Informatica® Dynamic Data Masking
9.8.0

Administrator Guide

Informatica Dynamic Data Masking Administrator Guide
9.8.0
December 2015

Publication Date: 2018-07-03

Table of Contents

Preface

The *Informatica Dynamic Data Masking Administrator Guide* contains information to help administrators manage and configure Dynamic Data Masking. This guide assumes that you have knowledge of your operating systems and relational database systems, which includes the database engines, flat files, and mainframe systems in your environment.

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.

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.
- 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 Velocity

Informatica Velocity is a collection of tips and best practices developed by Informatica Professional Services. Developed from the real-world experience of hundreds of data management projects, Informatica Velocity represents the collective knowledge of our consultants who have worked with organizations from around the world to plan, develop, deploy, and maintain successful data management solutions.

If you are an Informatica Network member, you can access Informatica Velocity resources at https://velocity.informatica.com.

If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

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

Introduction to Dynamic Data Masking Administration

This chapter includes the following topics:

Dynamic Data Masking Administration Overview

Dynamic Data Masking is a data security product that operates between an application and a database to prevent unauthorized access to sensitive information. Dynamic Data Masking intercepts requests sent to the database and applies data masking rules to the request to mask the data before it is sent back to the application.

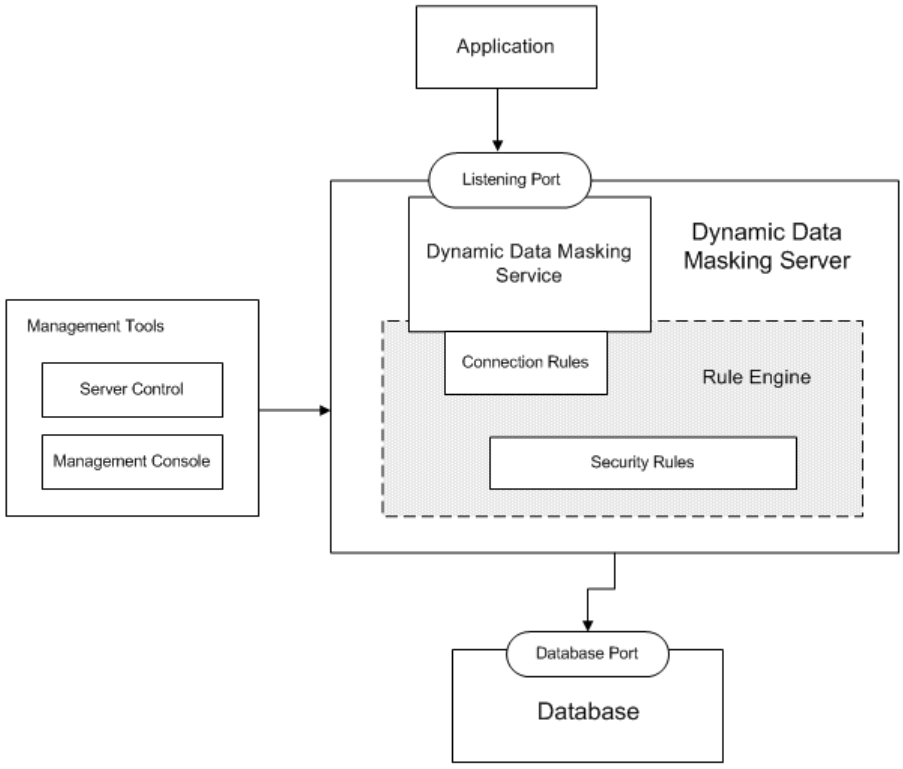
As an administrator, you use the Server Control program to manage the Dynamic Data Masking Server and start and stop the Dynamic Data Masking services. You use the Management Console to configure target databases, define listener ports, manage target databases, maintain system logs, and define rules. The administrative tasks that you perform help to ensure that Dynamic Data Masking operates effectively and efficiently.

Note: To prevent loss of data, such as connection rules, security rule sets, and database configurations, perform regular backups of the entire Dynamic Data Masking directory to a location on your network or an external storage location

Dynamic Data Masking Architecture

Dynamic Data Masking acts as a security layer between the application and the database to protect sensitive data stored in the database. The Dynamic Data Masking Server intercepts SQL requests sent to the database and uses a set of connection and security rules to determine how to process the request.

The following figure shows the architecture of Dynamic Data Masking and how the Dynamic Data Masking Server relates to the application and the database:



The Dynamic Data Masking Server listens on the port where the application sends database requests. When the application sends a request to the database, the Dynamic Data Masking Server receives the request before it goes to the database. The Rule Engine uses the connection rules and security rules to determine the action to perform on the incoming request. The Dynamic Data Masking service sends the modified request to the database. The database processes the request and sends the results back to the application.

Dynamic Data Masking provides management tools that you can use to manage the Dynamic Data Masking Server and set up connection and security rules. With the Server Control tool, you can start, stop, and manage the Dynamic Data Masking Server. On the Management Console, you can configure and manage the Dynamic Data Masking services and create and manage connection and security rules.

Dynamic Data Masking Components

Dynamic Data Masking includes server components to intercept and process database requests and a client component to manage the server.

Dynamic Data Masking has the following components:

Dynamic Data Masking Server

The Dynamic Data Masking Server provides services and resources to intercept database requests and perform data masking tasks.

The Dynamic Data Masking Server includes the following components:

- Dynamic Data Masking services
- Rule Engine

Dynamic Data Masking Service

The Dynamic Data Masking service listens on the listener port to monitor and routes incoming database requests.

You can run the following Dynamic Data Masking services:

- DDM for DB2. Listens for and routes database requests for an IBM DB2 database.
- DDM for Data Vault. Listens for and routes database requests for Data Vault.
- DDM for Hive. Listens for and routes database requests for a Hive database.
- DDM for Informix. Listens for and routes database requests in Informix native protocol to Informix databases.
- DDM for Informix (DRDA). Listens for and routes database requests in Distributed Relational Database Architecture protocol to Informix databases.
- DDM for JDBC. Listens for database requests for a database that uses JDBC connectivity.
- DDM for ODBC. Listens for database requests for a database that uses ODBC connectivity.
- DDM for Oracle. Listens for and routes database requests for an Oracle database.
- DDM for Microsoft SQL Server. Listens for and routes database requests for a Microsoft SQL Server database.
- DDM for Sybase. Listens for and routes database requests for a Sybase database.
- DDM for Teradata. Listens for and routes database requests for a Teradata database.

