associated repository. You must associate a repository with a Web Services Hub. The Web Services Hub exposes the web-enabled workflows in the associated repository.
- View the logs for the Web Services Hub. You can view the event logs for the Web Services Hub in the Log Viewer.
- Remove a Web Services Hub. You can remove a Web Services Hub if it becomes obsolete.

Creating a Web Services Hub

Create a Web Services Hub to run web service workflows so that external clients can access PowerCenter functionality as web services.

You must associate a PowerCenter repository with the Web Services Hub before you run it. The PowerCenter repository that you assign to the Web Services Hub is called the associated repository. The Web Services Hub runs web service workflows that are in the associated repository.

By default, the Web Services Hub has the same code page as the node on which it runs. When you associate a PowerCenter repository with the Web Services Hub, the code page of the Web Services Hub must be a subset of the code page of the associated repository.

If the domain contains multiple nodes and you create a secure Web Services Hub, you must generate the SSL certificate for the Web Services Hub on a gateway node and import the certificate into the certificate file of the same gateway node.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. On the Domain Navigator Actions menu, click New > Web Services Hub.

The New Web Services Hub Service window appears.

3. Configure the properties of the Web Services Hub.

The following table describes the properties for a Web Services Hub:

Property	Description
Name	Name of the Web Services Hub. The characters must be compatible with the code page of the associated repository. The name is not case sensitive and must be unique within the domain. It cannot exceed 128 characters or begin with @. It also cannot contain spaces or the following special characters: ` ~ % ^ * + = { } \ ; : ' " / ? . , < > ! () []
Description	Description of the Web Services Hub. The description cannot exceed 765 characters.
Location	Domain folder in which the Web Services Hub is created. Click Browse to select the folder in the domain where you want to create the Web Services Hub.
License	License to assign to the Web Services Hub. If you do not select a license now, you can assign a license to the service later. Required before you can enable the Web Services Hub.
Node	Node on which the Web Services Hub runs. A Web Services Hub runs on a single node. A node can run more than one Web Services Hub.
Associated Repository Service	PowerCenter Repository Service to which the Web Services Hub connects. The repository must be enabled before you can associate it with a Web Services Hub.
Repository User Name	User name to access the repository.
Repository Password	Password for the user.
Security Domain	Security domain for the user. Appears when the Informatica domain contains an LDAP security domain.

Property	Description
URLScheme	Indicates the security protocol that you configure for the Web Services Hub: <ul style="list-style-type: none"> - HTTP. Run the Web Services Hub on HTTP only. - HTTPS. Run the Web Services Hub on HTTPS only. - HTTP and HTTPS. Run the Web Services Hub in HTTP and HTTPS modes.
HubHostName	Name of the machine hosting the Web Services Hub.
HubPortNumber (http)	Optional. Port number for the Web Services Hub on HTTP. Default is 7333.
HubPortNumber (https)	Port number for the Web Services Hub on HTTPS. Appears when the URL scheme selected includes HTTPS. Required if you choose to run the Web Services Hub on HTTPS. Default is 7343.
KeystoreFile	Path and file name of the keystore file that contains the keys and certificates required if you use the SSL security protocol with the Web Services Hub. Required if you run the Web Services Hub on HTTPS.
Keystore Password	Password for the keystore file. The value of this property must match the password you set for the keystore file. If this property is empty, the Web Services Hub assumes that the password for the keystore file is the default password <i>changeit</i> .
InternalHostName	Host name on which the Web Services Hub listens for connections from the PowerCenter Integration Service. If not specified, the default is the Web Services Hub host name. Note: If the host machine has more than one network card that results in multiple IP addresses for the host machine, set the value of InternalHostName to the internal IP address.
InternalPortNumber	Port number on which the Web Services Hub listens for connections from the PowerCenter Integration Service. Default is 15555.

4. Click Create.

After you create the Web Services Hub, the Administrator tool displays the URL for the Web Services Hub Console. If you run the Web Services Hub on HTTP and HTTPS, the Administrator tool displays the URL for both.

If you configure a logical URL for an external load balancer to route requests to the Web Services Hub, the Administrator tool also displays the URL.

Click the service URL to start the Web Services Hub Console from the Administrator tool. If the Web Services Hub is not enabled, you cannot connect to the Web Services Hub Console.

Enabling and Disabling the Web Services Hub

Use the Administrator tool to enable or disable a Web Services Hub. You can disable a Web Services Hub to perform maintenance or to temporarily restrict users from accessing web services. Enable a disabled Web Services Hub to make it available again.

The PowerCenter Repository Service associated with the Web Services Hub must be running before you enable the Web Services Hub. If a Web Services Hub is associated with multiple PowerCenter Repository

Services, at least one of the PowerCenter Repository Services must be running before you enable the Web Services Hub.

If you enable the service but it fails to start, review the logs for the Web Services Hub to determine the reason for the failure. After you resolve the problem, you must disable and then enable the Web Services Hub to start it again.

When you disable a Web Services Hub, you must choose the mode to disable it in. You can choose one of the following modes:

- **Stop.** Stops all web enabled workflows and disables the Web Services Hub.
- **Abort.** Aborts all web-enabled workflows immediately and disables the Web Services Hub.

To disable or enable a Web Services Hub:

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the Web Services Hub.
When a Web Services Hub is running, the Disable button is available.
3. To disable the service, click the Disable the Service button.
The Disable Web Services Hub window appears.
4. Choose the disable mode and click OK.
The Service Manager disables the Web Services Hub. When a service is disabled, the Enable button is available.
5. To enable the service, click the Enable the Service button.
6. To disable the Web Services Hub with the default disable mode and then immediately enable the service, click the Restart the Service button.
By default, when you restart a Web Services Hub, the disable mode is Stop.

Web Services Hub Properties

You can configure general, service, advanced, and custom properties for the Web Services Hub.

Use the Administrator tool to view or edit the following Web Services Hub properties:

- **General properties.** Configure general properties such as license and node.
 - **Service properties.** Configure service properties such as host name and port number.
 - **Advanced properties.** Configure advanced properties such as the level of errors written to the Web Services Hub logs.
 - **Custom properties.** Configure custom properties that are unique to specific environments.
1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
 2. In the Domain Navigator, select a Web Services Hub.
 3. To view the properties of the service, click the Properties view.
 4. To edit the properties of the service, click Edit for the category of properties you want to update.
The Edit Web Services Hub Service window displays the properties in the category.
 5. Update the values of the properties.

General Properties

Select the node on which to run the Web Services Hub. You can run multiple Web Services Hub on the same node.

Disable the Web Services Hub before you assign it to another node. To edit the node assignment, select the Web Services Hub in the Navigator, click the Properties tab, and then click Edit in the Node Assignments section. Select a new node.

When you change the node assignment for a Web Services Hub, the host name for the web services running on the Web Services Hub changes. You must update the host name and port number of the Web Services Hub to match the new node. Update the following properties of the Web Services Hub:

- HubHostName
- InternalHostName

To access the Web Services Hub on a new node, you must update the client application to use the new host name. For example, you must regenerate the WSDL for the web service to update the host name in the endpoint URL. You must also regenerate the client proxy classes to update the host name.

The following table describes the general properties for the service:

Property	Description
Name	Name of the service. The name is not case sensitive and must be unique within the domain. It cannot exceed 128 characters or begin with @. It also cannot contain spaces or the following special characters: ` ~ % ^ * + = { } \ ; : ' " / ? . , < > ! () [] You cannot change the name of the service after you create it.
Description	Description of the service. The description cannot exceed 765 characters.
License	License object that allows use of the service.
Node	Node on which the service runs.

Service Properties

You must restart the Web Services Hub before changes to the service properties can take effect.

The following table describes the service properties for a Web Services Hub:

Property	Description
HubHostName	Name of the machine hosting the Web Services Hub. Default is the name of the machine where the Web Services Hub is running. If you change the node on which the Web Services Hub runs, update this property to match the host name of the new node. To apply changes, restart the Web Services Hub.
HubPortNumber (http)	Port number for the Web Services Hub running on HTTP. Required if you run the Web Services Hub on HTTP. Default is 7333. To apply changes, restart the Web Services Hub.
HubPortNumber (https)	Port number for the Web Services Hub running on HTTPS. Required if you run the Web Services Hub on HTTPS. Default is 7343. To apply changes, restart the Web Services Hub.

Property	Description
CharacterEncoding	Character encoding for the Web Services Hub. Default is UTF-8. To apply changes, restart the Web Services Hub.
URLScheme	Indicates the security protocol that you configure for the Web Services Hub: <ul style="list-style-type: none"> - HTTP. Run the Web Services Hub on HTTP only. - HTTPS. Run the Web Services Hub on HTTPS only. - HTTP and HTTPS. Run the Web Services Hub in HTTP and HTTPS modes. If you run the Web Services Hub on HTTPS, you must provide information on the keystore file. To apply changes, restart the Web Services Hub.
InternalHostName	Host name on which the Web Services Hub listens for connections from the Integration Service. If you change the node assignment of the Web Services Hub, update the internal host name to match the host name of the new node. To apply changes, restart the Web Services Hub.
InternalPortNumber	Port number on which the Web Services Hub listens for connections from the Integration Service. Default is 15555. To apply changes, restart the Web Services Hub.
KeystoreFile	Path and file name of the keystore file that contains the keys and certificates required if you use the SSL security protocol with the Web Services Hub. Required if you run the Web Services Hub on HTTPS.
KeystorePass	Password for the keystore file. The value of this property must match the password you set for the keystore file.

Advanced Properties

The following table describes the advanced properties for a Web Services Hub:

Property	Description
HubLogicalAddress	URL for the third party load balancer that manages the Web Services Hub. This URL is published in the WSDL for all web services that run on a Web Services Hub managed by the load balancer.
DTMTimeout	Length of time, in seconds, that the Web Services Hub tries to connect or reconnect to the DTM to run a session. Default is 60 seconds.
SessionExpiryPeriod	Number of seconds that a session can remain idle before the session times out and the session ID becomes invalid. The Web Services Hub resets the start of the timeout period every time a client application sends a request with a valid session ID. If a request takes longer to complete than the amount of time set in the SessionExpiryPeriod property, the session can time out during the operation. To avoid timing out, set the SessionExpiryPeriod property to a higher value. The Web Services Hub returns a fault response to any request with an invalid session ID. Default is 3600 seconds. You can set the SessionExpiryPeriod between 1 and 2,592,000 seconds.
MaxISConnections	Maximum number of connections to the PowerCenter Integration Service that can be open at one time for the Web Services Hub. Default is 20.

Property	Description
Log Level	<p>Configure the Log Level property to set the logging level. The following values are valid:</p> <ul style="list-style-type: none"> - Fatal. Writes FATAL messages to the log. FATAL messages include nonrecoverable system failures that cause the service to shut down or become unavailable. - Error. Writes FATAL and ERROR code messages to the log. ERROR messages include connection failures, failures to save or retrieve metadata, service errors. - Warning. Writes FATAL, WARNING, and ERROR messages to the log. WARNING errors include recoverable system failures or warnings. - Info. Writes FATAL, INFO, WARNING, and ERROR messages to the log. INFO messages include system and service change messages. - Trace. Write FATAL, TRACE, INFO, WARNING, and ERROR code messages to the log. TRACE messages log user request failures. - Debug. Write FATAL, DEBUG, TRACE, INFO, WARNING, and ERROR messages to the log. DEBUG messages are user request logs. <p>The default value is Info.</p>
MaxConcurrentRequests	Maximum number of request processing threads allowed, which determines the maximum number of simultaneous requests that can be handled. Default is 100.
MaxQueueLength	Maximum queue length for incoming connection requests when all possible request processing threads are in use. Any request received when the queue is full is rejected. Default is 5000.
MaxStatsHistory	Number of days that Informatica keeps statistical information in the history file. Informatica keeps a history file that contains information regarding the Web Services Hub activities. The number of days you set in this property determines the number of days available for which you can display historical statistics in the Web Services Report page of the Administrator tool.
Maximum Heap Size	<p>Amount of RAM allocated to the Java Virtual Machine (JVM) that runs the Web Services Hub. Use this property to increase the performance. Append one of the following letters to the value to specify the units:</p> <ul style="list-style-type: none"> - b for bytes. - k for kilobytes. - m for megabytes. - g for gigabytes. <p>Default is 512 megabytes.</p>
JVM Command Line Options	<p>Java Virtual Machine (JVM) command line options to run Java-based programs. When you configure the JVM options, you must set the Java SDK classpath, Java SDK minimum memory, and Java SDK maximum memory properties.</p> <p>You must set the following JVM command line option:</p> <ul style="list-style-type: none"> - Dfile.encoding. File encoding. Default is UTF-8.

Use the MaxConcurrentRequests property to set the number of clients that can connect to the Web Services Hub and the MaxQueueLength property to set the number of client requests the Web Services Hub can process at one time.

You can change the parameter values based on the number of clients you expect to connect to the Web Services Hub. In a test environment, set the parameters to smaller values. In a production environment, set the parameters to larger values. If you increase the values, more clients can connect to the Web Services Hub, but the connections use more system resources.

Custom Properties for the Web Services Hub

Configure custom properties that are unique to specific environments.

You might need to apply custom properties in special cases. When you define a custom property, enter the property name and an initial value. Define custom properties only at the request of Informatica Global Customer Support.

Configuring the Associated Repository

To expose web services through the Web Services Hub, you must associate the Web Services Hub with a repository. The code page of the Web Services Hub must be a subset of the code page of the associated repository.

When you associate a repository with a Web Services Hub, you specify the PowerCenter Repository Service and the user name and password used to connect to the repository. The PowerCenter Repository Service that you associate with a Web Services Hub must be in the same domain as the Web Services Hub.

You can associate more than one repository with a Web Services Hub. When you associate more than one repository with a Web Services Hub, the Web Services Hub can run web services located in any of the associated repositories.

You can associate more than one Web Services Hub with a PowerCenter repository. When you associate more than one Web Services Hub with a PowerCenter repository, multiple Web Services Hub Services can provide the same web services. Different Web Services Hub Services can run separate instances of a web service. You can use an external load balancer to manage the Web Services Hub Services.

When you associate a Web Services Hub with a PowerCenter Repository Service, the Repository Service does not have to be running. After you start the Web Services Hub, it periodically checks whether the PowerCenter Repository Services have started. The PowerCenter Repository Service must be running before the Web Services Hub can run a web service workflow.

Adding an Associated Repository

If you associate multiple PowerCenter repositories with a Web Services Hub, external clients can access web services from different repositories through the same Web Services Hub.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. On the Domain Navigator of the Administrator tool, select the Web Services Hub.
3. Click the Associated Repository tab.
4. Click Add.

The Select Repository section appears.

5. Enter the properties for the associated repository.

Property	Description
Associated Repository Service	Name of the PowerCenter Repository Service to which the Web Services Hub connects. To apply changes, restart the Web Services Hub.
Repository User Name	User name to access the repository. Not available for a domain with Kerberos authentication.
Repository Password	Password for the user. Not available for a domain with Kerberos authentication.
Security Domain	Security domain for the user. Appears when the Informatica domain contains an LDAP security domain.

6. Click OK to save the associated repository properties.

Editing an Associated Repository

If you want to change the repository that associated with the Web Services Hub, edit the properties of the associated repository.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. In the Domain Navigator, select the Web Services Hub for which you want to change an associated repository.
3. Click the Associated Repository view.
4. In the section for the repository you want to edit, click Edit.
The Edit associated repository window appears.
5. Edit the properties for the associated repository.

Property	Description
Associated Repository Service	Name of the PowerCenter Repository Service to which the Web Services Hub connects. To apply changes, restart the Web Services Hub.
Repository User Name	User name to access the repository. Not available for a domain with Kerberos authentication.
Repository Password	Password for the user. Not available for a domain with Kerberos authentication.
Security Domain	Security domain for the user. Appears when the Informatica domain contains an LDAP security domain.

6. Click OK to save the changes to the associated repository properties.

CHAPTER 23

Application Service Upgrade

This chapter includes the following topics:

- [Application Service Upgrade Overview, 382](#)
- [Running the Service Upgrade Wizard, 383](#)
- [Verify the Model Repository Service Upgrade, 384](#)

Application Service Upgrade Overview

The Informatica services version that you upgrade from determines the application service upgrade process.

Some Informatica services versions require that you upgrade the application services. When you upgrade an application service, you must also upgrade the dependent services. When you upgrade an application service, the upgrade process upgrades the database contents of the databases associated with the service.

Use the service upgrade wizard, the actions menu of each service, or the command line to upgrade application services. The service upgrade wizard upgrades multiple services in the appropriate order and checks for dependencies. If you use the actions menu of each service or the command line to upgrade application services, you must upgrade the application services in the correct order and verify that you upgrade dependent services.

The privileges required to upgrade application services depend on the service.

After you upgrade the Model Repository Service, check the log to verify that the upgrade completed successfully.

Privileges to Upgrade Services

The privileges required to upgrade application services depend on the application service.

A user with the Administrator role on the domain can access the service upgrade wizard.

A user must have these roles, privileges, and permissions to upgrade the following application services:

Model Repository Service

To upgrade the Model Repository Service using the service upgrade wizard, a user must have the following credentials:

- Administrator role on the domain.
- Create, Edit, and Delete Projects privilege for the Model Repository Service and write permission on projects.

To upgrade the Model Repository Service from the Actions menu or from the command line, a user must have the following credentials:

- Manage Services privilege for the domain and permission on the Model Repository Service.
- Create, Edit, and Delete Projects privilege for the Model Repository Service and write permission on projects.

Data Integration Service

To upgrade the Data Integration Service, a user must have the Administrator role on the Data Integration Service.

Content Management Service

To upgrade the Content Management Service, a user must have the Administrator role on the Content Management Service.

PowerCenter Repository Service

To upgrade the PowerCenter Repository Service, a user must have the Manage Services privilege for the domain and permission on the PowerCenter Repository Service.

Metadata Manager Service

To upgrade the Metadata Manager Service, a user must have the Manage Services privilege for the domain and permission on the Metadata Manager Service.

Service Upgrade from Previous Versions

When you upgrade from a previous version, some application services require an upgrade. Upgrade the application services that you used in the previous version.

Before you upgrade, verify that the Metadata Manager Service is disabled. Verify that all other application services are enabled.

To upgrade application services, upgrade the following services and associated databases in this order:

1. Model Repository Service
2. Data Integration Service
3. Profiling warehouse for the Data Integration Service
4. Metadata Manager Service
5. PowerCenter Repository Service

Note: When you upgrade all other application services, the upgrade process upgrades the database contents of the databases associated with the service.

Running the Service Upgrade Wizard

Use the service upgrade wizard to upgrade application services and the database contents of the databases associated with the services. The service upgrade wizard displays upgraded services in a list along with

services and associated databases that require an upgrade. You can also save the current or previous upgrade report.

Note: The Metadata Manager Service must be disabled before the upgrade. All other services must be enabled before the upgrade.

1. In the Informatica Administrator header area click **Manage > Upgrade**.
2. Select the application services and associated databases to upgrade.
3. Optionally, specify if you want to **Automatically recycle services after upgrade**.
If you choose to automatically recycle application services after the upgrade, the upgrade wizard restarts the services after they have been upgraded.
4. Click **Next**.
5. If dependency errors exist, the **Dependency Errors** dialog box appears. Review the dependency errors and click **OK**. Then, resolve dependency errors and click **Next**.
6. Enter the repository login information.
7. Click **Next**.

The service upgrade wizard upgrades each application service and associated database and displays the status and processing details.

8. When the upgrade completes, the **Summary** section displays the list of application services and their upgrade status. Click each service to view the upgrade details in the **Service Details** section.
9. Optionally, click **Save Report** to save the upgrade details to a file.
If you choose not to save the report, you can click **Save Previous Report** the next time you launch the service upgrade wizard.
10. Click **Close**.
11. If you did not choose to automatically recycle application services after the upgrade, restart the upgraded services.

You can view the upgrade report and save the upgrade report. The second time you run the service upgrade wizard, the Save Previous Report option appears in the service upgrade wizard. If you did not save the upgrade report after upgrading services, you can select this option to view or save the previous upgrade report.

Verify the Model Repository Service Upgrade

After you upgrade the Model Repository Service, check the Model Repository Service log to verify that the upgrade completed successfully.

Object Dependency Graph

When you upgrade a Model Repository Service, the upgrade process upgrades the contents of the Model repository and rebuilds the object dependency graph.

If the upgrade process encounters a fatal error while upgrading the Model repository contents, then the service upgrade fails. The Administrator tool or the command line program informs you that you must perform the upgrade again.

If the upgrade process encounters a fatal error while rebuilding the object dependency graph, then the upgrade of the service succeeds. You cannot view object dependencies in the Developer tool until you rebuild the object dependency graph.

After you upgrade the Model Repository Service, verify that the Model Repository Service log includes the following message:

```
MRS_50431 "Finished rebuilding the object dependency graph for project group '<project group>'."
```

If the message does not exist in the log, run the `infacmd mrs rebuildDependencyGraph` command to rebuild the object dependency graph. Users must not access Model repository objects until the rebuild process completes, or the object dependency graph might not be accurate. Ask the users to log out of the Model Repository Service before service upgrade.

The `infacmd mrs rebuildDependencyGraph` command uses the following syntax:

```
rebuildDependencyGraph
<-DomainName|-dn> domain_name
[<-SecurityDomain|-sdn> security_domain]
<-UserName|-un> user_name
<-Password|-pd> password
<-ServiceName|-sn> service_name
[<-ResilienceTimeout|-re> timeout_period_in_seconds]
```

Maximum Heap Size

The upgrade process resets the Model Repository Service maximum heap size to 4 GB. After the upgrade, reset the maximum heap size property to the value to which it was set prior to the upgrade, or to the setting that Global Customer Support recommended for your environment.

Prior to Model Repository Service upgrade, set the value for `-Xss` parameter in the JVM Command Line Options to 1m. After the upgrade, reset the `-Xss` parameter value to the pre-upgrade value. The default value for `-Xss` parameter is 512k.

Effective in version 10.1, the property **MaxPermSize** in the JVM Command Line Options is replaced with **MaxMetaspaceSize**.