Rule Engine

The Rule Engine evaluates incoming database requests and applies connection and security rules to determine how to route requests and mask data. The Rule Engine can modify the database request based on the rules defined in the Dynamic Data Masking Server.

The Rule Engine applies the following types of rules:

- Connection rule. Defines the conditions and actions that the Rule Engine applies to determine how to route a database connection request received from an application.
- Security rule. Contains the conditions and actions that define what to do with the database SQL request and how to apply SQL rewrites that manipulate the returned SQL result set.

Server Control

Server Control is a command line program that you use to configure and manage the Dynamic Data Masking Server. Use Server Control to start or stop the Dynamic Data Masking Server and services or to change the port number or password for the Dynamic Data Masking Server.

Management Console

The Management Console is a client application that you use to manage the Dynamic Data Masking Server. You can use the Management Console to create and manage rules and to configure and manage connections to databases.

Dynamic Data Masking Process

Dynamic Data Masking acts as a proxy server for the database.

The application sends a request to the database. As a proxy server, the Dynamic Data Masking Server intercepts the request and the Rule Engine evaluates the request before it sends the request to the database.

Dynamic Data Masking uses the following process to apply data masking to a database request:

1. The Dynamic Data Masking service listens on the listener port for requests sent to the database. When the application sends a database connection request, the Dynamic Data Masking service receives the request instead of the database.
2. The Rule Engine uses a connection rule to determine how to process the incoming connection request. The connection rule defines the criteria to identify and route the database request. If the database request matches the criteria, the Rule Engine determines the action to perform on the request. The connection rule action can include routing the connection to a specified database, host, and port, and applying a security rule set. The connection rule can block the connection request. If the database has a direct action, the connection rule can return a redirect request back to the application with the database host and port and the application can connect directly to the database.

   For example, you can define an Informatica ETL process or batch in Dynamic Data Masking that bypasses Dynamic Data Masking and reduces overhead on internal processes.
3. If the connection rule specifies the Use Rule Set processing action, Dynamic Data Masking connects the client to the database and applies the security rules to the SQL request and determines the action to perform on the request.
   The security rules can apply an action such as blocking the SQL request, modifying the SQL statement, doing nothing, or auditing the request.
4. The database processes the request and returns the results to the application. Because Dynamic Data Masking changes the SQL request, the results the database sends back to the application might be different from the results of the original database request.

Dynamic Data Masking Example

A reporting tool sends a request to the database for employee salaries. The connection matcher specifies that the Rule Engine apply a rule set called salary_rules to all incoming requests from the reporting tool.

The salary_rules rule set applies a masking function to all requests that reference employee salaries. The rule set restricts access to management salaries and masks the first three digits of the employee salary column.

The Dynamic Data Masking service intercepts the incoming SQL request, identifies that the request references the salary tables, rewrites it with the masking function, and sends the rewritten request to the database. The database receives the request and sends masked data back to the application through the Dynamic Data Masking service.

Dynamic Data Masking Implementation

Set up Dynamic Data Masking based on the type of environment where you want to implement data masking and the requirements for protecting sensitive data within that environment.

Use the following general guidelines when you set up Dynamic Data Masking:

1. Install the Dynamic Data Masking Server on the database server.
   Dynamic Data Masking adds a processing layer between applications and databases. To reduce latency in data transfer, you can install the Dynamic Data Masking Server on the same machine as the database.

2. Configure the client application and database server so that the application sends database requests to the Dynamic Data Masking listener port.
   For example, you install the Dynamic Data Masking Server on the Oracle database server. Edit the `listener.ora` file on the database server and set the database port to a different port number. Then set the Dynamic Data Masking listener port number to match the port number to which the application sends database requests.
3. Before you start using Dynamic Data Masking, examine the database and classify data as highly sensitive, moderately sensitive, or not sensitive based on the data classification policy of your organization.
   The data classification you identify determines the rules you need to create in Dynamic Data Masking.
4. Categorize users according to their access permissions.
   Create user access scenarios to determine the applications that access sensitive data and identify the data that must go through Dynamic Data Masking.
5. In Dynamic Data Masking, create connection and security rules to implement the data classification and user and application access policies of your organization.

Dynamic Data Masking Environments

You can use data masking to protect sensitive information in a production database and in non-production databases used for development, testing, and training purposes.

Production Environments

Privileged users, such as production database administrators, technical support personnel, and quality assurance teams, access personally identifiable information as a daily part of their jobs.

Non-Production Environments

Non-production environments, such as pre-production and development testing environments, typically use real data to test applications. The data can include identifiers such as national identification numbers or bank accounts numbers.

Dynamic Data Masking eliminates exposure of sensitive data without affecting the ability of users to perform their job functions.

Setting Up Dynamic Data Masking

Install and configure Dynamic Data Masking based on the type of environment where you want to implement data masking and the requirements for protecting sensitive data within that environment.

After installation, complete the following steps to set up Dynamic Data Masking:

1. Log in to the Management Console. Use the user name *admin* and the Dynamic Data Masking Server password to log in to the Management Console for the first time.
2. Optionally, change the default Internal authentication scheme. You can use LDAP or Active Directory authentication to authorize a list of Dynamic Data Masking administrators.
3. Create a Dynamic Data Masking service. Configure the listener port number to match the port number where the client sends requests to the database.
4. Define the database connection properties for the database that requires data masking.
5. Create a connection rule. Configure the rule to identify the database requests that must be masked. Assign a database and a security rule set to the connection rule set.
6. Create a security rule set. Define the rules for masking the data sent back to the application.

Dynamic Data Masking applies the security rule set to SQL requests from a client or application that initiates a connection that uses the Dynamic Data Masking listener port. When you modify rules within the security rule set, Dynamic Data Masking immediately applies the modified rules on new SQL requests.

Management Console

The Management Console is the client component of the Dynamic Data Masking Server.

You can install the Management Console on a remote machine or the local system to manage the Dynamic Data Masking service. Use the Management Console to manage and configure domains and Dynamic Data Masking services, define connection rules for Dynamic Data Masking services, define security rules, and configure target databases.

Logging In to the Management Console

You can access to the Dynamic Data Masking components through the Management Console. Log in to the Management Console to manage target databases, configure listener ports, and define rules.

To log in to the Management Console, you need the server address and port number of the server that Dynamic Data Masking operates on and the administrator credentials.

Logging In to the Management Console on Windows

On Windows, open the Management Console through the Start menu.