To reset the maximum heap size, select the service in the **Domain Navigator**, click the **Properties** view, and expand **Advanced Properties**. Set the **Maximum Heap Size** property to the pre-upgrade value. Set the **MaxMetaspaceSize** property to the minimum of 512 MB.

APPENDIX A

Application Service Databases

This appendix includes the following topics:

- [Application Service Databases Overview, 386](#)
- [Set Up Database User Accounts, 387](#)
- [Data Object Cache Database Requirements, 387](#)
- [Exception Management Audit Database Requirements, 388](#)
- [Metadata Manager Repository Database Requirements, 389](#)
- [Model Repository Database Requirements, 393](#)
- [PowerCenter Repository Database Requirements, 394](#)
- [Profiling Warehouse Requirements, 396](#)
- [Reference Data Warehouse Requirements, 398](#)
- [Workflow Database Requirements, 399](#)
- [Configure Native Connectivity on Service Machines, 401](#)

Application Service Databases Overview

Informatica stores data and metadata in repositories in the domain. Before you create the application services, set up the databases and database user accounts for the repositories associated with the application services.

Set up a database and user account for the following repositories:

- Data object cache repository
- Workflow repository
- Metadata Manager repository
- Model repository
- PowerCenter repository
- Profiling warehouse
- Reference data warehouse

To prepare the databases, verify the database requirements and set up the database. The database requirements depend on the application services that you create in the domain and the number of data integration objects that you build and store in the repositories.

Set Up Database User Accounts

Set up a database and user account for the domain configuration repository and for the repository databases associated with the applications services.

Use the following rules and guidelines when you set up the user accounts:

- The database user account must have permissions to create and drop tables, indexes, and views, and to select, insert, update, and delete data from tables.
- Use 7-bit ASCII to create the password for the account.
- To prevent database errors in one repository from affecting any other repository, create each repository in a separate database schema with a different database user account. Do not create a repository in the same database schema as the domain configuration repository or any other repository in the domain.
- If you create more than one domain, each domain configuration repository must have a separate user account.

Data Object Cache Database Requirements

The data object cache database stores cached logical data objects and virtual tables for the Data Integration Service. You specify the data object cache database connection when you create the Data Integration Service.

The data object cache database supports the following database types:

- IBM DB2 UDB
- Microsoft SQL Server
- Oracle

Allow 200 MB of disk space for the database.

Note: Ensure that you install the database client on the machine on which you want to run the Data Integration Service.

IBM DB2 Database Requirements

Use the following guidelines when you set up the repository on IBM DB2:

- Verify that the database user account has `CREATETAB` and `CONNECT` privileges.
- Informatica does not support IBM DB2 table aliases for repository tables. Verify that table aliases have not been created for any tables in the database.
- Set the `tablespace pageSize` parameter to 32768 bytes.
- Set the `NPAGES` parameter to at least 5000. The `NPAGES` parameter determines the number of pages in the tablespace.

Microsoft SQL Server Database Requirements

Use the following guidelines when you set up the repository on Microsoft SQL Server:

- Verify that the database user account has `CONNECT` and `CREATE TABLE` privileges.

Oracle Database Requirements

Use the following guidelines when you set up the repository on Oracle:

- Verify that the database user has the following privileges:
 - CREATE INDEX
 - CREATE SESSION
 - CREATE SYNONYM
 - CREATE TABLE
 - CREATE VIEW
 - DROP TABLE
 - INSERT INTO TABLE
 - UPDATE TABLE
- Informatica does not support Oracle public synonyms for repository tables. Verify that public synonyms have not been created for any tables in the database.

Exception Management Audit Database Requirements

The exception management audit database is a single repository for data that describes the work that Analyst tool users perform on Human task instances. The Analyst Service identifies the database connection and the schema name. The Data Integration Service writes the audit data to the database.

If the Analyst Service does not identify an exception management audit database, the Data Integration Service writes the audit data to the database that contains the task instance records.

The reference data warehouse supports the following database types:

- IBM DB2 UDB
- Microsoft SQL Server
- Oracle

Allow 200 MB of disk space for the database.

Note: Ensure that you install the database client on the machine on which you want to run the Content Management Service.

IBM DB2 Database Requirements

Use the following guidelines when you set up the repository on IBM DB2:

- The database user account must have the CREATETAB, CONNECT, CREATE VIEW, and CREATE FUNCTION privileges.
- Informatica does not support IBM DB2 table aliases for repository tables. Verify that table aliases have not been created for any tables in the database.
- Set the tablespace pageSize parameter to 32768 bytes.
- Set the NPAGES parameter to at least 5000. The NPAGES parameter determines the number of pages in the tablespace.

Microsoft SQL Server Database Requirements

Use the following guidelines when you set up the repository on Microsoft SQL Server:

- The database user account must have the CONNECT, CREATE TABLE, CREATE VIEW, and CREATE FUNCTION privileges.

Oracle Database Requirements

Use the following guidelines when you set up the repository on Oracle:

- Verify that the database user has the following privileges:
 - ALTER TABLE
 - CREATE SESSION
 - CREATE TABLE
 - DROP TABLE
 - UPDATE TABLE
- Informatica does not support Oracle public synonyms for repository tables. Verify that public synonyms have not been created for any tables in the database.
- Set the following parameters to the Informatica recommended values:

Parameter	Recommended Value
open_cursors	3000
Sessions	1000
Processes	1000

Metadata Manager Repository Database Requirements

Metadata Manager repository contains the Metadata Manager warehouse and models. The Metadata Manager warehouse is a centralized metadata warehouse that stores the metadata from metadata sources.

Specify the repository details when you create a Metadata Manager Service.

The Metadata Manager repository supports the following database types:

- IBM DB2 UDB
- Microsoft SQL Server
- Oracle

Allow 1 GB of disk space for the database.

For more information about configuring the database, see the documentation for your database system.

IBM DB2 Database Requirements

Use the following guidelines when you set up the repository on IBM DB2:

- The database user account that creates the repository must have privileges to perform the following operations:

```
ALTER TABLE
CREATE FUNCTION
CREATE INDEX
CREATE PROCEDURE
CREATE TABLE
CREATE VIEW
DROP PROCEDURE
DROP TABLE
INSERT INTO
```

- The database user that creates the repository must be able to create tablespaces with page sizes of 32 KB.
- Set up system temporary tablespaces larger than the default page size of 4 KB and update the heap sizes. Queries running against tables in tablespaces defined with a page size larger than 4 KB require system temporary tablespaces with a page size larger than 4 KB. If there are no system temporary table spaces defined with a larger page size, the queries can fail. The server displays the following error:

```
SQL 1585N A system temporary table space with sufficient page size does not exist.
SQLSTATE=54048
```

Create system temporary tablespaces with page sizes of 8 KB, 16 KB, and 32 KB. Run the following SQL statements on each database to configure the system temporary tablespaces and update the heap sizes:

```
CREATE Bufferpool RBF IMMEDIATE SIZE 1000 PAGESIZE 32 K EXTENDED STORAGE ;
CREATE Bufferpool STBF IMMEDIATE SIZE 2000 PAGESIZE 32 K EXTENDED STORAGE ;
CREATE REGULAR TABLESPACE REGTS32 PAGESIZE 32 K MANAGED BY SYSTEM USING ('C:
\DB2\NODE0000\reg32' ) EXTENTSIZE 16 OVERHEAD 10.5 PREFETCHSIZE 16 TRANSFERRATE 0.33
BUFFERPOOL RBF;
CREATE SYSTEM TEMPORARY TABLESPACE TEMP32 PAGESIZE 32 K MANAGED BY SYSTEM USING
('C:\DB2\NODE0000\temp32' ) EXTENTSIZE 16 OVERHEAD 10.5 PREFETCHSIZE 16 TRANSFERRATE
0.33 BUFFERPOOL STBF;
GRANT USE OF TABLESPACE REGTS32 TO USER <USERNAME>;
UPDATE DB CFG FOR <DB NAME> USING APP CTL HEAP SZ 16384
UPDATE DB CFG FOR <DB NAME> USING APPLHEAPSZ 16384
UPDATE DBM CFG USING QUERY HEAP SZ 8000
UPDATE DB CFG FOR <DB NAME> USING LOGPRIMARY 100
UPDATE DB CFG FOR <DB NAME> USING LOGFILSIZ 2000
UPDATE DB CFG FOR <DB NAME> USING LOCKLIST 1000
UPDATE DB CFG FOR <DB NAME> USING DBHEAP 2400
"FORCE APPLICATIONS ALL"
DB2STOP
DB2START
```

- Set the locking parameters to avoid deadlocks when you load metadata into a Metadata Manager repository on IBM DB2.

The following table lists the locking parameters you can configure:

Parameter Name	Value	IBM DB2 Description
LOCKLIST	8192	Max storage for lock list (4KB)
MAXLOCKS	10	Percent of lock lists per application
LOCKTIMEOUT	300	Lock timeout (sec)
DLCHKTIME	10000	Interval for checking deadlock (ms)

Also, for IBM DB2 9.7 and earlier, set the DB2_RR_TO_RS parameter to YES to change the read policy from Repeatable Read to Read Stability.

- Informatica does not support IBM DB2 table aliases for repository tables. Verify that table aliases have not been created for any tables in the database.

Note: If you use IBM DB2 as a metadata source, the source database has the same configuration requirements.

Microsoft SQL Server Database Requirements

Use the following guidelines when you set up the repository on Microsoft SQL Server:

- The database user account that creates the repository must have privileges to perform the following operations:
 - ALTER TABLE
 - CREATE CLUSTERED INDEX
 - CREATE INDEX
 - CREATE PROCEDURE
 - CREATE TABLE
 - CREATE VIEW
 - DROP PROCEDURE
 - DROP TABLE
 - INSERT INTO
- If the repository must store metadata in a multibyte language, set the database collation to that multibyte language when you install Microsoft SQL Server. For example, if the repository must store metadata in Japanese, set the database collation to a Japanese collation when you install Microsoft SQL Server. This is a one-time configuration and cannot be changed.

Oracle Database Requirements

Use the following guidelines when you set up the repository on Oracle:

- Verify that the database user has the following privileges:

- ALTER TABLE
 - CREATE CLUSTER
 - CREATE INDEX
 - CREATE OR REPLACE FORCE VIEW
 - CREATE OR REPLACE PROCEDURE
 - CREATE OR REPLACE VIEW
 - CREATE SESSION
 - CREATE TABLE
 - DROP TABLE
 - INSERT INTO TABLE

- Set the following parameters for the tablespace on Oracle:

<Temporary tablespace>

Resize to at least 2 GB.

CURSOR_SHARING

Set to FORCE.

MEMORY_TARGET

Set to at least 4 GB.

Run `SELECT * FROM v$memory_target_advice ORDER BY memory_size;` to determine the optimal MEMORY_SIZE.

MEMORY_MAX_TARGET

Set to greater than the MEMORY_TARGET size.

If MEMORY_MAX_TARGET is not specified, MEMORY_MAX_TARGET defaults to the MEMORY_TARGET setting.

OPEN_CURSORS

Set to 3000 shared.

Monitor and tune open cursors. Query `v$sesstat` to determine the number of currently-opened cursors. If the sessions are running close to the limit, increase the value of OPEN_CURSORS.

UNDO_MANAGEMENT

Set to AUTO.

- If the repository must store metadata in a multibyte language, set the NLS_LENGTH_SEMANTICS parameter to CHAR on the database instance. Default is BYTE.
- Informatica does not support Oracle public synonyms for repository tables. Verify that public synonyms have not been created for any tables in the database.

Model Repository Database Requirements

Informatica services and clients store data and metadata in the Model repository. Configure a separate Model repository to store monitoring statistics. Before you create the Model Repository Service, set up a database and database user account for the Model repository.

The Model repository supports the following database types:

- IBM DB2 UDB
- Microsoft SQL Server
- Oracle

Allow 3 GB of disk space for DB2. Allow 200 MB of disk space for all other database types.

For more information about configuring the database, see the documentation for your database system.

IBM DB2 Database Requirements

Use the following guidelines when you set up the repository on IBM DB2:

- If the repository is in an IBM DB2 9.7 database, verify that IBM DB2 Version 9.7 Fix Pack 7 or a later fix pack is installed.
- On the IBM DB2 instance where you create the database, set the following parameters to ON:
 - DB2_SKIPINSERTED
 - DB2_EVALUNCOMMITTED
 - DB2_SKIPDELETED
 - AUTO_RUNSTATS
- On the database, set the configuration parameters.

The following table lists the configuration parameters that you must set:

Parameter	Value
logfilsiz	8000
maxlocks	98
locklist	50000
auto_stmt_stats	ON

- Set the tablespace pageSize parameter to 32768 bytes.

In a single-partition database, specify a tablespace that meets the pageSize requirements. If you do not specify a tablespace, the default tablespace must meet the pageSize requirements.

In a multi-partition database, specify a tablespace that meets the pageSize requirements. Define the tablespace in the catalog partition of the database.
- Set the NPAGES parameter to at least 5000. The NPAGES parameter determines the number of pages in the tablespace.
- Verify that the database user has CREATETAB, CONNECT, and BINDADD privileges.
- Informatica does not support IBM DB2 table aliases for repository tables. Verify that table aliases have not been created for any tables in the database.

- In the DataDirect Connect for JDBC utility, update the DynamicSections parameter to 3000.
The default value for DynamicSections is too low for the Informatica repositories. Informatica requires a larger DB2 package than the default. When you set up the DB2 database for the domain configuration repository or a Model repository, you must set the DynamicSections parameter to at least 3000. If the DynamicSections parameter is set to a lower number, you can encounter problems when you install or run Informatica services.
For more information about updating the DynamicSections parameter, see [Appendix D, “Updating the DynamicSections Parameter of a DB2 Database” on page 435](#).

Microsoft SQL Server Database Requirements

Use the following guidelines when you set up the repository on Microsoft SQL Server:

- Set the allow snapshot isolation and read committed isolation level to ALLOW_SNAPSHOT_ISOLATION and READ_COMMITTED_SNAPSHOT to minimize locking contention.
To set the isolation level for the database, run the following commands:

```
ALTER DATABASE DatabaseName SET ALLOW_SNAPSHOT_ISOLATION ON
ALTER DATABASE DatabaseName SET READ_COMMITTED_SNAPSHOT ON
```


To verify that the isolation level for the database is correct, run the following commands:

```
SELECT snapshot_isolation_state FROM sys.databases WHERE name=[DatabaseName]
SELECT is_read_committed_snapshot_on FROM sys.databases WHERE name = DatabaseName
```
- The database user account must have the CONNECT, CREATE TABLE, and CREATE VIEW privileges.

Oracle Database Requirements

Use the following guidelines when you set up the repository on Oracle:

- Verify that the database user has the following privileges:

```
CREATE SEQUENCE
CREATE SESSION
CREATE SYNONYM
CREATE TABLE
CREATE VIEW
```
- Informatica does not support Oracle public synonyms for repository tables. Verify that public synonyms have not been created for any tables in the database.

PowerCenter Repository Database Requirements

A PowerCenter repository is a collection of database tables containing metadata. A PowerCenter Repository Service manages the repository and performs all metadata transactions between the repository database and repository clients.

The PowerCenter repository supports the following database types:

- IBM DB2 UDB

- Microsoft SQL Server
- Oracle
- Sybase ASE

Allow 35 MB of disk space for the database.

Note: Ensure that you install the database client on the machine on which you want to run the PowerCenter Repository Service.

For more information about configuring the database, see the documentation for your database system.

IBM DB2 Database Requirements

Use the following guidelines when you set up the repository on IBM DB2:

- To optimize repository performance, set up the database with the tablespace on a single node. When the tablespace is on one node, PowerCenter Client and PowerCenter Integration Service access the repository faster than if the repository tables exist on different database nodes.
Specify the single-node tablespace name when you create, copy, or restore a repository. If you do not specify the tablespace name, DB2 uses the default tablespace.
- Informatica does not support IBM DB2 table aliases for repository tables. Verify that table aliases have not been created for any tables in the database.

Microsoft SQL Server Database Requirements

Use the following guidelines when you set up the repository on Microsoft SQL Server:

- Set the database server page size to 8K or higher. This is a one-time configuration and cannot be changed afterwards.
- Verify that the database user account has the CONNECT, CREATE TABLE, and CREATE VIEW privileges.

Oracle Database Requirements

Use the following guidelines when you set up the repository on Oracle:

- Set the storage size for the tablespace to a small number to prevent the repository from using an excessive amount of space. Also verify that the default tablespace for the user that owns the repository tables is set to a small size.

The following example shows how to set the recommended storage parameter for a tablespace named REPOSITORY:

```
ALTER TABLESPACE "REPOSITORY" DEFAULT STORAGE ( INITIAL 10K NEXT 10K MAXEXTENTS
UNLIMITED PCTINCREASE 50 );
```

Verify or change the storage parameter for a tablespace before you create the repository.

- Verify that the database user has the following privileges:

```
CREATE SEQUENCE
CREATE SESSION
CREATE SYNONYM
CREATE TABLE
CREATE VIEW
```

- Informatica does not support Oracle public synonyms for repository tables. Verify that public synonyms have not been created for any tables in the database.

Sybase ASE Database Requirements

Use the following guidelines when you set up the repository on Sybase ASE:

- Set the database server page size to 8K or higher. This is a one-time configuration and cannot be changed afterwards.
- Set the Sybase database option "ddl in tran" to TRUE.
- Set "allow nulls by default" to TRUE.
- Verify the database user has CREATE TABLE and CREATE VIEW privileges.
- Set the database memory configuration requirements.

The following table lists the memory configuration requirements and the recommended baseline values:

Database Configuration	Sybase System Procedure	Value
Number of open objects	sp_configure "number of open objects"	5000
Number of open indexes	sp_configure "number of open indexes"	5000
Number of open partitions	sp_configure "number of open partitions"	8000
Number of locks	sp_configure "number of locks"	100000

Profiling Warehouse Requirements

The profiling warehouse database stores profiling and scorecard results. You specify the profiling warehouse connection when you create the Data Integration Service.

The profiling warehouse supports the following database types:

- IBM DB2 UDB
- Microsoft SQL Server
- Oracle

Allow 10 GB of disk space for the database.

Note: Ensure that you install the database client on the machine on which you want to run the Data Integration Service. You can specify a JDBC connection or Hive connection as a profiling warehouse connection for IBM DB2 UDB, Microsoft SQL Server, and Oracle database types.

For more information about configuring the database, see the documentation for your database system.

IBM DB2 Database Requirements

Use the following guidelines when you set up the repository on IBM DB2:

- The database user account must have the CREATETAB, CONNECT, CREATE VIEW, and CREATE FUNCTION privileges.

- Informatica does not support IBM DB2 table aliases for repository tables. Verify that table aliases have not been created for any tables in the database.
- Set the tablespace pageSize parameter to 32768 bytes.
- Set the NPAGES parameter to at least 5000. The NPAGES parameter determines the number of pages in the tablespace.

Note: Informatica does not support the partitioned database environment for IBM DB2 databases when you use a JDBC connection as the profiling warehouse connection.

Microsoft SQL Server Database Requirements

Use the following guidelines when you set up the repository on Microsoft SQL Server:

- The database user account must have the CONNECT, CREATE TABLE, CREATE VIEW, and CREATE FUNCTION privileges.

Oracle Database Requirements

Use the following guidelines when you set up the repository on Oracle:

- Verify that the database user has the following privileges:
 ALTER TABLE
 CREATE ANY INDEX
 CREATE PROCEDURE
 CREATE SESSION
 CREATE TABLE
 CREATE VIEW
 DROP TABLE
 UPDATE TABLE
- Informatica does not support Oracle public synonyms for repository tables. Verify that public synonyms have not been created for any tables in the database.
- Set the tablespace parameter.
- Set the following parameters to the Informatica recommended values:

Parameter	Recommended Value
open_cursors	3000
Sessions	1000
Processes	1000

Reference Data Warehouse Requirements

The reference data warehouse stores the data values for reference table objects that you define in a Model repository. You configure a Content Management Service to identify the reference data warehouse and the Model repository.

You associate a reference data warehouse with a single Model repository. You can select a common reference data warehouse on multiple Content Management Services if the Content Management Services identify a common Model repository. The reference data warehouse must support mixed-case column names.

The reference data warehouse supports the following database types:

- IBM DB2 UDB
- Microsoft SQL Server
- Oracle

Allow 200 MB of disk space for the database.

Note: Ensure that you install the database client on the machine on which you want to run the Content Management Service.

IBM DB2 Database Requirements

Use the following guidelines when you set up the repository on IBM DB2:

- Verify that the database user account has CREATETAB and CONNECT privileges.
- Verify that the database user has SELECT privileges on the SYSCAT.DBAUTH and SYSCAT.DBTAUTH tables.
- Informatica does not support IBM DB2 table aliases for repository tables. Verify that table aliases have not been created for any tables in the database.
- Set the tablespace pageSize parameter to 32768 bytes.
- Set the NPAGES parameter to at least 5000. The NPAGES parameter determines the number of pages in the tablespace.

Microsoft SQL Server Database Requirements

Use the following guidelines when you set up the repository on Microsoft SQL Server:

- Verify that the database user account has CONNECT and CREATE TABLE privileges.

Oracle Database Requirements

Use the following guidelines when you set up the repository on Oracle:

- Verify that the database user has the following privileges:
 - ALTER SEQUENCE
 - ALTER TABLE
 - CREATE SEQUENCE
 - CREATE SESSION
 - CREATE TABLE
 - CREATE VIEW
 - DROP SEQUENCE
 - DROP TABLE
- Informatica does not support Oracle public synonyms for repository tables. Verify that public synonyms have not been created for any tables in the database.

Workflow Database Requirements

The Data Integration Service stores run-time metadata for workflows in the workflow database. Before you create the workflow database, set up a database and database user account for the workflow database.

You specify the workflow database connection when you create the Data Integration Service.

The workflow database supports the following database types:

- IBM DB2 UDB
- Microsoft SQL Server
- Oracle

Allow 200 MB of disk space for the database.

Note: Ensure that you install the database client on the machine on which you want to run the Data Integration Service.