1. Select **Start** > **Programs** > **Informatica** > **Dynamic Data Masking** > **Management Console**.

   The **Login** window appears.
2. Verify that the **Server Host** and **Port** display the correct information for the Dynamic Data Masking Server.
3. Enter the Dynamic Data Masking administrator user name and password. If you use LDAP authentication, the user name must be in LDAP format. Click **Connect**.

   A tree is visible in the Management Console after you login successfully.

Logging In to the Management Console on Linux

On Linux, start the Management Console with the `mng` script.

You must have the X Window server installed on the machine that you use to log in to the Management Console.

1. Open a terminal and navigate to the Dynamic Data Masking installation directory.

   For example, you might enter the following command:

   ```
   cd /home/Informatica/DDM
   ```
2. Run the following command to start the Management Console:

   ```
   ./mng
   ```

   The **Login** window appears.
3. Verify that the **Server Host** and **Port** display the correct information for the Dynamic Data Masking Server.

4. Enter the Dynamic Data Masking administrator user name and password. If you use LDAP authentication, the user name must be in LDAP format. Click **Connect**.

   A tree is visible in the Management Console after you log in.

Database Management

A database node contains references to databases. The Dynamic Data Masking Server controls access to the databases that the database nodes reference.

A database node can reference an Oracle, Microsoft SQL Server, DB2, Informix, Sybase, Data Vault, Hive, or Teradata database. The Management Console tree can contain an unlimited number of database nodes. You can create database nodes under domain nodes. Database nodes do not have child nodes. You can set user permissions on database nodes.

Dynamic Data Masking Server Management

A server node contains a reference to the Dynamic Data Masking Server. By default, the server node is located under the root domain after a new installation of the Dynamic Data Masking Server.

The Management Console contains one server node. Each Dynamic Data Masking instance associates with one Dynamic Data Masking Server. You connect to a server when you log into the Management Console. The Dynamic Data Masking Server manages databases located under a parent domain or all sub domains of the server node in the tree.

The server node has a domain node parent. The server node can have Dynamic Data Masking service child nodes. You can edit and move the server node.

Note: You cannot add or remove the Dynamic Data Masking Server node with the Add or Remove options in the Management Console menu.

Dynamic Data Masking Service Management

The Dynamic Data Masking service routes SQL queries to Oracle, Microsoft SQL Server, DB2, Informix, Sybase, Data Vault, Hive, and Teradata databases.

The Dynamic Data Masking Server can contain single service nodes for each database. Create service nodes under the server node. Service nodes cannot have child nodes. You can add, edit, and remove service nodes.

Each Dynamic Data Masking service routes requests to a specific type of database. For example, the Dynamic Data Masking for Oracle service routes requests to Oracle databases and the Dynamic Data Masking for DB2 service routes requests to DB2 databases.

Dynamic Data Masking Listener Ports

The Dynamic Data Masking service controls connections between the client and the database through the listener port.

You must configure the database listener port to forward connections to the Dynamic Data Masking listener port. How you configure the listener ports depends on whether the Dynamic Data Masking service runs on the database server or on a standalone server. You can define the listener port that the Dynamic Data Masking service uses through the Services Editor in the Management Console.

If the Dynamic Data Masking service runs on a standalone server, you must route application connection requests to the Dynamic Data Masking listener port.

Defining a Dynamic Data Masking Listener Port

Before you can define a listener port, you must run the netstat system utility to verify port availability.

1. In the Management Console, right-click on a Dynamic Data Masking service and select **Edit**.

   The Service Editor appears.
2. Click **Add Port**.
3. Enter the listener port for the Dynamic Data Masking service.
4. Click **OK**.

Deleting a Dynamic Data Masking Listener Port

If the Dynamic Data Masking service no longer uses a listener port, delete the listener port with the Service Editor.

1. In the Management Console, right-click the Dynamic Data Masking service and select **Edit.**

   The Service Editor appears.
2. Select the port to delete.
3. Click **Remove port**.
4. Click **OK**.

Configuration Management

The Dynamic Data Masking configuration files store information about the Management Console tree nodes and user access.

You can find the Dynamic Data Masking configuration files in the following location:

```
<Dynamic Data Masking installation>/cfg
```

The config.properties file contains the configuration parameters of the Dynamic Data Masking Server and service nodes in the Management Console.

The config.cfg file contains information on domain, database, and security rule set nodes in the Management Console. The config.cfg file is binary and you must not edit it. The config.cfg file and the public key file, config.pbk, have a digital signature. Dynamic Data Masking updates the digital signature when a user performs an operation on the Dynamic Data Masking Server through the Management Console. If Dynamic

Data Masking cannot verify the public key against the configuration file, the Dynamic Data Masking Server writes an error message to the server.log file and does not start.

Dynamic Data Masking Administrator Required Privileges

The Dynamic Data Masking administrator that creates the database connection must have privileges to access every object that the client user that receives masked data can access.

When the Dynamic Data Masking Server intercepts a client command, it might access client objects, such as tables, views, and stored procedures. If the administrator that created the database connection in Dynamic Data Masking cannot access the client objects, the query can return incorrect or unmasked data.

CHAPTER 2

Authentication

This chapter includes the following topics:

Authentication Overview

When you use the Management Console to connect to the Dynamic Data Masking Server, you must log in with a user name and password. Dynamic Data Masking user authentication depends on the authentication scheme that you configure.

You can configure the following authentication schemes for Dynamic Data Masking:

- LDAP authentication. Authentication scheme that uses the LDAP software protocol to locate organizations, groups, individuals, and other resources in a network.
- Active Directory authentication. Authentication scheme that uses the Active Directory service to enforce security for users and resources in a Windows network.
- Internal authentication. Authentication scheme that uses the Dynamic Data Masking Server password to authenticate users who log in to the Management Console.

Use the Management Console to set the authentication scheme for Dynamic Data Masking. By default, the Management Console uses internal authentication. The first time you log in to the Management Console, you enter the user name and the server password that you created when you installed the Dynamic Data Masking Server.

LDAP Authentication

You can use LDAP authentication to authenticate users who log in to the Management Console.

If you use LDAP authentication, you must provide the user name in LDAP format when you log in to the Management Console.

The following table describes the LDAP properties you configure if you use the LDAP authentication scheme:

Property	Description
Domain Server	Host name or IP address of the LDAP server.
Domain Server Port	Port number for the LDAP server.
Security Protocol	LDAP transport security protocol. Select SSL or None.
Administration Group	Distinguished name (DN) of the LDAP group to use for authentication.
Root DN	Base distinguished name (DN) of the LDAP domain in which to begin user lookups.
User Object Class	Object class type used for the LDAP user object.
User MemberOf Attribute	Locate users by attribute. Name of the user attribute to use to identify the groups associated with a user.
Group Object Class	Object class type used for the LDAP group object.
Group MemberOf Attribute	Locate users by attribute. Name of the group attribute to use to identify the groups associated with a user.
Group Members Attribute	Locate users by group. Name of the attribute to use to identify the members of a group.

Active Directory Authentication

You can use Active Directory authentication to authenticate users who log in to the Management Console.

The following table describes the properties you configure if you use the Active Directory authentication scheme:

Property	Description
Domain Server	Host name or IP address of the Active Directory server.
Domain Server Port	Port number for the Active Directory server.
Domain	Name of the domain name that contains the Active Directory server
Administration Group	Name of the Active Directory group to use for authentication.

Internal Authentication

Dynamic Data Masking provides a simple authentication scheme that authenticates a user based on the Dynamic Data Masking Server password.

When you install the Dynamic Data Masking Server, you must create a password for the server. Dynamic Data Masking uses the server password for authentication when you initially log in to the Management Console.

The Management Console login requires a user name and password. The Internal authentication scheme uses only the password for authentication. It uses the user name to identify the user and to track user activity in the Management Console.

You can use the Server Control command SetInternalPassword to change the Dynamic Data Masking Server password at any time.

Admin User Name

The user name `admin` is a key user name for Dynamic Data Masking authentication. If you log in to the Management Console with the `admin` user name, Dynamic Data Masking uses the server password for authentication.

The `admin` user name supercedes the type of authentication configured for the Dynamic Data Masking. You can log in to the Management Console with the `admin` user name at any time. If you log in to the Management Console with the `admin` user name, Dynamic Data Masking authenticates based on the server password even if the Dynamic Data Masking is configured to use LDAP or Active Directory authentication.

When you log in as `admin`, Dynamic Data Masking uses internal authentication only for the session. It does not permanently change the authentication scheme configured for Dynamic Data Masking. When you log out, you can log back in with an LDAP or Active Directory user account.

If Dynamic Data Masking uses LDAP authentication and the configuration of the LDAP server or network changes, you might not be able to log in to the Management Console. You must update the LDAP server attributes in the Management Console for LDAP authentication to work properly. Log in to the Management Console with the `admin` user name and server password and update the attributes of the LDAP server. The next time you log in, you can use LDAP authentication.

Setting Up Authentication

Use the Management Console to set up the authentication system for Dynamic Data Making.

To set the authentication for the first time, you must log in with a user name and the server password. You set the initial server password when you install the Dynamic Data Masking Server.

1. Log in to the Management Console.
2. In the Management Console tree, select the Dynamic Data Masking Server.

   The properties of the Dynamic Data Masking Server displays on the right panel, showing the current authentication scheme.
3. Click **Tree** > **Edit** to edit the properties of the Dynamic Data Masking Server.
4. In the **Edit** window, select the authentication scheme you want to use.

   The **Edit** window displays the properties required to connect to the authentication server.
5. Enter the authentication server domain and user properties.

- For the LDAP authentication scheme, enter the LDAP domain, group, and user properties. Consult the LDAP administrator to get the correct information.
- For the Active Directory authentication scheme, enter the Active Directory domain properties and the administration group. Consult the Active Directory administrator to get the correct information.
- For the Internal authentication scheme, you do not need to enter any information. Use the Server Control command SetInternalPassword to manage the Dynamic Data Masking Server password.

6. Click OK.

Log out and log in again to the Management Console to verify that the selected authentication scheme is in effect.

CHAPTER 3

Connection Management

This chapter includes the following topics:

Connection Management Overview

Use the Add Database window to add a database to the Management Console tree. Select a database type and define database parameters. Test the database connection to verify that the Dynamic Data Masking service can access the database.

Configuring the Target Database

Define the database parameters that the Dynamic Data Masking service uses to connect to the target database. The target database configuration specifies the database, user account, and connection settings.

1. In the Management Console, click **Tree** > **Add Database**.

   The **Add Database** window appears.

2. Select the type of database to use.
3. In the **DDM Database Name** field, enter a name for the database connection you are creating.
4. Enter the database connection parameters and user credentials.

   The settings differ based on the type of database you connect to.
5. Click **Test Connection** to validate the connection to the database.
6. Click **OK**.

Testing a Connection

When you test a connection, the Dynamic Data Masking service validates the database configuration and connection information.

Define all connection parameters before you test a database connection. If the Dynamic Data Masking service cannot connect to the database, check with the database administrator to verify the database configuration information.

1. Select a database node in the Management Console tree and click **Tree** > **Edit**.

   The **Edit** window appears.
2. Click **Test Connection**.

   A JDBC connection to the Dynamic Data Masking service opens. The services uses the defined database connection parameters. A confirmation message appears if the connection is valid.

Data Vault Connection Management

Select the FAS database type to add a Data Vault connection node to the Management Console tree.

Use **Test Connection** to verify that the Dynamic Data Masking service can access the database.

Data Vault Connection Parameters

Define the following connection parameters for a Data Vault connection:

DDM Database Name

Name for the database node that appears in the Management Console tree.

Server Address

Server host name or TCP/IP address for the Data Vault.

Note: Verify that there is no firewall that prohibits the Dynamic Data Masking Server from connecting to the Data Vault server and port number.

Server Port

TCP/IP listener port for the Data Vault.

FAS Database Name

Database name for the Data Vault.

DBA Username

User name for the Data Vault user account to log in to the Data Vault.

DBA Password

Password for the Data Vault user.

DB2 Connection Management

Select the DB2 database type to add a DB2 database connection node to the Management Console tree.

The DB2 connection request does not contain information about the database. You must define the target database that Dynamic Data Masking forwards the request to. Make a connection rule that uses the Switch to Database rule action to define the target database. Specify a database in the rule that corresponds to the Dynamic Data Masking Database Name parameter.

Use **Test Connection** to verify that the Dynamic Data Masking service can access the database.

DB2 Connection Parameters

Define the following connection parameters for a DB2 database:

DDM Database Name

Name for the database in the Management Console tree.

Server Address

Server host name or TCP/IP address for the DB2 database.

Note: Verify that there is no firewall that prohibits the Dynamic Data Masking Server from connecting to the database server and port number.

Server Port

TCP/IP listener port for the DB2 database.

DB2 Database Name

Database name for the DB2 database.

Optional Parameters

Additional parameters for the Informatica driver for DB2.