IBM DB2 Database Requirements

Use the following guidelines when you set up the repository on IBM DB2:

- Verify that the database user account has CREATETAB and CONNECT privileges.
- Informatica does not support IBM DB2 table aliases for repository tables. Verify that table aliases have not been created for any tables in the database.
- Set the tablespace pageSize parameter to 32768 bytes.
- Set the NPAGES parameter to at least 5000. The NPAGES parameter determines the number of pages in the tablespace.
- Set the connection pooling parameters.

The following table lists the connection pooling parameters that you must set:

Parameter	Value
Maximum Connection Pool Size	128
Minimum Connection Pool Size	0
Maximum Idle Time	120 seconds

Microsoft SQL Server Database Requirements

Use the following guidelines when you set up the repository on Microsoft SQL Server:

- Verify that the database user account has CONNECT and CREATE TABLE privileges.
- Enable JTA and XA datasource functionality on the database.
- Set the connection pooling parameters.

The following table lists the connection pooling parameters that you must set:

Parameter	Value
Maximum Connection Pool Size	128
Minimum Connection Pool Size	0
Maximum Idle Time	120 seconds

Oracle Database Requirements

Use the following guidelines when you set up the repository on Oracle:

- Verify that the database user has the following privileges:
 - ALTER TABLE
 - ALTER VIEW
 - CREATE SEQUENCE
 - CREATE SESSION
 - CREATE SYNONYM
 - CREATE TABLE
 - CREATE VIEW
 - DROP TABLE
 - DROP VIEW
- Informatica does not support Oracle public synonyms for repository tables. Verify that public synonyms have not been created for any tables in the database.
- Set the connection pooling parameters.

The following table lists the connection pooling parameters that you must set:

Parameter	Value
Maximum Connection Pool Size	128
Minimum Connection Pool Size	0
Maximum Idle Time	120 seconds

Configure Native Connectivity on Service Machines

To establish native connectivity between an application service and a database, install the database client software for the database that you want to access.

Native drivers are packaged with the database server and client software. Configure connectivity on the machines that need to access the databases. To ensure compatibility between the application service and the database, install a client software that is compatible with the database version and use the appropriate database client libraries.

The following services use native connectivity to connect to different databases:

Data Integration Service

The Data Integration Service uses native database drivers to connect to the following databases:

- Source and target databases. Reads data from source databases and writes data to target databases.
- Data object cache database. Stores the data object cache.
- Profiling source databases. Reads from relational source databases to run profiles against the sources.
- Profiling warehouse. Writes the profiling results to the profiling warehouse.
- Reference tables. Runs mappings to transfer data between the reference tables and the external data sources.

When the Data Integration Service runs on a single node or on primary and back-up nodes, install database client software and configure connectivity on the machines where the Data Integration Service runs.

When the Data Integration Service runs on a grid, install database client software and configure connectivity on each machine that represents a node with the compute role or a node with both the service and compute roles.

PowerCenter Repository Service

The PowerCenter Repository Service uses native database drivers to connect to the PowerCenter repository database.

Install database client software and configure connectivity on the machines where the PowerCenter Repository Service and the PowerCenter Repository Service processes run.

PowerCenter Integration Service

The PowerCenter Integration Service uses native database drivers to connect to the following databases:

- Source and target databases. Reads from the source databases and writes to the target databases.
- Metadata Manager source databases. Loads the relational data sources in Metadata Manager.

Install database client software associated with the relational data sources and the repository databases on the machines where the PowerCenter Integration Service runs.

Install Database Client Software

You must install the database clients on the required machines based on the types of databases that the application services access.

To ensure compatibility between the application service and the database, use the appropriate database client libraries and install a client software that is compatible with the database version.

Install the following database client software based on the type of database that the application service accesses:

IBM DB2 Client Application Enabler (CAE)

Configure connectivity on the required machines by logging in to the machine as the user who starts Informatica services.

Microsoft SQL Server 2012 Native Client

Download the client from the following Microsoft website:
<http://www.microsoft.com/en-in/download/details.aspx?id=29065>.

Oracle client

Install compatible versions of the Oracle client and Oracle database server. You must also install the same version of the Oracle client on all machines that require it. To verify compatibility, contact Oracle.

Sybase Open Client (OCS)

Install an Open Client version that is compatible with the Sybase ASE database server. You must also install the same version of Open Client on the machines hosting the Sybase ASE database and Informatica. To verify compatibility, contact Sybase.

Configure Database Client Environment Variables on UNIX

Configure database client environment variables on the machines that run the Data Integration Service, PowerCenter Integration Service, and PowerCenter Repository Service processes.

The database client path variable name and requirements depend on the UNIX platform and the database.

After you configure the database environment variables, you can test the connection to the database from the database client.

The following table lists the database environment variables you need to set in UNIX:

Database	Environment Variable Name	Database Utility	Value
Oracle	ORACLE_HOME PATH	sqlplus	Set to: <DatabasePath> Add: <DatabasePath>/bin
IBM DB2	DB2DIR DB2INSTANCE PATH	db2connect	Set to: <DatabasePath> Set to: <DB2InstanceName> Add: <DatabasePath>/bin
Sybase ASE	SYBASE15 SYBASE_ASE SYBASE_OCS PATH	isql	Set to: <DatabasePath>/sybase<version> Set to: \${SYBASE15}/ASE-<version> Set to: \${SYBASE15}/OCS-<version> Add: \${SYBASE_ASE}/bin:\${SYBASE_OCS}/bin:\$PATH

APPENDIX B

Connecting to Databases from Windows

This appendix includes the following topics:

- [Connecting to Databases from Windows Overview, 404](#)
- [Connecting to an IBM DB2 Universal Database from Windows, 405](#)
- [Connecting to an Informix Database from Windows, 405](#)
- [Connecting to Microsoft Access and Microsoft Excel from Windows, 406](#)
- [Connecting to a Microsoft SQL Server Database from Windows, 406](#)
- [Connecting to a Netezza Database from Windows, 408](#)
- [Connecting to an Oracle Database from Windows, 408](#)
- [Connecting to a Sybase ASE Database from Windows, 410](#)
- [Connecting to a Teradata Database from Windows, 411](#)

Connecting to Databases from Windows Overview

Configure connectivity to enable communication between clients, services, and other components in the domain.

To use native connectivity, you must install and configure the database client software for the database that you want to access. To ensure compatibility between the application service and the database, install a client software that is compatible with the database version and use the appropriate database client libraries. To increase performance, use native connectivity.

The Informatica installation includes DataDirect ODBC drivers. If you have existing ODBC data sources created with an earlier version of the drivers, you must create new ODBC data sources using the new drivers. Configure ODBC connections using the DataDirect ODBC drivers provided by Informatica or third party ODBC drivers that are Level 2 compliant or higher.

The Informatica installation includes DataDirect JDBC drivers. You can use these drivers without performing additional steps. You can also download JDBC Type 4 drivers from third-party vendors to connect to sources and targets. You can use any third-party JDBC driver that is JDBC 3.0 or later.

You must configure a database connection for the following services in the Informatica domain:

- PowerCenter Repository Service
- Model Repository Service

- Data Integration Service
- Analyst Service

Connecting to an IBM DB2 Universal Database from Windows

For native connectivity, install the version of IBM DB2 Client Application Enabler (CAE) appropriate for the IBM DB2 database server version. To ensure compatibility between Informatica and databases, use the appropriate database client libraries.

Configuring Native Connectivity

You can configure native connectivity to an IBM DB2 database to increase performance.

The following steps provide a guideline for configuring native connectivity. For specific instructions, see the database documentation.

1. Verify that the following environment variable settings have been established by IBM DB2 Client Application Enabler (CAE):


```
DB2HOME=C:\IBM\SQLLIB
DB2INSTANCE=DB2
DB2CODEPAGE=1208 (Sometimes required. Use only if you encounter problems. Depends on the locale, you may use other values.)
```
2. Verify that the PATH environment variable includes the IBM DB2 bin directory. For example:


```
PATH=C:\WINNT\SYSTEM32;C:\SQLLIB\BIN;...
```
3. Configure the IBM DB2 client to connect to the database that you want to access. To configure the IBM DB2 client:
 - a. Launch the IBM DB2 Configuration Assistant.
 - b. Add the database connection.
 - c. Bind the connection.
4. Run the following command in the IBM DB2 Command Line Processor to verify that you can connect to the IBM DB2 database:


```
CONNECT TO <dbalias> USER <username> USING <password>
```
5. If the connection is successful, run the TERMINATE command to disconnect from the database. If the connection fails, see the database documentation.

Connecting to an Informix Database from Windows

Use ODBC to connect to an Informix database on Windows. Create an ODBC data source by using the DataDirect ODBC drivers installed with Informatica. To ensure compatibility between Informatica and databases, use the appropriate database client libraries.

Note: If you use the DataDirect ODBC driver provided by Informatica, you do not need the database client. The ODBC wire protocols do not require the database client software to connect to the database.

Configuring ODBC Connectivity

You can configure ODBC connectivity to an Informix database.

The following steps provide a guideline for configuring ODBC connectivity. For specific instructions, see the database documentation.

1. Create an ODBC data source using the DataDirect ODBC Wire Protocol driver for Informix provided by Informatica.
2. Verify that you can connect to the Informix database using the ODBC data source.

Connecting to Microsoft Access and Microsoft Excel from Windows

Configure connectivity to the Informatica components on Windows.

Install Microsoft Access or Excel on the machine where the Data Integration Service and PowerCenter Integration Service processes run. Create an ODBC data source for the Microsoft Access or Excel data you want to access.

Configuring ODBC Connectivity

You can configure ODBC connectivity to a Microsoft Access or Excel database.

The following steps provide a guideline for configuring ODBC connectivity. For specific instructions, see the database documentation.

1. Create an ODBC data source using the driver provided by Microsoft.
2. To avoid using empty string or nulls, use the reserved words PmNullUser for the user name and PmNullPasswd for the password when you create a database connection.

Connecting to a Microsoft SQL Server Database from Windows

On Informatica 10.0, you can connect to Microsoft SQL Server database through the ODBC provider type by default.

You can also connect to Microsoft SQL Server database using the OLEDB provider type, but the OLEDB provider type is deprecated. Support for the OLEDB provider type will be dropped in a future release.

Configuring Native Connectivity

On Informatica 10.0, you can configure native connectivity to the Microsoft SQL Server database by using the ODBC (default) or OLEDB (deprecated) provider types.

If you choose the ODBC provider type, you can enable the Use DSN option to use the DSN configured in the Microsoft ODBC Administrator as the connect string. If you do not enable the Use DSN option, you must specify the server name and database name in the connection properties.

If you choose the OLEDB provider type, you must install the Microsoft SQL Server 2012 Native Client to configure native connectivity to the Microsoft SQL Server database. If you cannot connect to the database, verify that you correctly entered all of the connectivity information.

You can download the Microsoft SQL Server 2012 Native Client from the following Microsoft website:
<http://www.microsoft.com/en-in/download/details.aspx?id=29065>.

After you upgrade, the Microsoft SQL Server connection is set to the OLEDB provider type by default. It is recommended that you upgrade all your Microsoft SQL Server connections to use the ODBC provider type. You can upgrade all your Microsoft SQL Server connections to the ODBC provider type by using the following commands:

- If you are using PowerCenter, run the following command: `pmrep upgradeSqlServerConnection`
- If you are using the Informatica platform, run the following command: `infacmd.sh isp upgradeSQLSConnection`

For specific connectivity instructions, see the database documentation.

Rules and Guidelines for Microsoft SQL Server

Consider the following rules and guidelines when you configure ODBC connectivity to a Microsoft SQL Server database:

- If you want to use a Microsoft SQL Server connection without using a Data Source Name (DSN less connection), you must configure the `odbcinst.ini` environment variable.
- If you are using a DSN connection, you must add the entry "EnableQuotedIdentifiers=1" to the ODBC DSN. If you do not add the entry, data preview and mapping run fail.
- You can use the Microsoft SQL Server NTLM authentication on a DSN less Microsoft SQL Server connection on the Microsoft Windows platform.
- If the Microsoft SQL Server table contains a UUID data type and if you are reading data from an SQL table and writing data to a flat file, the data format might not be consistent between the OLE DB and ODBC connection types.
- You cannot use SSL connection on a DSN less connection. If you want to use SSL, you must use the DSN connection. Enable the Use DSN option and configure the SSL options in the `odbc.ini` file.
- If the Microsoft SQL Server uses Kerberos authentication, you must set the `GSSClient` property to point to the Informatica Kerberos libraries. Use the following path and filename: `<Informatica installation directory>/server/bin/libgssapi_krb5.so.2`. Create an entry for the `GSSClient` property in the DSN entries section in `odbc.ini` for a DSN connection or in the SQL Server wire protocol section in `odbcinst.ini` for a connection that does not use DSN.

Configuring Custom Properties for Microsoft SQL Server

You can configure custom properties for Microsoft SQL Server to improve bulk load performance.

1. Launch the PowerCenter client and connect to Workflow Manager.
2. Open a workflow and select a session that you want to configure.
3. Click the **Config Object** tab.
4. Change the value of the **Default Buffer Block** size to 5 MB. You can also use the following command:

```
$INFA_HOME/server/bin/.pmrep massupdate -t session_config_property -n "Default buffer block size" -v "5MB" -f $<folderName>
```

To get optimum throughput for a row size of 1 KB, you must set the Buffer Block size to 5 MB.
5. Click the **Properties** tab.

6. Change the **Commit Interval** to 100000 if the session contains a relational target.
7. Set the **DTM Buffer Size**. The optimum DTM Buffer Size is ((10 x Block Buffer size) x number of partitions).

Connecting to a Netezza Database from Windows

Install and configure ODBC on the machines where the PowerCenter Integration Service process runs and where you install the PowerCenter Client. You must configure connectivity to the following Informatica components on Windows:

- **PowerCenter Integration Service.** Install the Netezza ODBC driver on the machine where the PowerCenter Integration Service process runs. Use the Microsoft ODBC Data Source Administrator to configure ODBC connectivity.
- **PowerCenter Client.** Install the Netezza ODBC driver on each PowerCenter Client machine that accesses the Netezza database. Use the Microsoft ODBC Data Source Administrator to configure ODBC connectivity. Use the Workflow Manager to create a database connection object for the Netezza database.

Configuring ODBC Connectivity

You can configure ODBC connectivity to a Netezza database.

The following steps provide a guideline for configuring ODBC connectivity. For specific instructions, see the database documentation.

1. Create an ODBC data source for each Netezza database that you want to access.

To create the ODBC data source, use the driver provided by Netezza.

Create a System DSN if you start the Informatica service with a Local System account logon. Create a User DSN if you select the This account log in option to start the Informatica service.

After you create the data source, configure the properties of the data source.

2. Enter a name for the new ODBC data source.
3. Enter the IP address/host name and port number for the Netezza server.
4. Enter the name of the Netezza schema where you plan to create database objects.
5. Configure the path and file name for the ODBC log file.
6. Verify that you can connect to the Netezza database.

You can use the Microsoft ODBC Data Source Administrator to test the connection to the database. To test the connection, select the Netezza data source and click Configure. On the Testing tab, click Test Connection and enter the connection information for the Netezza schema.

Connecting to an Oracle Database from Windows

For native connectivity, install the version of Oracle client appropriate for the Oracle database server version. To ensure compatibility between Informatica and databases, use the appropriate database client libraries.

You must install compatible versions of the Oracle client and Oracle database server. You must also install the same version of the Oracle client on all machines that require it. To verify compatibility, contact Oracle.

Configuring Native Connectivity

You can configure native connectivity to an Oracle database to increase performance.

The following steps provide a guideline for configuring native connectivity using Oracle Net Services or Net8. For specific connectivity instructions, see the database documentation.

1. Verify that the Oracle home directory is set.

For example:

```
ORACLE_HOME=C:\Oracle
```

2. Verify that the PATH environment variable includes the Oracle bin directory.

For example, if you install Net8, the path might include the following entry:

```
PATH=C:\ORANT\BIN;
```

3. Configure the Oracle client to connect to the database that you want to access.

Launch SQL*Net Easy Configuration Utility or edit an existing `tnsnames.ora` file to the home directory and modify it.

Note: By default, the `tnsnames.ora` file is stored in the following directory: `<OracleInstallationDir>\network\admin`.

Enter the correct syntax for the Oracle connect string, typically `databasename.world`. Make sure the SID entered here matches the database server instance ID defined on the Oracle server.

Here is a sample `tnsnames.ora` file. Enter the information for the database.

```
mydatabase.world =
  (DESCRIPTION
    (ADDRESS_LIST =
      (ADDRESS =
        (COMMUNITY = mycompany.world
          (PROTOCOL = TCP)
          (Host = mymachine)
          (Port = 1521)
        )
      )
    )
  (CONNECT_DATA =
    (SID = MYORA7)
    (GLOBAL_NAMES = mydatabase.world)
```

4. Set the NLS_LANG environment variable to the locale, including language, territory, and character set, you want the database client and server to use with the login.

The value of this variable depends on the configuration. For example, if the value is `american_america.UTF8`, you must set the variable as follows:

```
NLS_LANG=american_america.UTF8;
```

To determine the value of this variable, contact the database administrator.

5. To set the default session time zone when the Data Integration Service reads or writes the Timestamp with Local Time Zone data, specify the ORA_SDTZ environment variable.

You can set the ORA_SDTZ environment variable to any of the following values:

- Operating system local time zone ('OS_TZ')
- Database time zone ('DB_TZ')
- Absolute offset from UTC (for example, '-05:00')
- Time zone region name (for example, 'America/Los_Angeles')

You can set the environment variable at the machine where Informatica server runs.

6. If the `tnsnames.ora` file is not in the same location as the Oracle client installation location, set the `TNS_ADMIN` environment variable to the directory where the `tnsnames.ora` file resides.

For example, if the `tnsnames.ora` file is in the `C:\oracle\files` directory, set the variable as follows:

```
TNS_ADMIN= C:\oracle\files
```

7. Verify that you can connect to the Oracle database.

To connect to the database, launch SQL*Plus and enter the connectivity information. If you fail to connect to the database, verify that you correctly entered all of the connectivity information.

Use the connect string as defined in the `tnsnames.ora` file.

Connecting to a Sybase ASE Database from Windows

For native connectivity, install the version of Open Client appropriate for your database version. To ensure compatibility between Informatica and databases, use the appropriate database client libraries.

Install an Open Client version that is compatible with the Sybase ASE database server. You must also install the same version of Open Client on the machines hosting the Sybase ASE database and Informatica. To verify compatibility, contact Sybase.

If you want to create, restore, or upgrade a Sybase ASE repository, set *allow nulls by default* to TRUE at the database level. Setting this option changes the default null type of the column to null in compliance with the SQL standard.

Configuring Native Connectivity

You can configure native connectivity to a Sybase ASE database to increase performance.

The following steps provide a guideline for configuring native connectivity. For specific instructions, see the database documentation.

1. Verify that the `SYBASE` environment variable refers to the Sybase ASE directory.

For example:

```
SYBASE=C:\SYBASE
```

2. Verify that the `PATH` environment variable includes the Sybase OCS directory.

For example:

```
PATH=C:\SYBASE\OCS-15_0\BIN;C:\SYBASE\OCS-15_0\DLL
```

3. Configure Sybase Open Client to connect to the database that you want to access.

Use SQLEDT to configure the Sybase client, or copy an existing `SQL.INI` file (located in the `%SYBASE%\INI` directory) and make any necessary changes.

Select `NLWNSCK` as the Net-Library driver and include the Sybase ASE server name.

Enter the host name and port number for the Sybase ASE server. If you do not know the host name and port number, check with the system administrator.

4. Verify that you can connect to the Sybase ASE database.

To connect to the database, launch ISQL and enter the connectivity information. If you fail to connect to the database, verify that you correctly entered all of the connectivity information.

User names and database names are case sensitive.

Connecting to a Teradata Database from Windows

Install and configure native client software on the machines where the Data Integration Service and PowerCenter Integration Service process runs and where you install Informatica Developer and the PowerCenter Client. To ensure compatibility between Informatica and databases, use the appropriate database client libraries. You must configure connectivity to the following Informatica components on Windows:

- **Integration Service.** Install the Teradata client, the Teradata ODBC driver, and any other Teradata client software that you might need on the machine where the Data Integration Service and PowerCenter Integration Service run. You must also configure ODBC connectivity.
- **Informatica Developer.** Install the Teradata client, the Teradata ODBC driver, and any other Teradata client software that you might need on each machine that hosts a Developer tool that accesses Teradata. You must also configure ODBC connectivity.
- **PowerCenter Client.** Install the Teradata client, the Teradata ODBC driver, and any other Teradata client software that you might need on each PowerCenter Client machine that accesses Teradata. Use the Workflow Manager to create a database connection object for the Teradata database.

Note: Based on a recommendation from Teradata, Informatica uses ODBC to connect to Teradata. ODBC is a native interface for Teradata.

Configuring ODBC Connectivity

You can configure ODBC connectivity to a Teradata database.

The following steps provide a guideline for configuring ODBC connectivity. For specific instructions, see the database documentation.

1. Create an ODBC data source for each Teradata database that you want to access.
To create the ODBC data source, use the driver provided by Teradata.
Create a System DSN if you start the Informatica service with a *Local System account* logon. Create a User DSN if you select the *This account* log in option to start the Informatica service.
2. Enter the name for the new ODBC data source and the name of the Teradata server or its IP address.
To configure a connection to a single Teradata database, enter the DefaultDatabase name. To create a single connection to the default database, enter the user name and password. To connect to multiple databases, using the same ODBC data source, leave the DefaultDatabase field and the user name and password fields empty.
3. Configure Date Options in the Options dialog box.
In the Teradata Options dialog box, specify AAA for DateTime Format.
4. Configure Session Mode in the Options dialog box.
When you create a target data source, choose ANSI session mode. If you choose ANSI session mode, Teradata does not roll back the transaction when it encounters a row error. If you choose Teradata session mode, Teradata rolls back the transaction when it encounters a row error. In Teradata mode, the Integration Service cannot detect the rollback and does not report this in the session log.
5. Verify that you can connect to the Teradata database.
To test the connection, use a Teradata client program, such as WinDDI, BTEQ, Teradata Administrator, or Teradata SQL Assistant.