For example, if the DB2 database is configured with the SERVER_ENCRYPT authentication method, you might enter the following parameter:

```
AuthenticationMethod=encryptedUIDPassword
```

DBA Username

User name for the database user account to log in to the DB2 database.

DBA Password

Password for the database user.

DB2 Database Administrator Required Privileges

The database administrator must have privileges to access sensitive tables and columns.

Use the DB2 Control Center to create a privileged user, <DBA Username>, that corresponds to an administrator user on your operating system.

Run the following command from an SQL window:

```
GRANT SETSESSIONUSER ON PUBLIC TO <DBA Username>
```

If the DB2 database is encrypted and uses an ODBC driver, to allow impersonation, the database administrator must have the following privilege to access the database table:

```
SYSIBMADM.SNAPAPPL_INFO
```

Generic Database Connection Management

Select the Generic Database database type to add a Generic Database connection node to the Management Console. Use the DDM for JDBC and DDM for ODBC services and a Generic Database node to mask data for a database that uses JDBC or ODBC connectivity.

You can use the Generic Database node for databases that do not have a dedicated database node. For example, use the Generic Database node with a Netezza or Greenplum database.

You provide information about the database and the database drivers. The client sends the request to Dynamic Data Masking. Dynamic Data Masking connects to the database to retrieve metadata in order to mask the SQL statement. Dynamic Data Masking masks the request and sends the masked request back to the client. The client then sends the masked request to the database. The database returns masked data.

Make a connection rule that uses the Switch to Database rule action to define the target database. Specify a database in the rule that corresponds to the Dynamic Data Masking Database Name parameter.

Use **Test Connection** to verify that the Dynamic Data Masking service can access the database.

Generic Database Connection Parameters

Define the following connection parameters for a database that uses an ODBC or JDBC connection:

DDM Database Name

Name for the database in the Management Console tree.

Driver Class Name

Fully qualified class name of the target database driver.

For example, you might enter the following text:

```
org.netezza.Driver
```

Connect String (URL)

JDBC connection string used to connect to the database.

For example, you might enter the following text:

```
jdbc:netezza://hostname:port/database_name
```

Optional Parameters

Optional parameters for the Informatica driver for the database.

DSN Name

Logical data source name used to connect to the database.

DBA Username

Username for the database administrator account to log in to the database.

DBA Password

Password for the database administrator.

Unary Table

Name of the unary table for the database.

Bind Argument Representation

Representation of the bind argument for a PL/SQL stored function.

Supports ANSI Join

Select if the database supports ANSI join syntax.

Are Function Calls Allowed in Select

Select if the database allows a function call to be included in a SELECT statement.

Are Parentheses Allowed for Function Call with No Arguments

Select if the database allows parentheses in a function call that does not have any arguments.

Impersonation Commands

Impersonation commands for the database, separated by a semicolon (;). Words that start with a dollar sign ($) are variables and must be replaced with values from the connection context.

Cleanup Commands

Cleanup commands for the database, separated by a semicolon (;).

Sanity Check Script

Sanity check script for the database.

Hive Connection Management

Select the Hive database type to add a Hive database connection node to the Management Console tree.

You can use the Hive database type to access Hadoop-compatible file systems. To connect to multiple Hive databases, create database nodes in the Management Console and enter different sets of driver files in the classpath parameter.

Use **Test Connection** to verify that the Dynamic Data Masking service can access the database.

Hive Connection Parameters

Define the following connection parameters for a Hive database:

DDM Database Name

Name for the database that appears in the Management Console tree.

Server Address

Server host name or TCP/IP address for the Hive database.

Note: Verify that there is no firewall that prohibits the Dynamic Data Masking Server from connecting to the database server and port number.

Server Port

TCP/IP listener port for the Hive database.

Hive Database Name

Database name for the Hive database.

Driver Classpath

Classpath of the Hive database driver files on the server. Use semicolons to separate multiple classpaths.

You can use an asterisk (*) to indicate all the jar files in a directory. Dynamic Data Masking ignores files in the directory that are not jar files. For example, you might enter the following location for Windows:

```
C:\JDBC_Drivers\Hive\hive\*;C:\JDBC_Drivers\Hive\hadoop\*
```

DBA Username

Optional user name for the database user account to log in to the Hive database.

DBA Password

Optional password for the database user.

Informix Connection Management

Select the Informix database type to add an Informix database connection node to the Management Console tree.

The Informix connection request does not contain information about the database. You must define the target database that Dynamic Data Masking forwards the request to. Make a connection rule that uses the Switch to Database rule action to define the target database. Specify a database in the rule that corresponds to the Dynamic Data Masking Database Name parameter.

Use **Test Connection** to verify that the Dynamic Data Masking service can access the database.

Informix Connection Parameters

Define the following connection parameters for an Informix database:

DDM Database Name

Name for the database in the Management Console tree.

Server Address

Server host name or TCP/IP address for the Informix database.

Note: Verify that there is no firewall that prohibits the Dynamic Data Masking Server from connecting to the database server and port number.

Server Port

TCP/IP port of the listener receiving requests in Informix native protocol. The DDM for Informix service uses this port to communicate with the Informix database.

DRDA Port

TCP/IP port of the listener receiving requests in Informix DRDA protocol. The DDM for Informix (DRDA) service uses this port to communicate with the Informix database.

Informix Database Name

Database name for the Informix database.

Informix Server Name

The Informix server name specific to the database instance.

DBA Username

User name for the database user account to log in to the Informix database.

DBA Password

Password for the database user.

Microsoft SQL Server Connection Management

Select the Microsoft SQL Server database type to add a Microsoft SQL Server database connection node to the Management Console tree.

Use **Test Connection** to verify that the Dynamic Data Masking service can access the database.

Microsoft SQL Server Connection Parameters

Define the following connection parameters for a Microsoft SQL Server database:

DDM Database Name

Name for the database in the Management Console tree.

Server Address

Server host name or TCP/IP address for the Microsoft SQL Server database.

Note: Verify that there is no firewall that prohibits the Dynamic Data Masking Server from connecting to the database server and port number.

Server Instance Name

The instance name of the Microsoft SQL Server database.

If the Microsoft SQL Server database is configured to use dynamic port allocation, you can enter the Server Instance Name to identify the listener port. If you enter the instance name, you do not need to enter a Server Port number.

Server Port

TCP/IP listener port for the Microsoft SQL Server database. If you enter the Server Port number, you do not need to enter a Service Instance Name.

Optional Parameters

Additional parameters for the Informatica driver for Microsoft SQL Server.