APPENDIX C

Connecting to Databases from UNIX

This appendix includes the following topics:

- [Connecting to Databases from UNIX Overview, 412](#)
- [Connecting to an IBM DB2 Universal Database from UNIX, 413](#)
- [Connecting to an Informix Database from UNIX, 415](#)
- [Connecting to Microsoft SQL Server from UNIX, 416](#)
- [Connecting to a Netezza Database from UNIX, 418](#)
- [Connecting to an Oracle Database from UNIX, 420](#)
- [Connecting to a Sybase ASE Database from UNIX, 422](#)
- [Connecting to a Teradata Database from UNIX, 424](#)
- [Connecting to an ODBC Data Source, 426](#)
- [Sample odbc.ini File, 428](#)

Connecting to Databases from UNIX Overview

To use native connectivity, you must install and configure the database client software for the database that you want to access. To ensure compatibility between the application service and the database, install a client software that is compatible with the database version and use the appropriate database client libraries. To increase performance, use native connectivity.

The Informatica installation includes DataDirect ODBC drivers. If you have existing ODBC data sources created with an earlier version of the drivers, you must create new ODBC data sources using the new drivers. Configure ODBC connections using the DataDirect ODBC drivers provided by Informatica or third party ODBC drivers that are Level 2 compliant or higher.

You must configure a database connection for the following services in the Informatica domain:

- PowerCenter Repository Service
- Model Repository Service
- Data Integration Service
- Analyst Service

When you connect to databases from Linux or UNIX, use native drivers to connect to IBM DB2, Oracle, or Sybase ASE databases. You can use ODBC to connect to other sources and targets.

Connecting to an IBM DB2 Universal Database from UNIX

For native connectivity, install the version of IBM DB2 Client Application Enabler (CAE) appropriate for the IBM DB2 database server version. To ensure compatibility between Informatica and databases, use the appropriate database client libraries.

Configuring Native Connectivity

You can configure native connectivity to an IBM DB2 database to increase performance.

The following steps provide a guideline for configuring native connectivity. For specific instructions, see the database documentation.

1. To configure connectivity on the machine where the Data Integration Service, PowerCenter Integration Service, or PowerCenter Repository Service process runs, log in to the machine as a user who can start a service process.

2. Set the DB2INSTANCE, INSTHOME, DB2DIR, and PATH environment variables.

The UNIX IBM DB2 software always has an associated user login, often db2admin, which serves as a holder for database configurations. This user holds the instance for DB2.

DB2INSTANCE. The name of the instance holder.

Using a Bourne shell:

```
$ DB2INSTANCE=db2admin; export DB2INSTANCE
```

Using a C shell:

```
$ setenv DB2INSTANCE db2admin
```

INSTHOME. This is db2admin home directory path.

Using a Bourne shell:

```
$ INSTHOME=~db2admin
```

Using a C shell:

```
$ setenv INSTHOME ~db2admin>
```

DB2DIR. Set the variable to point to the IBM DB2 CAE installation directory. For example, if the client is installed in the /opt/IBM/db2/V9.7 directory:

Using a Bourne shell:

```
$ DB2DIR=/opt/IBM/db2/V9.7; export DB2DIR
```

Using a C shell:

```
$ setenv DB2DIR /opt/IBM/db2/V9.7
```

PATH. To run the IBM DB2 command line programs, set the variable to include the DB2 bin directory.

Using a Bourne shell:

```
$ PATH=${PATH}:$DB2DIR/bin; export PATH
```

Using a C shell:

```
$ setenv PATH ${PATH}:$DB2DIR/bin
```

3. Set the shared library variable to include the DB2 lib directory.

The IBM DB2 client software contains a number of shared library components that the Data Integration Service, PowerCenter Integration Service, and PowerCenter Repository Service processes load

dynamically. Set the shared library environment variable so that the services can find the shared libraries at run time.

The shared library path must also include the Informatica installation directory (*server_dir*).

Set the shared library environment variable based on the operating system.

The following table describes the shared library variables for each operating system:

Operating System	Variable
Solaris	LD_LIBRARY_PATH
Linux	LD_LIBRARY_PATH
AIX	LIBPATH

For example, use the following syntax for Solaris and Linux:

- Using a Bourne shell:

```
$ LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:$HOME/server_dir:$DB2DIR/lib; export
LD_LIBRARY_PATH
```

- Using a C shell:

```
$ setenv LD_LIBRARY_PATH ${LD_LIBRARY_PATH}:$HOME/server_dir:$DB2DIR/lib
```

For AIX:

- Using a Bourne shell:

```
$ LIBPATH=${LIBPATH}:$HOME/server_dir:$DB2DIR/lib; export LIBPATH
```

- Using a C shell:

```
$ setenv LIBPATH ${LIBPATH}:$HOME/server_dir:$DB2DIR/lib
```

4. Edit the .cshrc or .profile to include the complete set of shell commands. Save the file and either log out and log in again or run the source command.

Using a Bourne shell:

```
$ source .profile
```

Using a C shell:

```
$ source .cshrc
```

5. If the DB2 database resides on the same machine on which the Data Integration Service, PowerCenter Integration Service, or PowerCenter Repository Service process runs, configure the DB2 instance as a remote instance.

Run the following command to verify if there is a remote entry for the database:

```
DB2 LIST DATABASE DIRECTORY
```

The command lists all the databases that the DB2 client can access and their configuration properties. If this command lists an entry for "Directory entry type" of "Remote," skip to [7](#).

6. If the database is not configured as remote, run the following command to verify whether a TCP/IP node is cataloged for the host:

```
DB2 LIST NODE DIRECTORY
```

If the node name is empty, you can create one when you set up a remote database. Use the following command to set up a remote database and, if needed, create a node:

```
db2 CATALOG TCPIP NODE <nodename> REMOTE <hostname_or_address> SERVER <port number>
```

Run the following command to catalog the database:

```
db2 CATALOG DATABASE <dbname> as <dbalias> at NODE <nodename>
```

For more information about these commands, see the database documentation.

7. Verify that you can connect to the DB2 database. Run the DB2 Command Line Processor and run the command:

```
CONNECT TO <dbalias> USER <username> USING <password>
```

If the connection is successful, clean up with the `CONNECT RESET` or `TERMINATE` command.

Connecting to an Informix Database from UNIX

Use ODBC to connect to an Informix database on UNIX.

Configuring ODBC Connectivity

You can configure ODBC connectivity to an Informix database.

The following steps provide a guideline for configuring ODBC connectivity. For specific instructions, see the database documentation.

1. Set the `ODBCHOME` environment variable to the ODBC installation directory. For example:

Using a Bourne shell:

```
$ ODBCHOME=<Informatica server home>/ODBC7.1; export ODBCHOME
```

Using a C shell:

```
$ setenv ODBCHOME <Informatica server home>/ODBC7.1
```

2. Set the `ODBCINI` environment variable to the location of the `odbc.ini` file. For example, if the `odbc.ini` file is in the `$ODBCHOME` directory:

Using a Bourne shell:

```
ODBCINI=$ODBCHOME/odbc.ini; export ODBCINI
```

Using a C shell:

```
$ setenv ODBCINI $ODBCHOME/odbc.ini
```

3. Edit the existing `odbc.ini` file in the `$ODBCHOME` directory or copy this `odbc.ini` file to the UNIX home directory and edit it.

```
$ cp $ODBCHOME/odbc.ini $HOME/.odbc.ini
```

4. Add an entry for the Informix data source under the section [ODBC Data Sources] and configure the data source. For example:

```
[Informix Wire Protocol]
Driver=/export/home/Informatica/10.0.0/ODBC7.1/lib/DWifcl27.so
Description=DataDirect 7.1 Informix Wire Protocol
AlternateServers=
ApplicationUsingThreads=1
CancelDetectInterval=0
ConnectionRetryCount=0
ConnectionRetryDelay=3
Database=<database_name>
HostName=<Informix_host>
LoadBalancing=0
LogonID=
Password=
PortNumber=<Informix_server_port>
```

```
ReportCodePageConversionErrors=0
ServerName=<Informix_server>
TrimBlankFromIndexName=1
```

5. Set the PATH and shared library environment variables by executing the script `odbc.sh` or `odbc.csh` in the `$ODBCHOME` directory.

Using a Bourne shell:

```
sh odbc.sh
```

Using a C shell:

```
source odbc.csh
```

6. Verify that you can connect to the Informix database using the ODBC data source. If the connection fails, see the database documentation.

Connecting to Microsoft SQL Server from UNIX

Use the Microsoft SQL Server connection to connect to a Microsoft SQL Server database from a UNIX machine.

Configuring Native Connectivity

You must choose ODBC as the provider type while configuring a Microsoft SQL Server connection. The OLEDB provider type is deprecated. Support for the OLEDB provider type will be dropped in a future release.

The server name and database name are retrieved from the connect string if you enable the Use DSN option. The connect string is the DSN configured in the `odbc.ini` file. If you do not enable the Use DSN option, you must specify the server name and database name in the connection properties. If you cannot to connect to the database, verify that you correctly entered all of the connectivity information.

After you upgrade, the Microsoft SQL Server connection is set to the OLEDB provider type by default. It is recommended that you upgrade all your Microsoft SQL Server connections to use the ODBC provider type. You can upgrade all your Microsoft SQL Server connections to the ODBC provider type by using the following commands:

- If you are using PowerCenter, run the following command: `pmrep upgradeSqlServerConnection`
- If you are using the Informatica platform, run the following command: `infacmd.sh isp upgradeSQLSConnection`

After you run the upgrade command, you must set the environment variable on each machine that hosts the Developer tool and on the machine that hosts Informatica services in the following format:

```
ODBCINST=<INFA_HOME>/ODBC7.1/odbcinst.ini
```

After you set the environment variable, you must restart the node that hosts the Informatica services.

For specific connectivity instructions, see the database documentation.

Rules and Guidelines for Microsoft SQL Server

Consider the following rules and guidelines when you configure ODBC connectivity to a Microsoft SQL Server database:

- If you want to use a Microsoft SQL Server connection without using a Data Source Name (DSN less connection), you must configure the `odbcinst.ini` environment variable.

- If you are using a DSN connection, you must add the entry "EnableQuotedIdentifiers=1" to the ODBC DSN. If you do not add the entry, data preview and mapping run fail.
- You can use the Microsoft SQL Server NTLM authentication on a DSN less Microsoft SQL Server connection on the Microsoft Windows platform.
- If the Microsoft SQL Server table contains a UUID data type and if you are reading data from an SQL table and writing data to a flat file, the data format might not be consistent between the OLE DB and ODBC connection types.
- You cannot use SSL connection on a DSN less connection. If you want to use SSL, you must use the DSN connection. Enable the Use DSN option and configure the SSL options in the `odbc.ini` file.
- If the Microsoft SQL Server uses Kerberos authentication, you must set the GSSClient property to point to the Informatica Kerberos libraries. Use the following path and filename: `<Informatica installation directory>/server/bin/libgssapi_krb5.so.2`. Create an entry for the GSSClient property in the DSN entries section in `odbc.ini` for a DSN connection or in the SQL Server wire protocol section in `odbcinst.ini` for a connection that does not use DSN.

Configuring SSL Authentication through ODBC

You can configure SSL authentication for Microsoft SQL Server through ODBC using the DataDirect New SQL Server Wire Protocol driver.

1. Open the `odbc.ini` file and add an entry for the ODBC data source and DataDirect New SQL Server Wire Protocol driver under the section [ODBC Data Sources].
2. Add the attributes in the `odbc.ini` file for configuring SSL.

The following table lists the attributes that you must add to the `odbc.ini` file when you configure SSL authentication:

Attribute	Description
EncryptionMethod	The method that the driver uses to encrypt the data sent between the driver and the database server. Set the value to 1 to encrypt data using SSL.
ValidateServerCertificate	Determines whether the driver validates the certificate sent by the database server when SSL encryption is enabled. Set the value to 1 for the driver to validate the server certificate.
TrustStore	The location and name of the trust store file. The trust store file contains a list of Certificate Authorities (CAs) that the driver uses for SSL server authentication.
TrustStorePassword	The password to access the contents of the trust store file.
HostNameInCertificate	Optional. The host name that is established by the SSL administrator for the driver to validate the host name contained in the certificate.

Configuring Custom Properties for Microsoft SQL Server

You can configure custom properties for Microsoft SQL Server to improve bulk load performance.

1. Launch the PowerCenter client and connect to Workflow Manager.
2. Open a workflow and select a session that you want to configure.
3. Click the **Config Object** tab.

4. Change the value of the **Default Buffer Block** size to 5 MB. You can also use the following command:
`$INFA_HOME/server/bin/./pmrep massupdate -t session_config_property -n "Default buffer block size" -v "5MB" -f $<folderName>`
 To get optimum throughput for a row size of 1 KB, you must set the Buffer Block size to 5 MB.
5. Click the **Properties** tab.
6. Change the **Commit Interval** to 100000 if the session contains a relational target.
7. Set the **DTM Buffer Size**. The optimum DTM Buffer Size is ((10 x Block Buffer size) x number of partitions).

Connecting to a Netezza Database from UNIX

Install and configure Netezza ODBC driver on the machine where the PowerCenter Integration Service process runs. Use the DataDirect Driver Manager in the DataDirect driver package shipped with the Informatica product to configure the Netezza data source details in the `odbc.ini` file.

Configuring ODBC Connectivity

You can configure ODBC connectivity to a Netezza database.

The following steps provide a guideline for configuring ODBC connectivity. For specific instructions, see the database documentation.

1. To configure connectivity for the integration service process, log in to the machine as a user who can start a service process.
2. Set the `ODBCHOME`, `NZ_ODBC_INI_PATH`, and `PATH` environment variables.

ODBCHOME. Set the variable to the ODBC installation directory. For example:

Using a Bourne shell:

```
$ ODBCHOME=<Informatica server home>/ODBC7.1; export ODBCHOME
```

Using a C shell:

```
$ setenv ODBCHOME =<Informatica server home>/ODBC7.1
```

PATH. Set the variable to the `ODBCHOME/bin` directory. For example:

Using a Bourne shell:

```
PATH="${PATH}:${ODBCHOME}/bin"
```

Using a C shell:

```
$ setenv PATH ${PATH}:${ODBCHOME}/bin
```

NZ_ODBC_INI_PATH. Set the variable to point to the directory that contains the `odbc.ini` file. For example, if the `odbc.ini` file is in the `$ODBCHOME` directory:

Using a Bourne shell:

```
NZ_ODBC_INI_PATH=$ODBCHOME; export NZ_ODBC_INI_PATH
```

Using a C shell:

```
$ setenv NZ_ODBC_INI_PATH $ODBCHOME
```

3. Set the shared library environment variable.

The shared library path must contain the ODBC libraries. It must also include the Informatica services installation directory (`server_dir`).

Set the shared library environment variable based on the operating system. Set the Netezza library folder to `<NetezzaInstallationDir>/lib64`.

The following table describes the shared library variables for each operating system:

Operating System	Variable
Solaris	LD_LIBRARY_PATH
Linux	LD_LIBRARY_PATH
AIX	LIBPATH

For example, use the following syntax for Solaris and Linux:

- Using a Bourne shell:

```
$ LD_LIBRARY_PATH="${LD_LIBRARY_PATH}:%HOME/server_dir:%ODBCHOME/
lib:<NetezzaInstallationDir>/lib64"
export LD_LIBRARY_PATH
```

- Using a C shell:

```
$ setenv LD_LIBRARY_PATH "${LD_LIBRARY_PATH}:%HOME/server_dir:%ODBCHOME/
lib:<NetezzaInstallationDir>/lib64"
```

For AIX

- Using a Bourne shell:

```
$ LIBPATH=${LIBPATH}:%HOME/server_dir:%ODBCHOME/lib:<NetezzaInstallationDir>/
lib64; export LIBPATH
```

- Using a C shell:

```
$ setenv LIBPATH ${LIBPATH}:%HOME/server_dir:%ODBCHOME/
lib:<NetezzaInstallationDir>/lib64
```

- Edit the existing `odbc.ini` file or copy the `odbc.ini` file to the home directory and edit it.

This file exists in `%ODBCHOME` directory.

```
$ cp %ODBCHOME/odbc.ini %HOME/.odbc.ini
```

Add an entry for the Netezza data source under the section `[ODBC Data Sources]` and configure the data source.

For example:

```
[NZSQL]
Driver = /export/home/appsga/thirdparty/netezza/lib64/libnzodbc.so
Description = NetezzaSQL ODBC
Servername = netezza1.informatica.com
Port = 5480
Database = infa
Username = admin
Password = password
Debuglogging = true
StripCRLF = false
PreFetch = 256
Protocol = 7.0
ReadOnly = false
ShowSystemTables = false
Socket = 16384
DateFormat = 1
TranslationDLL =
TranslationName =
```

```
TranslationOption =  
NumericAsChar = false
```

For more information about Netezza connectivity, see the Netezza ODBC driver documentation.

5. Verify that the last entry in the `odbc.ini` file is `InstallDir` and set it to the ODBC installation directory.

For example:

```
InstallDir=<Informatica install directory>/<ODBCHOME directory>
```

6. Edit the `.cshrc` or `.profile` file to include the complete set of shell commands.
7. Restart the Informatica services.

Connecting to an Oracle Database from UNIX

For native connectivity, install the version of Oracle client appropriate for the Oracle database server version. To ensure compatibility between Informatica and databases, use the appropriate database client libraries.

You must install compatible versions of the Oracle client and Oracle database server. You must also install the same version of the Oracle client on all machines that require it. To verify compatibility, contact Oracle.

Configuring Native Connectivity

You can configure native connectivity to an Oracle database to increase performance.

The following steps provide a guideline for configuring native connectivity through Oracle Net Services or Net8. For specific instructions, see the database documentation.

1. To configure connectivity for the Data Integration Service, PowerCenter Integration Service, or PowerCenter Repository Service process, log in to the machine as a user who can start the server process.
2. Set the `ORACLE_HOME`, `NLS_LANG`, `TNS_ADMIN`, and `PATH` environment variables.

ORACLE_HOME. Set the variable to the Oracle client installation directory. For example, if the client is installed in the `/HOME2/oracle` directory, set the variable as follows:

Using a Bourne shell:

```
$ ORACLE_HOME=/HOME2/oracle; export ORACLE_HOME
```

Using a C shell:

```
$ setenv ORACLE_HOME /HOME2/oracle
```

NLS_LANG. Set the variable to the locale (language, territory, and character set) you want the database client and server to use with the login. The value of this variable depends on the configuration. For example, if the value is `american_america.UTF8`, set the variable as follows:

Using a Bourne shell:

```
$ NLS_LANG=american_america.UTF8; export NLS_LANG
```

Using a C shell:

```
$ NLS_LANG american_america.UTF8
```

To determine the value of this variable, contact the administrator.

ORA_SDTZ. To set the default session time zone when the Data Integration Service reads or writes the Timestamp with Local Time Zone data, specify the `ORA_SDTZ` environment variable.

You can set the ORA_SDTZ environment variable to any of the following values:

- Operating system local time zone ('OS_TZ')
- Database time zone ('DB_TZ')
- Absolute offset from UTC (for example, '-05:00')
- Time zone region name (for example, 'America/Los_Angeles')

You can set the environment variable at the machine where Informatica server runs.

TNS_ADMIN. If the tnsnames.ora file is not in the same location as the Oracle client installation location, set the TNS_ADMIN environment variable to the directory where the tnsnames.ora file resides. For example, if the file is in the /HOME2/oracle/files directory, set the variable as follows:

Using a Bourne shell:

```
$ TNS_ADMIN=$HOME2/oracle/files; export TNS_ADMIN
```

Using a C shell:

```
$ setenv TNS_ADMIN=$HOME2/oracle/files
```

Note: By default, the tnsnames.ora file is stored in the following directory: \$ORACLE_HOME/network/admin.

PATH. To run the Oracle command line programs, set the variable to include the Oracle bin directory.

Using a Bourne shell:

```
$ PATH=${PATH}:$ORACLE_HOME/bin; export PATH
```

Using a C shell:

```
$ setenv PATH ${PATH}:ORACLE_HOME/bin
```

3. Set the shared library environment variable.

The Oracle client software contains a number of shared library components that the Data Integration Service, PowerCenter Integration Service, and PowerCenter Repository Service processes load dynamically. To locate the shared libraries during run time, set the shared library environment variable.

The shared library path must also include the Informatica installation directory (server_dir).

Set the shared library environment variable to LD_LIBRARY_PATH.

For example, use the following syntax:

- Using a Bourne shell:

```
$ LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:$HOME/server_dir:$ORACLE_HOME/lib; export LD_LIBRARY_PATH
```

- Using a C shell:

```
$ setenv LD_LIBRARY_PATH ${LD_LIBRARY_PATH}:$HOME/server_dir:$ORACLE_HOME/lib
```

4. Edit the .cshrc or .profile to include the complete set of shell commands. Save the file and either log out and log in again, or run the source command.

Using a Bourne shell:

```
$ source .profile
```

Using a C shell:

```
$ source .cshrc
```

5. Verify that the Oracle client is configured to access the database.

Use the SQL*Net Easy Configuration Utility or copy an existing tnsnames.ora file to the home directory and modify it.

The tnsnames.ora file is stored in the following directory: \$ORACLE_HOME/network/admin.

Enter the correct syntax for the Oracle connect string, typically databasename.world.

Here is a sample `tnsnames.ora` file. Enter the information for the database.

```
mydatabase.world =
  (DESCRIPTION
    (ADDRESS_LIST =
      (ADDRESS =
        (COMMUNITY = mycompany.world
          (PROTOCOL = TCP)
          (Host = mymachine)
          (Port = 1521)
        )
      )
    )
  (CONNECT_DATA =
    (SID = MYORA7)
    (GLOBAL_NAMES = mydatabase.world)
```

6. Verify that you can connect to the Oracle database.

To connect to the Oracle database, launch SQL*Plus and enter the connectivity information. If you fail to connect to the database, verify that you correctly entered all of the connectivity information.

Enter the user name and connect string as defined in the `tnsnames.ora` file.

Connecting to a Sybase ASE Database from UNIX

For native connectivity, install the version of Open Client appropriate for your database version. To ensure compatibility between Informatica and databases, use the appropriate database client libraries.