DBA Username

User name for the database user account to log in to the Microsoft SQL Server database.

DBA Password

Password for the database user.

Microsoft SQL Server Database Administrator Required Privileges

The database administrator must have privileges to access sensitive tables and columns.

Log in as the system administrator and run the following commands:

- `USE master;`
- `CREATE LOGIN <DBA Username> WITH PASSWORD=<DBA Password>, DEFAULT_DATABASE = <default database>;`
- `GRANT CONTROL SERVER TO <DBA Username>;`
- `USE <default database>;`
- `CREATE USER <db user> FOR LOGIN <dba user login>;`

Netezza Connection Management

You can use a Microsoft SQL Server connection to access a Netezza database. However, when you create a new database node for Netezza, Informatica recommends that you use the Generic Database node.

To use Dynamic Data Masking with Netezza, you must configure a server link between a Microsoft SQL Server database and the Netezza database. The client tool or application that you use to access the database must support Microsoft SQL Server.

When you send a request to the Microsoft SQL Server database, the request passes through the Dynamic Data Masking Server, which alters the request. The Dynamic Data Masking Server applies masking and auditing rules and uses OpenQuery to direct the request through the Microsoft SQL Server database to the Netezza database. The Netezza database returns masked data through the Microsoft SQL Server database to the Dynamic Data Masking Server.

To connect to a Netezza database, create a Microsoft SQL Server connection node in the Management Console.

Oracle Connection Management

Select the Oracle database type to add an Oracle database connection node to the Management Console tree.

If the Dynamic Data Masking service runs on the Oracle database server, you must switch the Oracle listener to a hidden port. Edit the `listener.ora` file to change the Oracle listener to a port that is not in use. When you change the Oracle listener to a hidden port, applications connect to the Dynamic Data Masking listener port instead of the database.

To route applications to the Dynamic Data Masking listener port, you must edit `tnsnames.ora` to add a database alias to `tnsnames.ora` for the Dynamic Data Masking service. Dynamic Data Masking uses the database alias to listen to incoming connections to the database.

Informatica recommends that you change the Oracle listener to a port that is not the default. If you do not change the port, it is possible for an unauthorized user to edit the connection properties from the client and bypass the Dynamic Data Masking Server. Also, when you change the Oracle listener and configure Dynamic Data Masking to listen on the default Oracle listener, you do not have to edit the client connection properties.

Use **Test Connection** to verify that the Dynamic Data Masking service can access the database. If a database defines multiple instances, **Test Connection** validates each Oracle instance cyclically. The test connection verifies each Oracle instance.

Oracle Connection Parameters

Define the following connection parameters for an Oracle database:

DDM Database Name

Logical name defined for the target database.

Instance Name

Instance name for the target database.

Listener Address

Server host name or TCP/IP address for the target database.

Listener Port

TCP/IP listener port for the target database.

Service Name

Service name for the target database. Dynamic Data Masking determines the target database based on the service name or SID in the client connection request.

DBA Username

User name for the database user account to log in to the Oracle database.

DBA Password

Password for database user.

Oracle Database Administrator Required Privileges

The database administrator must have privileges to access sensitive tables and columns.

Run the following database commands to create a database administrator user name and password for Dynamic Data Masking:

- `CREATE USER ACTIVE IDENTIFIED BY <XXXX>`
- `ALTER USER ACTIVE QUOTA UNLIMITED ON USERS`
- `GRANT BECOME USER`
- `GRANT CREATE SESSION TO ACTIVE`
- `GRANT ALTER SESSION`
- `GRANT SELECT ANY TABLE TO ACTIVE`
- `GRANT SELECT ANY DICTIONARY TO ACTIVE`

Using DBLink

If you use DBLink to access the Oracle database, you must set DBLink to `PUBLIC`. If DBLink is not set to `PUBLIC`, Dynamic Data Masking will not be able to access the database.

Changing the Listener Port

If the Dynamic Data Masking service runs on a standalone server, you must modify the `tnsnames.ora` file to add a database alias for Dynamic Data Masking service. The `tnsnames.ora` file is a configuration file that defines the addresses that the client uses to connect to the database. When you add a database alias to the file, applications send requests to the Dynamic Data Masking listener instead of the database listener.

Before you change the `tnsnames.ora` file, back up the original copy.

1. Open `tnsnames.ora`. By default, this file is located in the following location: `<Oracle install directory>/app/oracle/product/<product version>/server/NETWORK/ADMIN`
2. Find the following entry in the `tnsnames.ora` file:

```
DBNAME=(DESCRIPTION=
        (ADDRESS=(PROTOCOL=TCP)(HOST=dbServer)(PORT=1521))
                (CONNECT_DATA=(SERVICE_NAME=prod.mycompany.com)))
```

3. Replace the service listener port and host with the Dynamic Data Masking host and listener port. The following tnsnames.ora file shows the entry updated with the updated host and listener port.

```
DBNAME=(DESCRIPTION=
   (ADDRESS=(PROTOCOL=TCP)(HOST=DynamicDataMasking)(PORT=1525))
     (CONNECT_DATA=(SERVICE_NAME=prod.mycompany.com)))
```

Configuring the Oracle Target Database Example

The target database configuration defines the connection parameters that the Dynamic Data Masking service uses to connect to the database. The easiest way to identify the correct database connection settings is to use the information from `tnsnames.ora`. The following example shows how you can use the database parameters from `tnsnames.ora` to configure a database in the Database Editor.

The following entry is from `tnsnames.ora`:

```
ORA11G =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 127.0.0.1)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = ora11g)
    )
  )
```

The following table shows how the parameters in `tnsnames.ora` correspond to the parameters in the Management Console:

Parameter Value in tnsnames.ora	Parameter Name in the Database Editor
ORA11G	Instance Name
localhost	Listener Address
1521	Listener Port
ora11g	Service Name

Sybase Connection Management

Select the Sybase database type to add a Sybase database connection node to the Management Console tree.

The Sybase connection request does not contain information about the database. You must define the target database that Dynamic Data Masking forwards the request to. Make a connection rule that uses the Switch to Database rule action to define the target database. Specify a database in the rule that corresponds to the Dynamic Data Masking Database Name parameter.

Use **Test Connection** to verify that the Dynamic Data Masking service can access the database.

Sybase Connection Parameters

Define the following connection parameters for a Sybase database:

DDM Database Name

Name for the database in the Management Console tree.

Server Address

Server host name or TCP/IP address for the Sybase database.