Install an Open Client version that is compatible with the Sybase ASE database server. You must also install the same version of Open Client on the machines hosting the Sybase ASE database and Informatica. To verify compatibility, contact Sybase.

If you want to create, restore, or upgrade a Sybase ASE repository, set *allow nulls by default* to TRUE at the database level. Setting this option changes the default null type of the column to null in compliance with the SQL standard.

Configuring Native Connectivity

You can configure native connectivity to a Sybase ASE database to increase performance.

The following steps provide a guideline for configuring native connectivity. For specific instructions, see the database documentation.

1. To configure connectivity to the Data Integration Service, PowerCenter Integration Service, or PowerCenter Repository Service process, log in to the machine as a user who can start the server process.
2. Set the SYBASE and PATH environment variables.

SYBASE. Set the variable to the Sybase Open Client installation directory. For example if the client is installed in the `/usr/sybase` directory:

Using a Bourne shell:

```
$ SYBASE=/usr/sybase; export SYBASE
```

Using a C shell:

```
$ setenv SYBASE /usr/sybase
```

PATH. To run the Sybase command line programs, set the variable to include the Sybase OCS bin directory.

Using a Bourne shell:

```
$ PATH=${PATH}:/usr/sybase/OCS-15_0/bin; export PATH
```

Using a C shell:

```
$ setenv PATH ${PATH}:/usr/sybase/OCS-15_0/bin
```

3. Set the shared library environment variable.

The Sybase Open Client software contains a number of shared library components that the Data Integration Service, PowerCenter Integration Service, and PowerCenter Repository Service processes load dynamically. Set the shared library environment variable so that the services can find the shared libraries at run time.

The shared library path must also include the installation directory of the Informatica services (*server_dir*).

Set the shared library environment variable based on the operating system.

The following table describes the shared library variables for each operating system.

Operating System	Variable
Solaris	LD_LIBRARY_PATH
Linux	LD_LIBRARY_PATH
AIX	LIBPATH

For example, use the following syntax for Solaris and Linux:

- Using a Bourne shell:

```
$ LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:${HOME}/server_dir:$SYBASE/OCS-15_0/lib;
$SYBASE/OCS-15_0/lib3p;$SYBASE/OCS-15_0/lib3p64; export LD_LIBRARY_PATH
```

- Using a C shell:

```
$ setenv LD_LIBRARY_PATH ${LD_LIBRARY_PATH}:${HOME}/server_dir:$SYBASE/OCS-15_0/lib;
$SYBASE/OCS-15_0/lib3p;$SYBASE/OCS-15_0/lib3p64;
```

For AIX

- Using a Bourne shell:

```
$ LIBPATH=${LIBPATH}:${HOME}/server_dir:$SYBASE/OCS-15_0/lib;$SYBASE/OCS-15_0/lib3p;
$SYBASE/OCS-15_0/lib3p64; export LIBPATH
```

- Using a C shell:

```
$ setenv LIBPATH ${LIBPATH}:${HOME}/server_dir:$SYBASE/OCS-15_0/lib;$SYBASE/
OCS-15_0/lib3p;$SYBASE/OCS-15_0/lib3p64;
```

4. Edit the .cshrc or .profile to include the complete set of shell commands. Save the file and either log out and log in again, or run the source command.

Using a Bourne shell:

```
$ source .profile
```

Using a C shell:

```
$ source .cshrc
```

5. Verify the Sybase ASE server name in the Sybase interfaces file stored in the \$SYBASE directory.

6. Verify that you can connect to the Sybase ASE database.

To connect to the Sybase ASE database, launch ISQL and enter the connectivity information. If you fail to connect to the database, verify that you correctly entered all of the connectivity information.

User names and database names are case sensitive.

Connecting to a Teradata Database from UNIX

Install and configure native client software on the machines where the Data Integration Service or PowerCenter Integration Service process runs. To ensure compatibility between Informatica and databases, use the appropriate database client libraries.

Install the Teradata client, the Teradata ODBC driver, and any other Teradata client software that you might need on the machine where the Data Integration Service or PowerCenter Integration Service runs. You must also configure ODBC connectivity.

Note: Based on a recommendation from Teradata, Informatica uses ODBC to connect to Teradata. ODBC is a native interface for Teradata.

Configuring ODBC Connectivity

You can configure ODBC connectivity to a Teradata database.

The following steps provide a guideline for configuring ODBC connectivity. For specific instructions, see the database documentation.

1. To configure connectivity for the integration service process, log in to the machine as a user who can start a service process.
2. Set the `TERADATA_HOME`, `ODBCHOME`, and `PATH` environment variables.

TERADATA_HOME. Set the variable to the Teradata driver installation directory. The defaults are as follows:

Using a Bourne shell:

```
$ TERADATA_HOME=/opt/teradata/client/<version>; export TERADATA_HOME
```

Using a C shell:

```
$ setenv TERADATA_HOME /opt/teradata/client/<version>
```

ODBCHOME. Set the variable to the ODBC installation directory. For example:

Using a Bourne shell:

```
$ ODBCHOME=$INFA_HOME/ODBC<version>; export ODBCHOME
```

Using a C shell:

```
$ setenv ODBCHOME $INFA_HOME/ODBC<version>
```

PATH. To run the `ddtestlib` utility, to verify that the DataDirect ODBC driver manager can load the driver files, set the variable as follows:

Using a Bourne shell:

```
PATH="{PATH}:%ODBCHOME/bin:$TERADATA_HOME/bin"
```

Using a C shell:

```
$ setenv PATH {PATH}:%ODBCHOME/bin:$TERADATA_HOME/bin
```

3. Set the shared library environment variable.

The Teradata software contains multiple shared library components that the integration service process loads dynamically. Set the shared library environment variable so that the services can find the shared libraries at run time.

The shared library path must also include installation directory of the Informatica service (*server_dir*).

Set the shared library environment variable based on the operating system.

The following table describes the shared library variables for each operating system:

Operating System	Variable
Solaris	LD_LIBRARY_PATH
Linux	LD_LIBRARY_PATH
AIX	LIBPATH

For example, use the following syntax for Solaris and Linux:

- Using a Bourne shell:

```
$ LD_LIBRARY_PATH="${LD_LIBRARY_PATH}:${HOME}/server_dir:$ODBCHOME/lib:
$TERADATA_HOME/lib64:$TERADATA_HOME/odbc_64/lib";

export LD_LIBRARY_PATH
```

- Using a C shell:

```
$ setenv LD_LIBRARY_PATH "${LD_LIBRARY_PATH}:${HOME}/server_dir:$ODBCHOME/lib:
$TERADATA_HOME/lib64:
$TERADATA_HOME/odbc_64/lib"
```

For AIX

- Using a Bourne shell:

```
$ LIBPATH=${LIBPATH}:${HOME}/server_dir:$ODBCHOME/lib:$TERADATA_HOME/
lib64:$TERADATA_HOME/odbc_64/lib; export LIBPATH
```

- Using a C shell:

```
$ setenv LIBPATH ${LIBPATH}:${HOME}/server_dir:$ODBCHOME/lib:$TERADATA_HOME/lib64:
$TERADATA_HOME/odbc_64/lib
```

- Edit the existing odbc.ini file or copy the odbc.ini file to the home directory and edit it.

This file exists in \$ODBCHOME directory.

```
$ cp $ODBCHOME/odbc.ini $HOME/.odbc.ini
```

Add an entry for the Teradata data source under the section [ODBC Data Sources] and configure the data source.

For example:

```
MY_TERADATA_SOURCE=Teradata Driver
[MY_TERADATA_SOURCE]
Driver=/u01/app/teradata/td-tuf611/odbc/drivers/tdata.so
Description=NCR 3600 running Teradata V1R5.2
DBCName=208.199.59.208
DateTimeFormat=AAA
SessionMode=ANSI
DefaultDatabase=
Username=
Password=
```

- Set the DateTimeFormat to AAA in the Teradata data ODBC configuration.

6. Optionally, set the SessionMode to ANSI. When you use ANSI session mode, Teradata does not roll back the transaction when it encounters a row error.

If you choose Teradata session mode, Teradata rolls back the transaction when it encounters a row error. In Teradata mode, the integration service process cannot detect the rollback, and does not report this in the session log.

7. To configure connection to a single Teradata database, enter the DefaultDatabase name. To create a single connection to the default database, enter the user name and password. To connect to multiple databases, using the same ODBC DSN, leave the DefaultDatabase field empty.

For more information about Teradata connectivity, see the Teradata ODBC driver documentation.

8. Verify that the last entry in the `odbc.ini` is `InstallDir` and set it to the `odbc` installation directory.

For example:

```
InstallDir=<Informatica installation directory>/ODBC<version>
```

9. Edit the `.cshrc` or `.profile` to include the complete set of shell commands.
10. Save the file and either log out and log in again, or run the `source` command.

Using a Bourne shell:

```
$ source .profile
```

Using a C shell:

```
$ source .cshrc
```

11. For each data source you use, make a note of the file name under the `Driver=<parameter>` in the data source entry in `odbc.ini`. Use the `ddtestlib` utility to verify that the DataDirect ODBC driver manager can load the driver file.

For example, if you have the driver entry:

```
Driver=/u01/app/teradata/td-tuf611/odbc/drivers/tdata.so
```

run the following command:

```
ddtestlib /u01/app/teradata/td-tuf611/odbc/drivers/tdata.so
```

12. Test the connection using BTEQ or another Teradata client tool.

Connecting to an ODBC Data Source

Install and configure native client software on the machine where the Data Integration Service, PowerCenter Integration Service, and PowerCenter Repository Service run. Also install and configure any underlying client access software required by the ODBC driver. To ensure compatibility between Informatica and the databases, use the appropriate database client libraries.

The Informatica installation includes DataDirect ODBC drivers. If the `odbc.ini` file contains connections that use earlier versions of the ODBC driver, update the connection information to use the new drivers. Use the System DSN to specify an ODBC data source on Windows.

1. On the machine where the application service runs, log in as a user who can start a service process.
2. Set the `ODBCHOME` and `PATH` environment variables.

ODBCHOME. Set to the DataDirect ODBC installation directory. For example, if the install directory is `/export/home/Informatica/10.0.0/ODBC7.1`.

Using a Bourne shell:

```
$ ODBCHOME=/export/home/Informatica/10.0.0/ODBC7.1; export ODBCHOME
```

Using a C shell:

```
$ setenv ODBC_HOME /export/home/Informatica/10.0.0/ODBC7.1
```

PATH. To run the ODBC command line programs, like *ddtestlib*, set the variable to include the *odbc bin* directory.

Using a Bourne shell:

```
$ PATH=${PATH}:${ODBC_HOME}/bin; export PATH
```

Using a C shell:

```
$ setenv PATH ${PATH}:${ODBC_HOME}/bin
```

Run the *ddtestlib* utility to verify that the DataDirect ODBC driver manager can load the driver files.

3. Set the shared library environment variable.

The ODBC software contains a number of shared library components that the service processes load dynamically. Set the shared library environment variable so that the services can find the shared libraries at run time.

The shared library path must also include the Informatica installation directory (*server_dir*).

Set the shared library environment variable based on the operating system.

The following table describes the shared library variables for each operating system:

Operating System	Variable
Solaris	LD_LIBRARY_PATH
Linux	LD_LIBRARY_PATH
AIX	LIBPATH

For example, use the following syntax for Solaris and Linux:

- Using a Bourne shell:

```
$ LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:${HOME}/server_dir:${ODBC_HOME}/lib; export LD_LIBRARY_PATH
```

- Using a C shell:

```
$ setenv LD_LIBRARY_PATH $HOME/server_dir:${ODBC_HOME}:${LD_LIBRARY_PATH}
```

For AIX

- Using a Bourne shell:

```
$ LIBPATH=${LIBPATH}:${HOME}/server_dir:${ODBC_HOME}/lib; export LIBPATH
```

- Using a C shell:

```
$ setenv LIBPATH ${LIBPATH}:${HOME}/server_dir:${ODBC_HOME}/lib
```

4. Edit the existing *odbc.ini* file or copy the *odbc.ini* file to the home directory and edit it.

This file exists in *\$ODBC_HOME* directory.

```
$ cp $ODBC_HOME/odbc.ini $HOME/.odbc.ini
```

Add an entry for the ODBC data source under the section [ODBC Data Sources] and configure the data source.

For example:

```
MY_MSSQLSERVER_ODBC_SOURCE=<Driver name or data source description>
[MY_MSSQLSERVER_ODBC_SOURCE]
Driver=<path to ODBC drivers>
Description=DataDirect 7.1 SQL Server Wire Protocol
```

```

Database=<SQLServer_database_name>
LogonID=<username>
Password=<password>
Address=<TCP/IP address>,<port number>
QuoteId=No
AnsiNPW=No
ApplicationsUsingThreads=1

```

This file might already exist if you have configured one or more ODBC data sources.

5. Verify that the last entry in the `odbc.ini` is `InstallDir` and set it to the `odbc` installation directory.

For example:

```
InstallDir=/export/home/Informatica/10.0.0/ODBC7.1
```

6. If you use the `odbc.ini` file in the home directory, set the `ODBCINI` environment variable.

Using a Bourne shell:

```
$ ODBCINI=$HOME/.odbc.ini; export ODBCINI
```

Using a C shell:

```
$ setenv ODBCINI $HOME/.odbc.ini
```

7. Edit the `.cshrc` or `.profile` to include the complete set of shell commands. Save the file and either log out and log in again, or run the source command.

Using a Bourne shell:

```
$ source .profile
```

Using a C shell:

```
$ source .cshrc
```

8. Use the `ddtestlib` utility to verify that the DataDirect ODBC driver manager can load the driver file you specified for the data source in the `odbc.ini` file.

For example, if you have the driver entry:

```
Driver = /export/home/Informatica/10.0.0/ODBC7.1/lib/DWxxxxnn.so
```

run the following command:

```
ddtestlib /export/home/Informatica/10.0.0/ODBC7.1/lib/DWxxxxnn.so
```

9. Install and configure any underlying client access software needed by the ODBC driver.

Note: While some ODBC drivers are self-contained and have all information inside the `.odbc.ini` file, most are not. For example, if you want to use an ODBC driver to access Sybase IQ, you must install the Sybase IQ network client software and set the appropriate environment variables.

To use the Informatica ODBC drivers (`DWxxxxnn.so`), manually set the `PATH` and shared library path environment variables. Alternatively, run the `odbc.sh` or `odbc.csh` script in the `$ODBCHOME` folder. This script will set the required `PATH` and shared library path environment variables for the ODBC drivers provided by Informatica.

Sample odbc.ini File

The following sample shows the entries for the ODBC drivers in the `ODBC.ini` file:

```

[ODBC Data Sources]
DB2 Wire Protocol=DataDirect 7.1 DB2 Wire Protocol
Informix Wire Protocol=DataDirect 7.1 Informix Wire Protocol
Oracle Wire Protocol=DataDirect 8.0 Oracle Wire Protocol
Sybase Wire Protocol=DataDirect 7.1 Sybase Wire Protocol
SQL Server Wire Protocol=DataDirect 8.0 SQL Server Wire Protocol
MySQL Wire Protocol=DataDirect 7.1 MySQL Wire Protocol
PostgreSQL Wire Protocol=DataDirect 7.1 PostgreSQL Wire Protocol

```

Greenplum Wire Protocol=DataDirect 7.1 Greenplum Wire Protocol

```
[ODBC]
IANAAppCodePage=4
InstallDir=/<Informatica installation directory>/ODBC7.1
Trace=0
TraceFile=odbctrace.out
TraceDll=/<Informatica installation directory>/ODBC7.1/lib/DWtrc27.so

[DB2 Wire Protocol]
Driver=/<Informatica installation directory>/ODBC7.1/lib/DWdb227.so
Description=DataDirect 7.1 DB2 Wire Protocol
AccountingInfo=
AddStringToCreateTable=
AlternateID=
AlternateServers=
ApplicationName=
ApplicationUsingThreads=1
AuthenticationMethod=0
BulkBinaryThreshold=32
BulkCharacterThreshold=-1
BulkLoadBatchSize=1024
BulkLoadFieldDelimiter=
BulkLoadRecordDelimiter=
CatalogSchema=
CharsetFor65535=0
ClientHostName=
ClientUser=
#Collection applies to z/OS and iSeries only
Collection=
ConcurrentAccessResolution=0
ConnectionReset=0
ConnectionRetryCount=0
ConnectionRetryDelay=3
CurrentFuncPath=
#Database applies to DB2 UDB only
Database=<database_name>
DefaultIsolationLevel=1
DynamicSections=1000
EnableBulkLoad=0
EncryptionMethod=0
FailoverGranularity=0
FailoverMode=0
FailoverPreconnect=0
GrantAuthid=PUBLIC
GrantExecute=1
GSSClient=native
HostNameInCertificate=
IpAddress=<DB2_server_host>
KeyPassword=
KeyStore=
KeyStorePassword=
LoadBalanceTimeout=0
LoadBalancing=0
#Location applies to z/OS and iSeries only
Location=<location_name>
LogonID=
MaxPoolSize=100
MinPoolSize=0
Password=
PackageCollection=NULLID
PackageNamePrefix=DD
PackageOwner=
Pooling=0
ProgramID=
QueryTimeout=0
ReportCodePageConversionErrors=0
TcpPort=50000
TrustStore=
TrustStorePassword=
UseCurrentSchema=0
```

```

ValidateServerCertificate=1
WithHold=1
XMLDescribeType=-10

[Informix Wire Protocol]
Driver=/<Informatica installation directory>/ODBC7.1/lib/DWifcl27.so
Description=DataDirect 7.1 Informix Wire Protocol
AlternateServers=
ApplicationUsingThreads=1
CancelDetectInterval=0
ConnectionRetryCount=0
ConnectionRetryDelay=3
Database=<database_name>
HostName=<Informix_host>
LoadBalancing=0
LogonID=
Password=
PortNumber=<Informix_server_port>
ServerName=<Informix_server>
TrimBlankFromIndexName=1
UseDelimitedIdentifiers=0

[Oracle Wire Protocol]
Driver=/<Informatica installation directory>/ODBC7.1/lib/DWora28.so
Description=DataDirect 8.0 Oracle Wire Protocol
AlternateServers=
ApplicationUsingThreads=1
AccountingInfo=
Action=
ApplicationName=
ArraySize=60000
AuthenticationMethod=1
BulkBinaryThreshold=32
BulkCharacterThreshold=-1
BulkLoadBatchSize=1024
BulkLoadFieldDelimiter=
BulkLoadRecordDelimiter=
CachedCursorLimit=32
CachedDescLimit=0
CatalogIncludesSynonyms=1
CatalogOptions=0
ClientHostName=
ClientID=
ClientUser=
ConnectionReset=0
ConnectionRetryCount=0
ConnectionRetryDelay=3
DataIntegrityLevel=0
DataIntegrityTypes=MD5,SHA1
DefaultLongDataBuffLen=1024
DescribeAtPrepare=0
EditionName=
EnableBulkLoad=0
EnableDescribeParam=0
EnableNcharSupport=0
EnableScrollableCursors=1
EnableStaticCursorsForLongData=0
EnableTimestampWithTimeZone=0
EncryptionLevel=0
EncryptionMethod=0
EncryptionTypes=AES128,AES192,AES256,DES,3DES112,3DES168,RC4_40,RC4_56,RC4_128,
RC4_256
FailoverGranularity=0
FailoverMode=0
FailoverPreconnect=0
FetchTSWTZasTimestamp=0
GSSClient=native
HostName=<Oracle_server>
HostNameInCertificate=
InitializationString=
KeyPassword=

```

```

KeyStore=
KeyStorePassword=
LoadBalanceTimeout=0
LoadBalancing=0
LocalTimeZoneOffset=
LockTimeOut=-1
LoginTimeout=15
LogonID=
MaxPoolSize=100
MinPoolSize=0
Module=
Password=
Pooling=0
PortNumber=<Oracle_server_port>
ProcedureRetResults=0
ProgramID=
QueryTimeout=0
ReportCodePageConversionErrors=0
ReportRecycleBin=0
ServerName=<server_name in tnsnames.ora>
ServerType=0
ServiceName=
SID=<Oracle_System_Identifier>
TimestampEscapeMapping=0
TNSNamesFile=<tnsnames.ora_filename>
TrustStore=
TrustStorePassword=
UseCurrentSchema=1
ValidateServerCertificate=1
WireProtocolMode=2

[Sybase Wire Protocol]
Driver=/<Informatica installation directory>/ODBC7.1/lib/DWase27.so
Description=DataDirect 7.1 Sybase Wire Protocol
AlternateServers=
ApplicationName=
ApplicationUsingThreads=1
ArraySize=50
AuthenticationMethod=0
BulkBinaryThreshold=32
BulkCharacterThreshold=-1
BulkLoadBatchSize=1024
BulkLoadFieldDelimiter=
BulkLoadRecordDelimiter=
Charset=
ConnectionReset=0
ConnectionRetryCount=0
ConnectionRetryDelay=3
CursorCacheSize=1
Database=<database_name>
DefaultLongDataBuffLen=1024
EnableBulkLoad=0
EnableDescribeParam=0
EnableQuotedIdentifiers=0
EncryptionMethod=0
FailoverGranularity=0
FailoverMode=0
FailoverPreconnect=0
GSSClient=native
HostNameInCertificate=
InitializationString=
Language=
LoadBalancing=0
LoadBalanceTimeout=0
LoginTimeout=15
LogonID=
MaxPoolSize=100
MinPoolSize=0
NetworkAddress=<Sybase_host,Sybase_server_port>
OptimizePrepare=1
PacketSize=0

```

```

Password=
Pooling=0
QueryTimeout=0
RaiseErrorPositionBehavior=0
ReportCodePageConversionErrors=0
SelectMethod=0
ServicePrincipalName=
TruncateTimeTypeFractions=0
TrustStore=
TrustStorePassword=
ValidateServerCertificate=1
WorkStationID=

[SQL Server Wire Protocol]
Driver=/<Informatica installation directory>/ODBC7.1/lib/DWsqls28.so
Description=DataDirect 8.0 SQL Server Wire Protocol
AlternateServers=
AlwaysReportTriggerResults=0
AnsiNPW=1
ApplicationName=
ApplicationUsingThreads=1
AuthenticationMethod=1
BulkBinaryThreshold=32
BulkCharacterThreshold=-1
BulkLoadBatchSize=1024
BulkLoadOptions=2
ConnectionReset=0
ConnectionRetryCount=0
ConnectionRetryDelay=3
Database=<database_name>
EnableBulkLoad=0
EnableQuotedIdentifiers=0
EncryptionMethod=0
FailoverGranularity=0
FailoverMode=0
FailoverPreconnect=0
FetchTSWTZasTimestamp=0
FetchTWFSasTime=1
GSSClient=native
HostName=<SQL_Server_host>
HostNameInCertificate=
InitializationString=
Language=
LoadBalanceTimeout=0
LoadBalancing=0
LoginTimeout=15
LogonID=
MaxPoolSize=100
MinPoolSize=0
PacketSize=-1
Password=
Pooling=0
PortNumber=<SQL_Server_server_port>
QueryTimeout=0
ReportCodePageConversionErrors=0
SnapshotSerializable=0
TrustStore=
TrustStorePassword=
ValidateServerCertificate=1
WorkStationID=
XML Describe Type=-10

[MySQL Wire Protocol]
Driver=/<Informatica installation directory>/ODBC7.1/lib/DWmysql27.so
Description=DataDirect 7.1 MySQL Wire Protocol
AlternateServers=
ApplicationUsingThreads=1
ConnectionReset=0
ConnectionRetryCount=0
ConnectionRetryDelay=3
Database=<database_name>

```

```

DefaultLongDataBuffLen=1024
EnableDescribeParam=0
EncryptionMethod=0
FailoverGranularity=0
FailoverMode=0
FailoverPreconnect=0
HostName=<MySQL_host>
HostNameInCertificate=
InteractiveClient=0
LicenseNotice=You must purchase commercially licensed MySQL database software or
a MySQL Enterprise subscription in order to use the DataDirect Connect for ODBC
for MySQL Enterprise driver with MySQL software.
KeyStore=
KeyStorePassword=
LoadBalanceTimeout=0
LoadBalancing=0
LogonID=
LoginTimeout=15
MaxPoolSize=100
MinPoolSize=0
Password=
Pooling=0
PortNumber=<MySQL_server_port>
QueryTimeout=0
ReportCodepageConversionErrors=0
TreatBinaryAsChar=0
TrustStore=
TrustStorePassword=
ValidateServerCertificate=1

[PostgreSQL Wire Protocol]
Driver=/<Informatica installation directory>/ODBC7.1/lib/DWpsql27.so
Description=DataDirect 7.1 PostgreSQL Wire Protocol
AlternateServers=
ApplicationUsingThreads=1
ConnectionReset=0
ConnectionRetryCount=0
ConnectionRetryDelay=3
Database=<database_name>
DefaultLongDataBuffLen=2048
EnableDescribeParam=1
EncryptionMethod=0
ExtendedColumnMetadata=0
FailoverGranularity=0
FailoverMode=0
FailoverPreconnect=0
FetchTSWTZasTimestamp=0
FetchTWFSasTime=0
HostName=<PostgreSQL_host>
HostNameInCertificate=
InitializationString=
KeyPassword=
KeyStore=
KeyStorePassword=
LoadBalanceTimeout=0
LoadBalancing=0
LoginTimeout=15
LogonID=
MaxPoolSize=100
MinPoolSize=0
Password=
Pooling=0
PortNumber=<PostgreSQL_server_port>
QueryTimeout=0
ReportCodepageConversionErrors=0
TransactionErrorBehavior=1
TrustStore=
TrustStorePassword=
ValidateServerCertificate=1
XMLDescribeType=-10

```

```

[Greenplum Wire Protocol]
Driver=/<Informatica installation directory>/ODBC7.1/lib/DWgplm27.so
Description=DataDirect 7.1 Greenplum Wire Protocol
AlternateServers=
ApplicationUsingThreads=1
ConnectionReset=0
ConnectionRetryCount=0
ConnectionRetryDelay=3
Database=<database_name>
DefaultLongDataBuffLen=2048
EnableDescribeParam=0
EnableKeysetCursors=0
EncryptionMethod=0
ExtendedColumnMetadata=0
FailoverGranularity=0
FailoverMode=0
FailoverPreconnect=0
FetchTSWTZasTimestamp=0
FetchTWFSasTime=0
HostName=<Greenplum_host>
InitializationString=
KeyPassword=
KeysetCursorOptions=0
KeyStore=
KeyStorePassword=
LoadBalanceTimeout=0
LoadBalancing=0
LoginTimeout=15
LogonID=
MaxPoolSize=100
MinPoolSize=0
Password=
Pooling=0
PortNumber=<Greenplum_server_port>
QueryTimeout=0
ReportCodepageConversionErrors=0
TransactionErrorBehavior=1
XMLDescribeType=-10

```

Note: You might have to customize the DSN entries in the `ODBC.ini` file based on the third-party driver that you use. For more information about the DSN entries, see the corresponding third-party driver documentation.

APPENDIX D

Updating the DynamicSections Parameter of a DB2 Database

This appendix includes the following topics:

- [DynamicSections Parameter Overview, 435](#)
- [Updating the DynamicSections Parameter, 435](#)

DynamicSections Parameter Overview

IBM DB2 packages contain the SQL statements to be executed on the database server. The DynamicSections parameter of a DB2 database determines the maximum number of executable statements that the database driver can have in a package. You can raise the value of the DynamicSections parameter to allow a larger number of executable statements in a DB2 package. To modify the DynamicSections parameter, connect to the database using a system administrator user account with BINDADD authority.

Updating the DynamicSections Parameter

Use the DataDirect Connect for JDBC utility to raise the value of the DynamicSections parameter in the DB2 database.

To use the DataDirect Connect for JDBC utility to update the DynamicSections parameter, complete the following tasks:

- Download and install the DataDirect Connect for JDBC utility.
- Run the Test for JDBC tool.

Downloading and Installing the DataDirect Connect for JDBC Utility

Download the DataDirect Connect for JDBC utility from the DataDirect download web site to a machine that has access to the DB2 database server. Extract the contents of the utility file and run the installer.

1. Go to the DataDirect download site:
<http://www.datadirect.com/support/product-documentation/downloads>
2. Choose the Connect for JDBC driver for an IBM DB2 data source.

3. Register to download the DataDirect Connect for JDBC Utility.
4. Download the utility to a machine that has access to the DB2 database server.
5. Extract the contents of the utility file to a temporary directory.
6. In the directory where you extracted the file, run the installer.

The installation program creates a folder named testforjdbc in the installation directory.

Running the Test for JDBC Tool

After you install the DataDirect Connect for JDBC Utility, run the Test for JDBC tool to connect to the DB2 database. You must use a system administrator user account with the BINDADD authority to connect to the database.

1. In the DB2 database, set up a system administrator user account with the BINDADD authority.
2. In the directory where you installed the DataDirect Connect for JDBC Utility, run the Test for JDBC tool.
On Windows, run testforjdbc.bat. On UNIX, run testforjdbc.sh.

3. On the Test for JDBC Tool window, click Press Here to Continue.

4. Click Connection > Connect to DB.

5. In the Database field, enter the following text:

```
jdbc:datadirect:db2://
HostName:PortNumber;databaseName=DatabaseName;CreateDefaultPackage=TRUE;ReplacePackage=TRUE;DynamicSections=3000
```

HostName is the name of the machine hosting the DB2 database server.

PortNumber is the port number of the database.

DatabaseName is the name of the DB2 database.

6. In the User Name and Password fields, enter the system administrator user name and password you use to connect to the DB2 database.
7. Click Connect, and then close the window.

INDEX

A

Abort

- option to disable PowerCenter Integration Service [223](#)
- option to disable PowerCenter Integration Service process [223](#)
- option to disable the Web Services Hub [375](#)

adaptive dispatch mode

- description [252](#)
- overview [262](#)

Additional JDBC Parameters

- description [184](#)

address validation properties

- configuring [43](#)

Administrator tool

- SAP BW Service, configuring [331](#)

advanced profiling properties

- configuring [64](#)

advanced properties

- Metadata Manager Service [188](#)
- PowerCenter Integration Service [231](#)
- PowerCenter Repository Service [292](#)
- Web Services Hub [376](#), [378](#)

Agent Cache Capacity (property)

- description [292](#)

agent port

- description [183](#)

AggregateTreatNullsAsZero

- option [233](#)
- option override [233](#)

AggregateTreatRowsAsInsert

- option [233](#)
- option override [233](#)

Aggregator transformation

- caches [271](#), [276](#)
- treating nulls as zero [233](#)
- treating rows as insert [233](#)

Allow Writes With Agent Caching (property)

- description [292](#)

Analyst Service

- Analyst Service security process properties [30](#)
- creating [33](#)
- custom service process properties [31](#)
- environment variables [32](#)
- Human task properties [28](#)
- Maximum Heap Size [31](#)
- node process properties [30](#)
- process properties [30](#)
- properties [26](#), [29](#)
- run-time properties [28](#)

application

- backing up [160](#)
- changing the name [159](#)
- deploying [156](#)
- enabling [159](#)
- properties [157](#)
- refreshing [161](#)

application service upgrade

- privileges [382](#)

application services

- system [346](#)

architecture

- Data Integration Service [80](#)

ASCII mode

- ASCII data movement mode, setting [229](#)
- Data Integration Service [83](#)
- overview [272](#)

associated PowerCenter Repository Service

- PowerCenter Integration Service [221](#)

associated repository

- Web Services Hub, adding to [380](#)
- Web Services Hub, editing for [381](#)

associated Repository Service

- Web Services Hub [374](#), [380](#), [381](#)

audit trails

- creating [314](#)

Authenticate MS-SQL User (property)

- description [292](#)

B

backing up

- list of backup files [311](#)
- performance [314](#)
- repositories [310](#)

backup directory

- Model Repository Service [207](#)

backup node

- license requirement [228](#)
- node assignment, configuring [228](#)
- PowerCenter Integration Service [221](#)

baseline system

- CPU profile [255](#)

basic authentication

- REST web services [124](#)

basic dispatch mode

- overview [262](#)

blocking

- description [268](#)

blocking source data

- PowerCenter Integration Service handling [268](#)

buffer memory

- buffer blocks [271](#)
- DTM process [271](#)

C

Cache Connection

- property [59](#)

cache files

- directory [242](#)

- cache files (*continued*)
 - overview [276](#)
 - permissions [273](#)
- Cache Removal Time
 - property [59](#)
- caches
 - default directory [276](#)
 - memory [271](#)
 - memory usage [271](#)
 - multiple directories [110](#)
 - overview [273](#)
 - transformation [276](#)
- certificate
 - keystore file [374](#), [377](#)
- character data sets
 - handling options for Microsoft SQL Server and PeopleSoft on Oracle [233](#)
- character encoding
 - Web Services Hub [377](#)
- classpath
 - Java SDK [242](#)
- ClientStore
 - option [231](#)
- Code Page (property)
 - PowerCenter Integration Service process [242](#)
 - PowerCenter Repository Service [286](#)
- code pages
 - data movement modes [272](#)
 - for PowerCenter Integration Service process [240](#)
 - global repository [304](#)
 - PowerCenter repository [286](#)
 - repository [303](#)
 - repository, Web Services Hub [374](#)
 - validation for sources and targets [235](#)
- command line programs
 - team-based development, administering [217](#)
- compatibility properties
 - PowerCenter Integration Service [233](#)
- Complete
 - option to disable PowerCenter Integration Service [223](#)
 - option to disable PowerCenter Integration Service process [223](#)
- compute node
 - overriding attributes [149](#)
- compute role
 - Data Integration Service node [84](#)
- Compute view
 - Data Integration Service [70](#)
 - environment variables [72](#)
 - execution options [71](#)
- concurrent jobs
 - Data Integration Service grid [151](#)
- configuration properties
 - Listener Service [320](#)
 - Logger Service [325](#)
 - PowerCenter Integration Service [235](#)
- configure and synchronize with version control system
 - how to [213](#)
- connect string
 - examples [179](#), [288](#)
 - PowerCenter repository database [290](#)
 - syntax [179](#), [288](#)
- connecting
 - Integration Service to IBM DB2 (Windows) [405](#), [413](#)
 - Integration Service to Informix (UNIX) [415](#)
 - Integration Service to Informix (Windows) [405](#)
 - Integration Service to Microsoft Access [406](#)
 - Integration Service to Microsoft SQL Server [406](#)
 - Integration Service to ODBC data sources (UNIX) [426](#)
- connecting (*continued*)
 - Integration Service to Oracle (UNIX) [420](#)
 - Integration Service to Oracle (Windows) [408](#)
 - Integration Service to Sybase ASE (UNIX) [422](#)
 - Integration Service to Sybase ASE (Windows) [410](#)
 - Microsoft Excel to Integration Service [406](#)
 - SQL data service [125](#)
 - UNIX databases [412](#)
 - Windows databases [404](#)
 - Windows using JDBC [404](#)
- connection performance
 - optimizing [102](#)
- connection pooling
 - description [100](#)
 - example [102](#)
 - management [101](#)
 - PowerExchange [103](#)
 - properties [101](#)
- connection resources
 - assigning [250](#)
- connections
 - adding pass-through security [126](#)
 - pass-through security [125](#)
- connectivity
 - connect string examples [179](#), [288](#)
 - overview [258](#)
- Content Management Service
 - architecture [35](#)
 - classifier model file path [47](#)
 - creating [47](#)
 - Data Integration Service grid [150](#)
 - file transfer option [41](#)
 - identity data properties [46](#)
 - log events [41](#)
 - Multi-Service Options [40](#)
 - operating system profiles [35](#)
 - orphaned reference data [37](#)
 - overview [34](#)
 - probabilistic model file path [47](#)
 - purge orphaned reference data [38](#)
 - reference data storage location [37](#), [40](#)
 - rule specifications [34](#), [35](#)
 - staging directory for reference data [41](#)
- control file
 - overview [275](#)
 - permissions [273](#)
- control files
 - Data Integration Service [97](#)
- CPU profile
 - computing [255](#)
 - description [255](#)
- CPU usage
 - Integration Service [271](#)
- CreateIndicatorFiles
 - option [235](#)
- custom properties
 - configuring for Data Integration Service [67](#), [70](#)
 - configuring for Metadata Manager [189](#)
 - configuring for Web Services Hub [380](#)
 - PowerCenter Integration Service process [244](#)
 - PowerCenter Repository Service [294](#)
 - PowerCenter Repository Service process [294](#)
 - Web Services Hub [376](#)
- custom resources
 - defining [250](#)
 - naming conventions [250](#)
- Custom transformation
 - directory for Java components [242](#)

D

- Data Analyzer repository
 - Oracle database requirements [395](#)
- data cache
 - memory usage [271](#)
- data handling
 - setting up prior version compatibility [233](#)
- Data Integration Service
 - architecture [80](#)
 - ASCII mode [83](#)
 - assign to grid [51](#)
 - assign to node [51](#)
 - compute component [80, 84](#)
 - compute properties [70](#)
 - configuring Data Integration Service security [68](#)
 - connectivity [79](#)
 - control file directories [97](#)
 - creating [51](#)
 - custom properties [67, 70](#)
 - data movement mode [83](#)
 - data object cache database [112](#)
 - disabling [93](#)
 - DTM instance [84](#)
 - DTM instances [99](#)
 - DTM process pool [99](#)
 - DTM processes [99](#)
 - enabling [93](#)
 - failover [76](#)
 - file directories [71, 95](#)
 - file permissions [98](#)
 - grid [128](#)
 - grid and node assignment properties [54](#)
 - high availability [75](#)
 - HTTP Configuration Properties [62](#)
 - HTTP proxy server properties [61](#)
 - LDTM [83](#)
 - log directory [97](#)
 - logs [90](#)
 - Maximum Heap Size [69](#)
 - maximum parallelism [106, 108](#)
 - operating system profile components [73](#)
 - operating system profiles [72](#)
 - optimization [102](#)
 - output files [85, 95](#)
 - output files on grid [96](#)
 - prerequisites [50](#)
 - processes [98](#)
 - properties [54](#)
 - recycling [93](#)
 - required databases [50](#)
 - restart [76](#)
 - result set cache properties [63, 69](#)
 - service components [80, 81](#)
 - source files on grid [96](#)
 - system parameters [95](#)
 - threads [107](#)
 - Unicode mode [83](#)
 - Workflow Orchestration Service properties [66](#)
- Data Integration Service grid
 - compute nodes [149](#)
 - concurrent jobs [151](#)
 - Content Management Service [150](#)
 - deleting [153](#)
 - editing [152](#)
 - local mode [135](#)
 - logs for remote mode [148](#)
 - mappings in local mode [135, 138](#)

- Data Integration Service grid (*continued*)
 - mappings in remote mode [141, 145](#)
 - prerequisites [130](#)
 - profiles in local mode [135, 138](#)
 - profiles in remote mode [141, 145](#)
 - recycling [144](#)
 - remote mode [141](#)
 - SQL data services [130, 132](#)
 - troubleshooting [153](#)
 - web services [130, 132](#)
 - workflows in local mode [135, 138](#)
 - workflows in remote mode [141, 145](#)
- Data Integration Service process
 - disabling [95](#)
 - enabling [95](#)
 - HTTP configuration properties [68](#)
 - properties [67](#)
- Data Integration Service process nodes
 - license requirement [54](#)
- data lineage
 - PowerCenter Repository Service, configuring [293](#)
- data lineage graph database
 - location [183](#)
 - Metadata Manager Lineage Graph Location property description [183](#)
- data movement mode
 - Data Integration Service [83](#)
 - for PowerCenter Integration Service [221](#)
 - option [229](#)
 - setting [229](#)
- data movement modes
 - overview [272](#)
- data object cache
 - configuring [111](#)
 - Data Object Cache Manager [83](#)
 - database requirements [387](#)
 - database tables [112](#)
 - description [111](#)
 - enabling [112](#)
 - IBM DB2 database requirements [387](#)
 - index cache [111](#)
 - Microsoft SQL Server database requirements [387](#)
 - Oracle database requirements [388](#)
 - properties [59](#)
 - user-managed tables [111, 114, 115](#)
- data object cache database
 - configuring for the Data Integration Service [112](#)
- Data Object Cache Manager
 - cache tables [112](#)
 - description [83](#)
- data object caching
 - with pass-through security [126](#)
- data service security
 - configuring Data Integration Service [68](#)
- Data Transformation Manager
 - optimizing job stability [98](#)
 - optimizing performance [102](#)
- database
 - repositories, creating for [286](#)
- database array operation size
 - description [290](#)
- database client
 - environment variables [244, 294](#)
- database clients
 - configuring [402](#)
 - environment variables [402](#)
 - IBM DB2 client application enabler [402](#)
 - Microsoft SQL Server native clients [402](#)

- database clients (*continued*)
 - Oracle clients [402](#)
 - Sybase open clients [402](#)
- database connection timeout
 - description [290](#)
- database connections
 - PowerCenter Integration Service resilience [279](#)
- Database Hostname
 - description [184](#)
- Database Name
 - description [184](#)
- Database Pool Expiration Threshold (property)
 - description [292](#)
- Database Pool Expiration Timeout (property)
 - description [292](#)
- Database Pool Size (property)
 - description [290](#)
- Database Port
 - description [184](#)
- database preparation
 - repositories [386](#)
- database requirements
 - data object cache [387](#)
 - exception management audit database [388](#)
 - Metadata Manager repository [389](#)
 - Model repository [393](#)
 - PowerCenter repository [394](#)
 - profiling warehouse [396](#)
 - reference data warehouse [398](#)
 - workflow database [399](#)
- database resilience
 - repository [295](#)
- database statistics
 - IBM DB2 [123](#)
 - Microsoft SQL Server [124](#)
 - Oracle [124](#)
- database user accounts
 - guidelines for setup [387](#)
- databases
 - connecting to (UNIX) [412](#)
 - connecting to (Windows) [404](#)
 - connecting to IBM DB2 [405](#), [413](#)
 - connecting to Informix [405](#), [415](#)
 - connecting to Microsoft Access [406](#)
 - connecting to Microsoft SQL Server [406](#)
 - connecting to Netezza (UNIX) [418](#)
 - connecting to Netezza (Windows) [408](#)
 - connecting to Oracle [408](#), [420](#)
 - connecting to Sybase ASE [410](#), [422](#)
 - connecting to Teradata (UNIX) [424](#)
 - connecting to Teradata (Windows) [411](#)
 - Data Analyzer repository [387](#)
 - Metadata Manager repository [387](#)
 - PowerCenter repository [387](#)
 - testing connections [402](#)
- DateDisplayFormat
 - option [235](#)
- DateHandling40Compatibility
 - option [233](#)
- dates
 - default format for logs [235](#)
- dbs2 connect
 - testing database connections [402](#)
- deadlock retries
 - setting number [233](#)
- DeadlockSleep
 - option [233](#)

- Debug
 - error severity level [231](#), [378](#)
- Debugger
 - running [231](#)
- dependency graph
 - rebuilding [384](#)
- deployment
 - applications [156](#)
- directories
 - cache files [242](#)
 - external procedure files [242](#)
 - for Java components [242](#)
 - lookup files [242](#)
 - recovery files [242](#)
 - reject files [242](#)
 - root directory [242](#)
 - session log files [242](#)
 - source files [242](#)
 - target files [242](#)
 - temporary files [242](#)
 - workflow log files [242](#)
- disabling
 - Metadata Manager Service [181](#)
 - PowerCenter Integration Service [223](#)
 - PowerCenter Integration Service process [223](#)
 - Web Services Hub [375](#)
- dispatch mode
 - adaptive [252](#)
 - configuring [252](#)
 - Load Balancer [262](#)
 - metric-based [252](#)
 - round-robin [252](#)
- dispatch priority
 - configuring [254](#)
- dispatch queue
 - overview [260](#)
 - service levels, creating [254](#)
- dispatch wait time
 - configuring [254](#)
- domain
 - associated repository for Web Services Hub [374](#)
 - metadata, sharing [303](#)
- domain configuration repository
 - IBM DB2 database requirements [195](#), [393](#)
 - Microsoft SQL Server database requirements [196](#)
- DTM (Data Transformation Manager)
 - buffer memory [271](#)
 - distribution on PowerCenter grids [270](#)
 - instance [84](#)
 - master DTM [270](#)
 - output files [85](#)
 - preparer DTM [270](#)
 - process [87](#), [263](#)
 - processing threads [85](#)
 - resource allocation policy [85](#)
 - worker DTM [270](#)
- DTM instances
 - Data Integration Service [84](#)
 - description [99](#)
- DTM process
 - environment variables [72](#)
- DTM processes
 - description [99](#)
 - pool [99](#)
 - pool management [99](#)
- DTM timeout
 - Web Services Hub [378](#)