Note: Verify that there is no firewall that prohibits the Dynamic Data Masking Server from connecting to the database server and port number.

Server Port

TCP/IP listener port for the Sybase database.

Optional Parameters

Additional parameters for the Informatica driver for Sybase.

DBA Username

User name for the database user account to log in to the Sybase database.

DBA Password

Password for the database user account.

Sybase Database Administrator Required Privileges

The database administrator must have privileges to access sensitive tables and columns.

Log in to a Sybase client as an administrator that is not the Dynamic Data Masking administrator and run the following commands:

- `sp_addlogin <dba user>, <dba password>`
- `grant role sso_role to <dba user>`

Search and Replace Rule

If you use Sybase with ODBC drivers and have the quoted_identifier option set to off or the query contains a table name with aliases, create a search and replace security rule.

If the quoted_identifier option is set to off or the query contains a table name with aliases, a query sent to the database through Dynamic Data Masking returns an error. You must create a security rule that removes double quotes from the query.

Queries to the database must use a masking rule with the following parameters:

Rule Name

The name of the rule. For example, Replace.

Matcher

Select the Any matcher. The rule applies to all queries sent to the database.

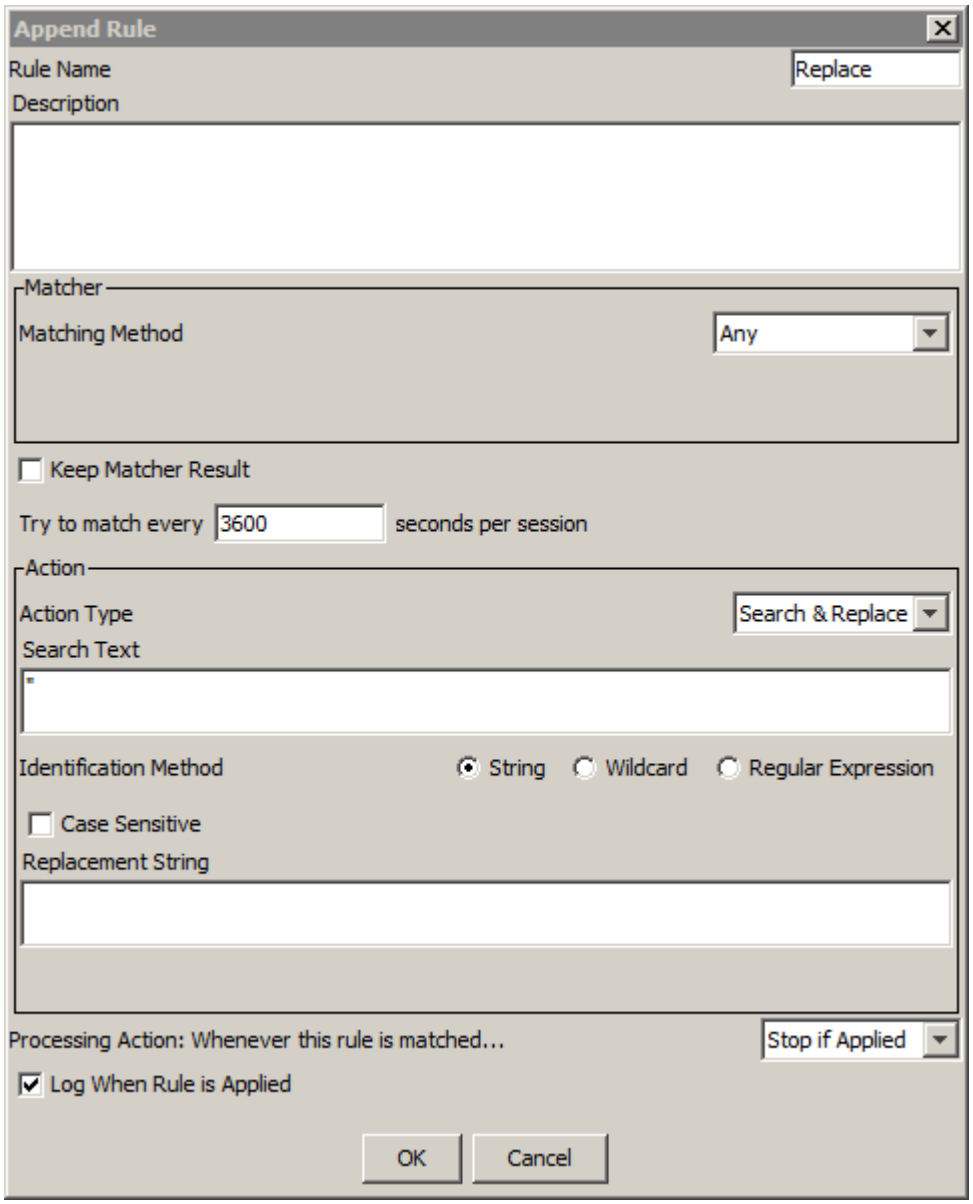
Rule Action

Select the Search and Replace rule action. In the Search Text field, enter a double quote ("). Leave the Replacement String field blank. In the query, the Rule Engine removes double quotes.

Log When Rule is Applied

Select the Log When Rule is Applied check box to create a line in the rule.log file when the Rule Engine applies the rule.

The following image is an example of the search and replace rule:



Teradata Connection Management

Select the Teradata database type to add a Teradata database connection node to the Management Console tree.

The Teradata connection request does not contain information about the database. You must define the target database that Dynamic Data Masking forwards the request to. Make a connection rule that uses the Switch to Database rule action to define the target database. Specify a database in the rule that corresponds to the Dynamic Data Masking Database Name parameter.

To access a Teradata database, you must manually add Teradata drivers into the Dynamic Data Masking installation directory.

Use **Test Connection** to verify that the Dynamic Data Masking service can access the database.

Teradata Connection Parameters

Define the following connection parameters for a Teradata database:

DDM Database Name

Name for the database in the Management Console tree.

Server Address

Server host name or TCP/IP address for the Teradata database.

Note: Verify that there is no firewall that prohibits the Dynamic Data Masking Server from connecting to the database server and port number.

Server Port

TCP/IP listener port for the Teradata database.

DBA Username

User name for the database user account to log in to the Teradata database.

DBA Password

Password for the database user account.

Configuring the Teradata Drivers

To access a Teradata database, you must manually place the Teradata drivers in the Dynamic Data Masking installation directory.

1. Download the Teradata drivers from the Teradata website. You need the following drivers:
   - tdgssconfig.jar
   - terajdbc4.jar
2. Stop the Dynamic Data Masking Server Windows service.