E

- Email Service
 - properties [348](#)
- Enable Nested LDO Cache
 - property [59](#)
- enabling
 - Metadata Manager Service [181](#)
 - PowerCenter Integration Service [223](#)
 - PowerCenter Integration Service process [223](#)
 - Web Services Hub [375](#)
- encoding
 - Web Services Hub [377](#)
- environment variables
 - compute node [72](#)
 - database client [244](#), [294](#)
 - database clients [402](#)
 - DTM process [72](#)
 - Listener Service process [320](#)
 - Logger Service process [328](#)
 - MapR [246](#)
 - PowerCenter Integration Service process [244](#)
 - PowerCenter Repository Service process [294](#)
 - UNIX database clients [402](#)
- Error
 - severity level [231](#), [378](#)
- error logs
 - messages [274](#)
- Error Severity Level (property)
 - Metadata Manager Service [188](#)
 - PowerCenter Integration Service [231](#)
- exception management audit database
 - IBM DB2 database requirements [388](#)
 - Microsoft SQL Server database requirements [389](#)
 - Oracle database requirements [389](#)
- execution Data Transformation Manager
 - Data Integration Service [84](#)
- execution options
 - configuring [56](#)
 - override for compute node [71](#)
- ExportSessionLogLibName
 - option [235](#)
- external procedure files
 - directory [242](#)

F

- failover
 - PowerCenter Integration Service [280](#)
 - PowerCenter Repository Service [296](#)
 - PowerExchange Listener Service [322](#)
 - PowerExchange Logger Service [329](#)
 - safe mode [227](#)
- file permissions
 - Data Integration Service [98](#)
- file/directory resources
 - defining [250](#)
 - naming conventions [250](#)
- filtering data
 - SAP BW, parameter file location [336](#)
- flat files
 - output files [276](#)
- folders
 - operating system profile, assigning [310](#)
- FTP connections
 - PowerCenter Integration Service resilience [279](#)

G

- general properties
 - Listener Service [319](#)
 - Logger Service [325](#)
 - Metadata Manager Service [182](#)
 - PowerCenter Integration Service [229](#)
 - PowerCenter Integration Service process [242](#)
 - PowerCenter Repository Service [289](#)
 - SAP BW Service [334](#)
 - Web Services Hub [376](#), [377](#)
- global repositories
 - code page [303](#), [304](#)
 - creating [304](#)
 - creating from local repositories [304](#)
 - moving to another Informatica domain [307](#)
- grid
 - Data Integration Service file directories [96](#)
 - troubleshooting for PowerCenter Integration Service [251](#)
- grid assignment properties
 - Data Integration Service [54](#)
 - PowerCenter Integration Service [228](#)
- grids
 - assigning to a PowerCenter Integration Service [248](#)
 - configuring for PowerCenter Integration Service [246](#)
 - creating [247](#)
 - Data Integration Service [128](#)
 - description for PowerCenter Integration Service [269](#)
 - DTM processes for PowerCenter [270](#)
 - for PowerCenter Integration Service [221](#)
 - license requirement [54](#)
 - license requirement for PowerCenter Integration Service [228](#)
 - operating system profile [248](#)
 - PowerCenter Integration Service processes, distributing [269](#)
 - troubleshooting for Data Integration Service [153](#)

H

- heartbeat interval
 - description [292](#)
- high availability
 - licensed option [228](#)
 - Listener Service [322](#)
 - Logger Service [329](#)
 - PowerCenter Integration Service [278](#)
 - PowerCenter Repository Service [295](#)
 - PowerCenter Repository Service failover [296](#)
 - PowerCenter Repository Service recovery [296](#)
 - PowerCenter Repository Service resilience [295](#)
 - PowerCenter Repository Service restart [296](#)
- high availability option
 - service processes, configuring [299](#)
- high availability persistence tables
 - PowerCenter Integration Service [283](#)
- host names
 - Web Services Hub [374](#), [377](#)
- host port number
 - Web Services Hub [374](#), [377](#)
- how to
 - configure and synchronize a Model repository with a version control system [213](#)
- HTTP
 - Data Integration Service [79](#)
- HTTP configuration properties
 - Data Integration Service process [68](#)
- HTTP Configuration Properties
 - Data Integration Service [62](#)

- HTTP proxy
 - domain setting [237](#)
 - password setting [237](#)
 - port setting [237](#)
 - server setting [237](#)
 - user setting [237](#)
- HTTP proxy properties
 - PowerCenter Integration Service [237](#)
- HTTP proxy server
 - usage [237](#)
- HTTP proxy server properties
 - Data Integration Service [61](#)
- HttpProxyDomain
 - option [237](#)
- HttpProxyPassword
 - option [237](#)
- HttpProxyPort
 - option [237](#)
- HttpProxyServer
 - option [237](#)
- HttpProxyUser
 - option [237](#)
- HTTPS
 - Data Integration Service [79](#)
 - keystore file [374, 377](#)
 - keystore password [374, 377](#)
- Hub Logical Address (property)
 - Web Services Hub [378](#)

I

- IBM DB2
 - connect string example [179, 288](#)
 - connecting to Integration Service (Windows) [405, 413](#)
 - repository database schema, optimizing [290](#)
 - setting DB2CODEPAGE [405](#)
 - setting DB2INSTANCE [405](#)
 - single-node tablespaces [395](#)
- IBM DB2 database requirements
 - data object cache [387](#)
 - domain repository [195, 393](#)
 - exception management audit database [388](#)
 - Metadata Manager repository [390](#)
 - Model repository database [195, 393](#)
 - PowerCenter repository [395](#)
 - profiling warehouse [396](#)
 - reference data warehouse [398](#)
 - workflow repository [399](#)
- IgnoreResourceRequirements
 - option [231](#)
- incremental aggregation
 - files [276](#)
- index caches
 - memory usage [271](#)
- indicator files
 - description [276](#)
 - session output [276](#)
- infacmd mrs
 - listing checked-out object [217](#)
 - listing locked object [217](#)
 - reassigning locked or checked-out object [217](#)
 - undoing checked-out object [217](#)
 - unlocking locked object [217](#)
- infacmd ps
 - purging profile and scorecard results [120](#)
- Informatica Administrator
 - repositories, backing up [310](#)

- Informatica Administrator (*continued*)
 - repositories, restoring [311](#)
 - repository notifications, sending [309](#)
 - tasks for Web Services Hub [373](#)
- Information error severity level
 - description [231, 378](#)
- Informix
 - connecting to Integration Service (UNIX) [415](#)
 - connecting to Integration Service (Windows) [405](#)
- internal host name
 - Web Services Hub [374, 377](#)
- internal port number
 - Web Services Hub [374, 377](#)
- isAuthenticationRequired
 - REST web services [124](#)
- isql
 - testing database connections [402](#)

J

- Java
 - configuring for JMS [242](#)
 - configuring for PowerExchange for Web Services [242](#)
 - configuring for webMethods [242](#)
- Java components
 - directories, managing [242](#)
- Java SDK
 - class path [242](#)
 - maximum memory [242](#)
 - minimum memory [242](#)
- Java SDK Class Path
 - option [242](#)
- Java SDK Maximum Memory
 - option [242](#)
- Java SDK Minimum Memory
 - option [242](#)
- Java transformation
 - directory for Java components [242](#)
- JCEProvider
 - option [231](#)
- JDBC
 - connecting to (Windows) [404](#)
 - Data Integration Service [79](#)
- jobs
 - launch as separate processes [98](#)
- Joiner transformation
 - caches [271, 276](#)
 - setting up for prior version compatibility [233](#)
- JoinerSourceOrder6xCompatibility
 - option [233](#)
- JVM Command Line Options
 - advanced Web Services Hub property [378](#)

K

- keystore file
 - Metadata Manager [187](#)
 - Web Services Hub [374, 377](#)
- keystore password
 - Web Services Hub [374, 377](#)

L

- LDTM
 - Data Integration Service [83](#)

- license
 - for PowerCenter Integration Service [221](#)
 - Web Services Hub [374](#), [377](#)
- licensed options
 - high availability [228](#)
 - server grid [228](#)
- Limit on Resilience Timeouts (property)
 - description [292](#)
- linked domain
 - multiple domains [305](#)
- Linux
 - database client environment variables [402](#)
- listCheckedoutObjects (infacmd mrs) [217](#)
- Listener Service process
 - environment variables [320](#)
- listing
 - checked-out object [217](#)
 - locked object [217](#)
- listLockedObjects (infacmd mrs) [217](#)
- Load Balancer
 - configuring to check resources [261](#)
 - defining resource provision thresholds [255](#)
 - dispatch mode [262](#)
 - dispatching tasks in a grid [261](#)
 - dispatching tasks on a single node [261](#)
 - resource provision thresholds [262](#)
 - resources [248](#), [261](#)
- Load Balancer for PowerCenter Integration Service
 - assigning priorities to tasks [254](#), [263](#)
 - configuring to check resources [231](#), [255](#)
 - CPU profile, computing [255](#)
 - dispatch mode, configuring [252](#)
 - dispatch queue [260](#)
 - overview [260](#)
 - service levels [263](#)
 - service levels, creating [254](#)
 - settings, configuring [251](#)
- load balancing
 - SAP BW Service [337](#)
 - support for SAP BW system [337](#)
- LoadManagerAllowDebugging
 - option [231](#)
- local mode
 - Data Integration Service grid [135](#)
- local repositories
 - code page [303](#)
 - moving to another Informatica domain [307](#)
 - promoting [304](#)
 - registering [305](#)
- locks
 - managing [307](#)
 - viewing [307](#)
- log files
 - Data Integration Service [90](#), [97](#)
 - Data Integration Service permissions [98](#)
- Log Level (property)
 - Web Services Hub [378](#)
- Logger Service process
 - environment variables [328](#)
 - properties [328](#)
- logical data objects
 - caching in database [111](#)
- logical Data Transformation Manager
 - Data Integration Service [83](#)
- logs
 - error severity level [231](#)
 - in UTF-8 [231](#)
 - session [274](#)

- logs (*continued*)
 - workflow [274](#)
- LogInUTF8
 - option [231](#)
- lookup caches
 - persistent [277](#)
- lookup files
 - directory [242](#)
- Lookup transformation
 - caches [271](#), [276](#)

M

- Manage List
 - linked domains, adding [305](#)
- mapping pipelines
 - description [107](#)
- mapping properties
 - configuring [162](#)
- mappings
 - Data Integration Service grid [135](#), [141](#)
 - grids in local mode [138](#)
 - grids in remote mode [145](#)
 - maximum parallelism [106](#), [108](#)
 - partition points [107](#)
 - partitioned [108](#)
 - pipelines [107](#)
 - processing threads [107](#)
- master thread
 - description [264](#)
- Max Concurrent Resource Load
 - description, Metadata Manager Service [188](#)
- Max Heap Size
 - description, Metadata Manager Service [188](#)
- Max Lookup SP DB Connections
 - option [233](#)
- Max MSSQL Connections
 - option [233](#)
- Max Sybase Connections
 - option [233](#)
- MaxConcurrentRequests
 - advanced Web Services Hub property [378](#)
 - description, Metadata Manager Service [187](#)
- Maximum Active Connections
 - description, Metadata Manager Service [187](#)
 - SQL data service property [164](#)
- maximum active users
 - description [292](#)
- Maximum Catalog Child Objects
 - description [188](#)
- Maximum Concurrent Connections
 - configuring [70](#)
- Maximum Concurrent Refresh Requests
 - property [59](#)
- Maximum CPU Run Queue Length
 - node property [255](#)
- maximum dispatch wait time
 - configuring [254](#)
- Maximum Heap Size
 - advanced Web Services Hub property [378](#)
 - configuring Analyst Service [31](#)
 - configuring Data Integration Service [69](#)
 - configuring Model Repository Service [201](#)
 - configuring Search Service [343](#)
- maximum locks
 - description [292](#)

- Maximum Memory Percent
 - node property [255](#)
- maximum parallelism
 - description [106](#), [108](#)
 - guidelines [109](#)
- Maximum Processes
 - node property [255](#)
- Maximum Wait Time
 - description, Metadata Manager Service [187](#)
- MaxISConnections
 - Web Services Hub [378](#)
- MaxQueueLength
 - advanced Web Services Hub property [378](#)
 - description, Metadata Manager Service [187](#)
- MaxStatsHistory
 - advanced Web Services Hub property [378](#)
- memory
 - DTM buffer [271](#)
 - maximum for Java SDK [242](#)
 - Metadata Manager [188](#)
 - minimum for Java SDK [242](#)
- metadata
 - sharing between domains [303](#)
- Metadata Manager
 - components [173](#)
 - configuring PowerCenter Integration Service [189](#)
 - repository [174](#)
 - starting [181](#)
 - user for PowerCenter Integration Service [190](#)
- Metadata Manager File Location (property)
 - description [183](#)
- Metadata Manager lineage graph location
 - configuring [184](#)
- Metadata Manager repository
 - content, creating [180](#)
 - content, deleting [181](#)
 - creating [174](#)
 - database requirements [389](#)
 - heap sizes [390](#)
 - IBM DB2 database requirements [390](#)
 - Microsoft SQL Server database requirements [391](#)
 - optimizing IBM DB2 databases [390](#)
 - Oracle database requirements [392](#)
 - system temporary tablespaces [390](#)
- Metadata Manager Service
 - advanced properties [188](#)
 - components [173](#)
 - creating [175](#)
 - custom properties [189](#)
 - description [173](#)
 - disabling [181](#)
 - general properties [182](#)
 - properties [182](#), [183](#)
 - recycling [181](#)
 - steps to create [174](#)
- Metadata Manager Service properties
 - PowerCenter Repository Service [293](#)
- metric-based dispatch mode
 - description [252](#)
- Microsoft Access
 - connecting to Integration Service [406](#)
- Microsoft Excel
 - connecting to Integration Service [406](#)
 - using PmNullPasswd [406](#)
 - using PmNullUser [406](#)
- Microsoft SQL Server
 - connect string syntax [179](#), [288](#)
 - connecting from UNIX [416](#)

- Microsoft SQL Server (*continued*)
 - connecting to Integration Service [406](#)
 - repository database schema, optimizing [290](#)
 - setting Char handling options [233](#)
- Microsoft SQL Server database requirements
 - data object cache [387](#)
 - domain configuration repository [196](#)
 - exception management audit database [389](#)
 - Metadata Manager repository [391](#)
 - Model repository [394](#)
 - PowerCenter repository [395](#)
 - profiling warehouse [397](#)
 - reference data warehouse [398](#)
 - workflow repository [400](#)
- Minimum Severity for Log Entries (property)
 - PowerCenter Repository Service [292](#)
- model repository
 - backing up [208](#)
 - creating [207](#)
 - creating content [207](#)
 - deleting [207](#)
 - deleting content [207](#)
 - restoring content [208](#)
- Model repository
 - database requirements [393](#)
 - IBM DB2 database requirements [195](#), [393](#)
 - listing checked-out object in [217](#)
 - listing locked object in [217](#)
 - Microsoft SQL Server database requirements [394](#)
 - non-versioned [216](#)
 - Oracle database requirements [196](#), [394](#)
 - reassigning locked or checked-out object in [217](#)
 - reverting checked-out object in [217](#)
 - team-based development [216](#), [217](#)
 - undoing checked-out object in [217](#)
 - unlocking locked object in [217](#)
 - versioned [216](#)
- Model Repository Service
 - cache management [212](#)
 - backup directory [207](#)
 - Creating [218](#)
 - custom search analyzer [210](#)
 - disabling [197](#)
 - enabling [197](#)
 - failover [206](#)
 - high availability [206](#)
 - logs [211](#)
 - Maximum Heap Size [201](#)
 - overview [192](#)
 - properties [198](#)
 - recycling [197](#)
 - restart [206](#)
 - search analyzer [209](#)
 - search index [209](#)
 - upgrade error [384](#)
 - versioning [202](#)
- Model Repository Service process
 - disabling [197](#)
 - enabling [197](#)
- modules
 - disabling [61](#)
- MSExchangeProfile
 - option [235](#)

N

- native drivers
 - Data Integration Service [79](#)
- Netezza
 - connecting from Informatica clients(Windows) [408](#)
 - connecting from Integration Service (Windows) [408](#)
 - connecting to Informatica clients (UNIX) [418](#)
 - connecting to Integration Service (UNIX) [418](#)
- node assignment
 - Data Integration Service [54](#)
 - PowerCenter Integration Service [228](#)
 - Resource Manager Service [351](#)
 - Web Services Hub [376](#), [377](#)
- node properties
 - maximum CPU run queue length [255](#)
 - maximum memory percent [255](#)
 - maximum processes [255](#)
- nodes
 - node assignment, configuring [228](#)
 - Web Services Hub [374](#)
- normal mode
 - PowerCenter Integration Service [225](#)
- notifications
 - sending [309](#)
- null values
 - PowerCenter Integration Service, configuring [233](#)
- NumOfDeadlockRetries
 - option [233](#)

O

- object dependency graph
 - rebuilding [384](#)
- objects
 - filtering [216](#)
- ODBC
 - Data Integration Service [79](#)
- ODBC Connection Mode
 - description [188](#)
- ODBC data sources
 - connecting to (UNIX) [426](#)
 - connecting to (Windows) [404](#)
- odbc.ini file
 - sample [428](#)
- operating mode
 - effect on resilience [300](#)
 - normal mode for PowerCenter Integration Service [224](#)
 - PowerCenter Integration Service [224](#)
 - PowerCenter Repository Service [300](#)
 - safe mode for PowerCenter Integration Service [224](#)
- operating system profile
 - components [73](#)
 - configuration [238](#)
 - configuration, Data Integration Service [73](#)
 - enabling Data Integration Service [75](#)
 - folders, assigning to [310](#)
 - pmimpprocess [238](#)
 - pmsuid, Data Integration Service [73](#)
 - PowerCenter Integration Service grids [248](#)
 - system permissions, Data Integration Service [74](#)
 - troubleshooting [239](#)
 - troubleshooting, Data Integration Service [75](#)
- operating system profiles
 - overview, Data Integration Service [72](#)
 - overview, PowerCenter Integration Service [238](#)

- optimization
 - Data Integration [102](#)
 - PowerCenter repository [395](#)
- Oracle
 - connect string syntax [179](#), [288](#)
 - connecting to Integration Service (UNIX) [420](#)
 - connecting to Integration Service (Windows) [408](#)
- Oracle database requirements
 - Data Analyzer repository [395](#)
 - data object cache [388](#)
 - exception management audit database [389](#)
 - Metadata Manager repository [392](#)
 - Model repository [196](#), [394](#)
 - profiling warehouse [397](#)
 - reference data warehouse [399](#)
 - workflow repository [400](#)
- Oracle Net Services
 - using to connect Integration Service to Oracle (UNIX) [420](#)
 - using to connect Integration Service to Oracle (Windows) [408](#)
- output files
 - Data Integration Service [85](#), [95](#)
 - Data Integration Service permissions [98](#)
 - overview [273](#), [276](#)
 - permissions [273](#)
 - target files [276](#)
- OutputMetaDataForFF
 - option [235](#)
- overview
 - Content Management Service [34](#)

P

- page size
 - minimum for optimizing repository database schema [290](#)
- partition points
 - description [107](#)
- partitioning
 - enabling [110](#)
 - mappings [108](#)
 - maximum parallelism [106](#), [108](#)
- pass-through pipeline
 - overview [264](#)
- pass-through security
 - adding to connections [126](#)
 - connecting to SQL data service [125](#)
 - enabling caching [126](#)
 - properties [60](#)
 - web service operation mappings [125](#)
- PeopleSoft on Oracle
 - setting Char handling options [233](#)
- performance
 - details [274](#)
 - PowerCenter Integration Service [292](#)
 - PowerCenter Repository Service [292](#)
 - repository copy, backup, and restore [314](#)
 - repository database schema, optimizing [290](#)
- performance detail files
 - permissions [273](#)
- permissions
 - output and log files [273](#)
 - recovery files [273](#)
- persistent lookup cache
 - session output [277](#)
- pipeline partitioning
 - multiple CPUs [267](#)
 - overview [267](#)
 - symmetric processing platform [271](#)

- pipeline stages
 - description [107](#)
- plug-ins
 - registering [313](#)
 - unregistering [313](#)
- \$PMBadFileDir
 - option [242](#)
- \$PMCacheDir
 - option [242](#)
- \$PMExtProcDir
 - option [242](#)
- \$PMFailureEmailUser
 - option [229](#)
- pmimpprocess
 - description [238](#)
- \$PMLookupFileDir
 - option [242](#)
- \$PMRootDir
 - description [241](#)
 - option [242](#)
 - required syntax [241](#)
 - shared location [241](#)
- PMServer3XCompatibility
 - option [233](#)
- \$PMSessionErrorThreshold
 - option [229](#)
- \$PMSessionLogCount
 - option [229](#)
- \$PMSessionLogDir
 - option [242](#)
- \$PMSourceFileDir
 - option [242](#)
- \$PMStorageDir
 - option [242](#)
- \$PMSuccessEmailUser
 - option [229](#)
- pmsuid
 - description [73](#)
- \$PMTargetFileDir
 - option [242](#)
- \$PMTempDir
 - option [242](#)
- \$PMWorkflowLogCount
 - option [229](#)
- \$PMWorkflowLogDir
 - option [242](#)
- pooling
 - connection [100](#)
 - DTM process [99](#)
- pools
 - connection [100](#)
 - DTM process [99](#)
- port number
 - Metadata Manager Agent [183](#)
 - Metadata Manager application [183](#)
- post-session email
 - Microsoft Exchange profile, configuring [235](#)
 - overview [275](#)
- PowerCenter Analyst Service process
 - MapR environment variables [32](#)
- PowerCenter Integration Service
 - advanced properties [231](#)
 - architecture [257](#)
 - assign to grid [221](#), [248](#)
 - assign to node [221](#)
 - associated repository [239](#)
 - blocking data [268](#)
 - compatibility and database properties [233](#)

- PowerCenter Integration Service (*continued*)
 - configuration properties [235](#)
 - configuring for Metadata Manager [189](#)
 - connectivity overview [258](#)
 - creating [221](#)
 - data movement mode [221](#), [229](#)
 - data movement modes [272](#)
 - data, processing [267](#)
 - date display format [235](#)
 - disable process with Abort option [223](#)
 - disable process with Stop option [223](#)
 - disable with Abort option [223](#)
 - disable with Complete option [223](#)
 - disable with Stop option [223](#)
 - disabling [223](#)
 - enabling [223](#)
 - export session log lib name, configuring [235](#)
 - external component resilience [279](#)
 - fail over in safe mode [225](#)
 - failover [280](#)
 - failover configuration [283](#)
 - failover, on grid [282](#)
 - for Metadata Manager [173](#)
 - for Test Data Manager [359](#)
 - general properties [229](#)
 - grid and node assignment properties [228](#)
 - high availability [278](#)
 - high availability persistence tables [283](#)
 - HTTP proxy properties [237](#)
 - logs in UTF-8 [231](#)
 - name [221](#)
 - normal operating mode [225](#)
 - operating mode [224](#)
 - operating system profiles [238](#)
 - output files [276](#)
 - overview [220](#)
 - performance [292](#)
 - performance details [274](#)
 - PowerCenter Integration Service client resilience [279](#)
 - PowerCenter Repository Service, associating [221](#)
 - process [258](#)
 - recovery [283](#)
 - recovery configuration [283](#)
 - resilience [278](#)
 - resilience period [231](#)
 - resilience timeout [231](#)
 - resource requirements [231](#)
 - restart [280](#)
 - safe mode, running in [226](#)
 - safe operating mode [225](#)
 - session recovery [282](#)
 - shared storage [241](#)
 - sources, reading [267](#)
 - state of operations [283](#)
 - system resources [271](#)
 - version [233](#)
 - workflow recovery [283](#)
- PowerCenter Integration Service process
 - \$PMBadFileDir [242](#)
 - \$PMCacheDir [242](#)
 - \$PMExtProcDir [242](#)
 - \$PMLookupFileDir [242](#)
 - \$PMRootDir [242](#)
 - \$PMSessionLogDir [242](#)
 - \$PMSourceFileDir [242](#)
 - \$PMStorageDir [242](#)
 - \$PMTargetFileDir [242](#)
 - \$PMTempDir [242](#)

PowerCenter Integration Service process (*continued*)

- \$PMWorkflowLogDir [242](#)
- code page [240](#)
- code pages, specifying [242](#)
- custom properties [244](#)
- disable with Complete option [223](#)
- disabling [223](#)
- distribution on a grid [269](#)
- enabling [223](#)
- environment variables [244](#)
- general properties [242](#)
- Java component directories [242](#)
- MapR environment variables [246](#)

PowerCenter Integration Service process nodes

- license requirement [228](#)

PowerCenter repository

- associated with Web Services Hub [380](#)
- code pages [286](#)
- content, creating for Metadata Manager [180](#)
- data lineage, configuring [293](#)
- database requirements [394](#)
- IBM DB2 database requirements [395](#)
- Microsoft SQL Server database requirements [395](#)
- optimizing IBM DB2 databases [395](#)
- Sybase ASE database requirements [396](#)

PowerCenter Repository Service

- advanced properties [292](#)
- associating with a Web Services Hub [374](#)
- Code Page (property) [286](#)
- configuring [289](#)
- creating [286](#)
- data lineage, configuring [293](#)
- enabling and disabling [298](#)
- failover [296](#)
- for Metadata Manager [173](#)
- for Test Data Manager [359](#)
- general properties [289](#)
- high availability [295](#)
- Metadata Manager Service properties [293](#)
- operating mode [300](#)
- overview [285](#)
- performance [292](#)
- PowerCenter Integration Service, associating [221](#)
- properties [289](#)
- recovery [296](#)
- repository agent caching [292](#)
- repository properties [290](#)
- resilience [295](#)
- resilience to database [295](#)
- restart [296](#)
- service process [299](#)
- state of operations [296](#)

PowerCenter Repository Service process

- configuring [294](#)
- environment variables [294](#)
- properties [294](#)

PowerCenter tasks

- dispatch priorities, assigning [263](#)
- dispatching [260](#)

PowerExchange

- connection pooling [103](#)

PowerExchange for JMS

- directory for Java components [242](#)

PowerExchange for Web Services

- directory for Java components [242](#)

PowerExchange for webMethods

- directory for Java components [242](#)

PowerExchange Listener Service

- creating [318](#)
- disabling [321](#)
- enabling [321](#)
- failover [322](#)
- properties [318](#)
- restart [322](#)
- restarting [322](#)

PowerExchange Logger Service

- creating [324](#)
- disabling [328](#)
- enabling [328](#)
- failover [329](#)
- properties [325](#)
- restart [329](#)
- restarting [329](#)

Preserve MX Data (property)

- description [292](#)

primary node

- for PowerCenter Integration Service [221](#)
- node assignment, configuring [228](#)

processing threads

- mappings [107](#)

profile warehouse management

- database management [120](#)
- tablespace recovery [122](#)

profiles

- Data Integration Service grid [135](#), [141](#)
- grids in local mode [138](#)
- grids in remote mode [145](#)
- maximum parallelism [106](#)
- purging results for [120](#)

profiling properties

- configuring [64](#)

profiling warehouse

- creating [119](#)
- creating content [119](#)
- database requirements [396](#)
- deleting [119](#)
- deleting content [119](#)
- IBM DB2 database requirements [396](#)
- Microsoft SQL Server database requirements [397](#)
- Oracle database requirements [397](#)

Profiling Warehouse Connection Name

- configuring [63](#)

profiling warehouse management

- database statistics [123](#)

properties

- Metadata Manager Service [183](#)

Purge (infacmd ps) [120](#)

R

Rank transformation

- caches [271](#), [276](#)

reassignCheckedOutObject (infacmd mrs) [217](#)

reassigning

- checked-out object [217](#)
- locked object [217](#)

recovery

- files, permissions [273](#)
- PowerCenter Integration Service [283](#)
- PowerCenter Repository Service [296](#)
- safe mode [227](#)

recovery files

- directory [242](#)

- reference data
 - purge orphaned data [38](#)
- reference data warehouse
 - database requirements [398](#)
 - IBM DB2 database requirements [398](#)
 - Microsoft SQL Server database requirements [398](#)
 - Oracle database requirements [399](#)
- registering
 - local repositories [305](#)
 - plug-ins [313](#)
- reject files
 - directory [242](#)
 - overview [275](#)
 - permissions [273](#)
- remote mode
 - Data Integration Service grid [141](#)
 - logs [148](#)
- repagent caching
 - description [292](#)
- Reporting Service
 - using with Metadata Manager [174](#)
- repositories
 - associated with PowerCenter Integration Service [239](#)
 - backing up [310](#)
 - code pages [303](#), [304](#)
 - configuring native connectivity [401](#)
 - content, creating [180](#), [301](#)
 - content, deleting [180](#), [302](#)
 - database preparation [386](#)
 - database schema, optimizing [290](#)
 - database, creating [286](#)
 - installing database clients [402](#)
 - Metadata Manager [173](#)
 - moving [307](#)
 - notifications [309](#)
 - performance [314](#)
 - persisting run-time statistics [231](#)
 - restoring [311](#)
 - security log file [314](#)
 - Test Data Manager [359](#)
 - version control [303](#)
- repository agent cache capacity
 - description [292](#)
- repository agent caching
 - PowerCenter Repository Service [292](#)
- Repository Agent Caching (property)
 - description [292](#)
- repository domains
 - description [303](#)
 - managing [303](#)
 - moving to another Informatica domain [307](#)
 - prerequisites [303](#)
 - registered repositories, viewing [306](#)
 - user accounts [304](#)
- repository locks
 - managing [307](#)
 - releasing [309](#)
 - viewing [307](#)
- repository notifications
 - sending [309](#)
- repository password
 - associated repository for Web Services Hub [380](#), [381](#)
 - option [239](#)
- repository properties
 - PowerCenter Repository Service [290](#)
- Repository Service process
 - description [299](#)
- repository user name
 - associated repository for Web Services Hub [374](#), [380](#), [381](#)
 - option [239](#)
- repository user password
 - associated repository for Web Services Hub [374](#)
- request timeout
 - SQL data services requests [164](#)
- Required Comments for Checkin(property)
 - description [292](#)
- resilience
 - in exclusive mode [300](#)
 - period for PowerCenter Integration Service [231](#)
 - PowerCenter Integration Service [278](#)
 - PowerCenter Repository Service [295](#)
 - repository database [295](#)
- Resilience Timeout (property)
 - description [292](#)
 - option [231](#)
- Resource Manager Service
 - architecture [350](#)
 - compute node attributes [149](#)
 - disabling [352](#)
 - enabling [352](#)
 - log level [351](#)
 - node assignment [351](#)
 - overview [350](#)
 - properties [351](#)
 - recycling [352](#)
- Resource Manager Service process
 - properties [352](#)
- resource provision thresholds
 - defining [255](#)
 - description [255](#)
 - overview [262](#)
- resources
 - configuring [248](#)
 - configuring Load Balancer to check [231](#), [255](#), [261](#)
 - connection, assigning [250](#)
 - defining custom [250](#)
 - defining file/directory [250](#)
 - defining for nodes [248](#)
 - Load Balancer [261](#)
 - naming conventions [250](#)
 - node [261](#)
 - predefined [248](#)
 - user-defined [248](#)
- restart
 - PowerCenter Integration Service [280](#)
 - PowerCenter Repository Service [296](#)
 - PowerExchange Listener Service [322](#)
 - PowerExchange Logger Service [329](#)
- restoring
 - PowerCenter repository for Metadata Manager [180](#)
 - repositories [311](#)
- result set cache
 - configuring [111](#)
 - Data Integration Service properties [63](#), [69](#)
 - purging [111](#)
 - SQL data service properties [164](#)
- Result Set Cache Manager
 - description [83](#)
- result set caching
 - Result Set Cache Manager [83](#)
 - virtual stored procedure properties [167](#)
 - web service operation properties [170](#)
- reverting
 - checked-out object [217](#)
- revertObject (infacmd mrs) [217](#)

- root directory
 - process variable [242](#)
- round-robin dispatch mode
 - description [252](#)
- row error log files
 - permissions [273](#)
- rule specifications
 - Content Management Service [34](#), [35](#)
- run-time statistics
 - persisting to the repository [231](#)

S

- safe mode
 - configuring for PowerCenter Integration Service [227](#)
 - PowerCenter Integration Service [225](#)
- samples
 - odbc.ini file [428](#)
- SAP BW Service
 - associated PowerCenter Integration Service [336](#)
 - creating [331](#)
 - disabling [333](#)
 - enabling [333](#)
 - general properties [334](#)
 - log events, viewing [337](#)
 - managing [330](#)
 - properties [335](#)
 - SAP Destination R Type (property) [331](#), [334](#)
- SAP BW Service log
 - viewing [337](#)
- SAP Destination R Type (property)
 - SAP BW Service [331](#), [334](#)
- SAP NetWeaver BI Monitor
 - log messages [337](#)
- saprfc.ini
 - DEST entry for SAP NetWeaver BI [331](#), [334](#)
- Scheduler Service
 - disabling [357](#)
 - enabling [357](#)
 - overview [353](#)
 - properties [354](#)
 - recycling [357](#)
- scorecards
 - purging results for [120](#)
- search analyzer
 - changing [210](#)
 - custom [210](#)
 - Model Repository Service [209](#)
- search index
 - Model Repository Service [209](#)
 - updating [210](#)
- Search Service
 - creating [344](#)
 - custom service process properties [344](#)
 - disable [345](#)
 - enable [344](#)
 - environment variables [343](#)
 - Maximum Heap Size [343](#)
 - recycle [345](#)
 - service process properties [343](#)
 - service properties [341](#)
- security
 - audit trail, creating [314](#)
 - web service security [124](#)
- SecurityAuditTrail
 - logging activities [314](#)
- server grid
 - licensed option [228](#)
- service levels
 - creating and editing [254](#)
 - description [254](#)
 - overview [263](#)
- service name
 - Web Services Hub [374](#)
- service process variables
 - list of [242](#)
- service role
 - Data Integration Service node [81](#)
- service variables
 - list of [229](#)
- services
 - system [346](#)
- session caches
 - description [273](#)
- session logs
 - directory [242](#)
 - overview [274](#)
 - permissions [273](#)
 - session details [274](#)
- session output
 - cache files [276](#)
 - control file [275](#)
 - incremental aggregation files [276](#)
 - indicator file [276](#)
 - performance details [274](#)
 - persistent lookup cache [277](#)
 - post-session email [275](#)
 - reject files [275](#)
 - session logs [274](#)
 - target output file [276](#)
- SessionExpiryPeriod (property)
 - Web Services Hub [378](#)
- sessions
 - caches [273](#)
 - DTM buffer memory [271](#)
 - output files [273](#)
 - performance details [274](#)
 - running on a grid [270](#)
 - session details file [274](#)
- shared library
 - configuring the PowerCenter Integration Service [235](#)
- shared storage
 - PowerCenter Integration Service [241](#)
 - state of operations [241](#)
- SID/Service Name
 - description [184](#)
- sort order
 - SQL data services [164](#)
- source data
 - blocking [268](#)
- source databases
 - connecting through ODBC (UNIX) [426](#)
- source files
 - Data Integration Service [95](#)
 - directory [242](#)
- source pipeline
 - pass-through [264](#)
 - reading [267](#)
 - target load order groups [267](#)
- sources
 - reading [267](#)
- SQL data service
 - changing the service name [168](#)
 - properties [164](#)

- SQL data services
 - Data Integration Service grid [130, 132](#)
- sqlplus
 - testing database connections [402](#)
- startup type
 - configuring applications [157](#)
 - configuring SQL data services [164](#)
- state of operations
 - PowerCenter Integration Service [241, 283](#)
 - PowerCenter Repository Service [296](#)
 - shared location [241](#)
- Stop option
 - disable Integration Service process [223](#)
 - disable PowerCenter Integration Service [223](#)
 - disable the Web Services Hub [375](#)
- Sybase ASE
 - connecting to Integration Service (UNIX) [422](#)
 - connecting to Integration Service (Windows) [410](#)
- Sybase ASE database requirements
 - PowerCenter repository [396](#)
- symmetric processing platform
 - pipeline partitioning [271](#)
- system parameters
 - Data Integration Service [95](#)
 - defining values [95](#)
- system services
 - overview [346](#)
 - Resource Manager Service [350](#)
 - Scheduler Service [353](#)

T

- table owner name
 - description [290](#)
- tablespace name
 - for repository database [290](#)
- tablespace recovery
 - IBM DB2 [123](#)
 - Microsoft SQL Server [123](#)
 - Oracle [123](#)
- tablespaces
 - single nodes [395](#)
- target databases
 - connecting through ODBC (UNIX) [426](#)
- target files
 - directory [242](#)
 - multiple directories [110](#)
 - output files [276](#)
- target load order groups
 - mappings [267](#)
- targets
 - output files [276](#)
 - session details, viewing [274](#)
- tasks
 - dispatch priorities, assigning [254](#)
- TCP/IP network protocol
 - Data Integration Service [79](#)
- team-based development
 - administering [216, 217](#)
 - command line program administration [217](#)
 - Objects view [216, 217](#)
 - troubleshooting [216, 218](#)
- temporary files
 - directory [242](#)
- temporary tables
 - description [116](#)
 - operations [117](#)
- temporary tables (*continued*)
 - rules and guidelines [119](#)
- Teradata
 - connecting to Informatica clients (UNIX) [424](#)
 - connecting to Informatica clients (Windows) [411](#)
 - connecting to Integration Service (UNIX) [424](#)
 - connecting to Integration Service (Windows) [411](#)
- Test Data Manager
 - repository [364](#)
- Test Data Manager repository
 - creating [364](#)
- Test Data Manager Service
 - advanced properties [364](#)
 - assign a new license [366](#)
 - components [359](#)
 - description [359](#)
 - general properties [361](#)
 - properties [360](#)
 - service properties [361](#)
 - steps to create [364](#)
 - TDM repository configuration properties [362](#)
 - TDM server configuration properties [363](#)
- thread pool size
 - configuring maximum [63](#)
- threads
 - creation [264](#)
 - mapping [264](#)
 - master [264](#)
 - post-session [264](#)
 - pre-session [264](#)
 - processing mappings [107](#)
 - reader [264](#)
 - transformation [264](#)
 - types [265](#)
 - writer [264](#)
- timeout
 - SQL data service connections [164](#)
 - writer wait timeout [235](#)
- Timeout Interval (property)
 - description [188](#)
- Tracing
 - error severity level [231, 378](#)
- TreatCHARAsCHAROnRead
 - option [233](#)
- TreatDBPartitionAsPassThrough
 - option [235](#)
- TreatNullInComparisonOperatorsAs
 - option [235](#)
- troubleshooting
 - grid for Data Integration Service [153](#)
 - grid for PowerCenter Integration Service [251](#)
 - versioning [216, 218](#)
- TrustStore
 - option [231](#)

U

- undoing
 - checked-out object [217](#)
- Unicode mode
 - code pages [272](#)
 - Data Integration Service [83](#)
 - Unicode data movement mode, setting [229](#)
- UNIX
 - connecting to ODBC data sources [426](#)
 - database client environment variables [402](#)
 - database client variables [402](#)

- unlocking
 - locked object [217](#)
- UnlockObject (infacmd mrs) [217](#)
- unregistering
 - local repositories [305](#)
 - plug-ins [313](#)
- upgrade error
 - Model Repository Service [384](#)
- URL scheme
 - Metadata Manager [187](#)
 - Web Services Hub [374, 377](#)
- user connections
 - closing [309](#)
 - managing [307](#)
 - viewing [308](#)
- user-managed cache tables
 - configuring [115](#)
 - description [114](#)
- users
 - notifications, sending [309](#)
- UTF-8
 - repository code page, Web Services Hub [374](#)
 - writing logs [231](#)

V

- ValidateDataCodePages
 - option [235](#)
- validating
 - source and target code pages [235](#)
- version control
 - enabling [303](#)
 - repositories [303](#)
- version control system
 - synchronizing [215](#)
- versioning
 - troubleshooting [216, 218](#)
- virtual column properties
 - configuring [167](#)
- virtual stored procedure properties
 - configuring [167](#)
- virtual table properties
 - configuring [166](#)
- virtual tables
 - caching in database [111](#)

W

- Warning
 - error severity level [231, 378](#)
- web service
 - changing the service name [171](#)
 - enabling [170](#)
 - operation properties [170](#)
 - properties [168](#)
 - resource properties [170](#)
 - security [124](#)
- web service security
 - authentication [124](#)
 - authorization [124](#)
 - HTTP client filter [124](#)
 - HTTPS [124](#)
 - isAuthenticationRequired [124](#)
 - message layer security [124](#)
 - pass-through security [124](#)
 - permissions [124](#)

- web service security (*continued*)
 - transport layer security [124](#)
- web services
 - Data Integration Service grid [130, 132](#)
- Web Services Hub
 - advanced properties [376, 378](#)
 - associated PowerCenter repository [380](#)
 - associated Repository Service [374, 380, 381](#)
 - associated repository, adding [380](#)
 - associated repository, editing [381](#)
 - associating a PowerCenter repository Service [374](#)
 - character encoding [377](#)
 - creating [374](#)
 - custom properties [376](#)
 - disable with Abort option [375](#)
 - disable with Stop option [375](#)
 - disabling [375](#)
 - domain for associated repository [374](#)
 - DTM timeout [378](#)
 - enabling [375](#)
 - general properties [376, 377](#)
 - host names [374, 377](#)
 - host port number [374, 377](#)
 - Hub Logical Address (property) [378](#)
 - internal host name [374, 377](#)
 - internal port number [374, 377](#)
 - keystore file [374, 377](#)
 - keystore password [374, 377](#)
 - license [374, 377](#)
 - location [374](#)
 - MaxISCConnections [378](#)
 - node [374](#)
 - node assignment [376, 377](#)
 - password for administrator of associated repository [380, 381](#)
 - properties, configuring [376](#)
 - security domain for administrator of associated repository [380](#)
 - service name [374](#)
 - SessionExpiryPeriod (property) [378](#)
 - tasks on Informatica Administrator [373](#)
 - URL scheme [374, 377](#)
 - user name for administrator of associated repository [380, 381](#)
 - user name for associated repository [374](#)
 - user password for associated repository [374](#)
 - version [374](#)
- Web Services Hub Service
 - custom properties [380](#)
- workflow
 - enabling [171](#)
 - IBM DB2 database requirements [399](#)
 - Microsoft SQL Server database requirements [400](#)
 - Oracle database requirements [400](#)
 - properties [171](#)
- workflow log files
 - directory [242](#)
- workflow logs
 - overview [274](#)
 - permissions [273](#)
- Workflow Orchestration Service properties
 - Data Integration Service [66](#)
- workflow output
 - email [275](#)
 - workflow logs [274](#)
- workflow schedules
 - safe mode [227](#)
- workflows
 - Data Integration Service grid [135, 141](#)
 - database requirements [399](#)
 - grids in local mode [138](#)

- workflows (*continued*)
 - grids in remote mode [145](#)
 - running on a grid [269](#)
 - Workflow Orchestration Service properties [66](#)
- writer wait timeout
 - configuring [235](#)
- WriterWaitTimeOut
 - option [235](#)

X

- XMLWarnDupRows
 - option [235](#)

Z

- ZPMSENDSTATUS
 - log messages [337](#)