3. Open a Server Control window and run the following commands:
   - `server stop`
   - `server remove`
4. Save the Teradata drivers in the following directory:
   `<Dynamic Data Masking installation>\lib\ext`
5. From Server Control, run the following command:
   `server start`

Troubleshooting

This section provides solutions for errors that you might receive when you connect to a database with Dynamic Data Masking.

No Listener Defined

The `TNS: No Listener` error indicates that clients cannot reach the Dynamic Data Masking listener port.

If you receive this error, verify that the firewall configuration has the Dynamic Data Masking listener port and administrator port open. For example, the default Dynamic Data Masking listener port for Oracle is 1525 and the default administrator port is 8195.

Note: The `TNS: No Listener` error is an error for connections to Oracle databases. A similar error might appear for a different database.

Database Refuses Connection

If you do not define the service name for the database, the target database refuses a connection. If the database refuses the connection, you receive the `TNS connection refused` error.

To resolve the TNS connection error, define the database service name.

1. Right-click on the database name and select **Edit**.
2. Click **Add** and define a service name.
3. Click **OK.**

Dynamic Data Masking Service Refuses Connection Request

If the Dynamic Data Masking service refuses a connection from a client, you receive an `IO Exception, Connection Refused` error.

The following table describes the possible reasons for the connection refusal:

Reason	Solution
Error in the listener address	Run a ping command from the system running the Management Console to the Dynamic Data Masking service. Use the Dynamic Data Masking host name. For example: `ping 10.65.48.73`.
Error in the listener port	Verify that the firewall has the Dynamic Data Masking listener port and administrator port open.
Undefined service name	Verify the database service name. Add a service name if one does not exist.
Database server is down	Restart the database.

CHAPTER 4

JDBC Client Configuration

This chapter includes the following topics:

JDBC Client Configuration Overview

Before you use Dynamic Data Masking to mask data for a database that uses a JDBC connection, you must configure the client.

The Dynamic Data Masking installation contains a .jar file that you must save on the client machine. The .jar file contains a Java agent, a transformer that intercepts the method calls of JDBC objects, and proxy classes for JDBC classes. You can find the .jar file in the following location:

```
<Dynamic Data Masking installation>\Wrappers\jdbc\GenericJDBC.jar
```

You must perform additional configuration steps based on the client and operating system.

Note: Because the client looks for the Dynamic Data Masking service on the host and port, the client connection fails when the Dynamic Data Masking Server is down. To connect to the database, the Dynamic Data Masking Server must be running.

Apache Tomcat Configuration

Follow the configuration steps for the Apache Tomcat client based on the operating system.

Configure Apache Tomcat for Windows

Complete the following steps to configure Apache Tomcat for Windows:

1. Copy the GenericJDBC.jar file. You can find the file in the following location:

   ```
   <Dynamic Data Masking installation>\Wrappers\jdbc\GenericJDBC.jar
   ```

2. Save the GenericJDBC.jar file in the following location:

   ```
   <Apache Tomcat installation>\lib
   ```

3. Find the catalina.bat file. You can find the file in the following directory:

   ```
   <Apache Tomcat installation>\bin\catalina.bat
   ```

4. Save a backup of the catalina.bat file.
5. Open the catalina.bat file in a text editor and find the `setlocal` line. To append the -javaagent argument to the Java command line, enter the following text under the `setlocal` line:

   ```
   set JAVA_OPTS=-javaagent:..\lib
   \GenericJDBC.jar=host:<ddm_host>,port:<ddm_generic_service_port>
   ```

6. To set the classpath, add the following line under the line that you added in the previous step:

   ```
   set CLASSPATH=%CLASSPATH%;..\lib\GenericJDBC.jar;
   ```

7. Save catalina.bat.

Configure Apache Tomcat for Linux

Complete the following steps to configure Apache Tomcat for Linux:

1. Copy the GenericJDBC.jar file. You can find the file in the following location:

   ```
   <Dynamic Data Masking installation>/Wrappers/jdbc/GenericJDBC.jar
   ```

2. Save the GenericJDBC.jar file in the following location:

   ```
   <Apache Tomcat installation>/lib
   ```

3. Find the catalina.sh file. You can find the file in the following directory:

   ```
   <Apache Tomcat installation>/bin/catalina.bat
   ```

4. Save a backup of the catalina.sh file.
5. Open the catalina.sh file in a text editor and find the line that contains `#JAVA_OPTS=`. To append the -javaagent argument to the Java command line, enter the following text under the line that contains `#JAVA_OPTS=`:

   ```
   export JAVA_OPTS=-javaagent:../lib/
   GenericJDBC.jar=host:<ddm_host>,port:<ddm_generic_service_port>
   ```

6. To set the classpath, add the following line under the line that you added in the previous step:

   ```
   set CLASSPATH=$CLASSPATH:../lib/GenericJDBC.jar
   ```

7. Save catalina.sh.

Aqua Data Studio Configuration

Complete the following steps to configure the Aqua Data Studio client:

1. Copy the GenericJDBC.jar file. You can find the file in the following location:

   ```
   <Dynamic Data Masking installation>\Wrappers\jdbc\GenericJDBC.jar
   ```

2. Save the GenericJDBC.jar file in the following location:

   ```
   <Aqua Data Studio installation>\lib
   ```

3. If you want to use datastudio.bat to launch Aqua Data Studio, complete the following steps:
   a. Find the datastudio.bat file. You can find the file in the Aqua Data Studio installation directory.
   b. Save a backup of the datastudio.bat file.
   c. Open the datastudio.bat file in a text editor. To set the environment variable, add the following text to the last line of text in the file:

      ```
      -javaagent:.\lib\GenericJDBC.jar= host:<ddm_host>,port:<ddm_generic_service_port>
      ```

      The line that you modified is similar to the following text:

      ```
      java -Dfile.encoding=UTF-8 -Xms512M - true -javaagent:.\lib\GenericJDBC.jar=
      host:<ddm_host>,port:<ddm_generic_service_port> -cp ".\lib\ads.jar;%ADS_PATH%"
      com.aquafold.datastudio.DataStudio
      ```

   d. Save datastuido.bat.
4. If you want to use datastudio.exe to launch Aqua Data Studio, complete the following steps:
   a. Find the datastudio.cfg file. You can find the file in the Aqua Data Studio installation directory.
   b. Save a backup of the datastudio.cfg file.
   c. Open the datastudio.cfg file in a text editor. To append the -javaagent argument to the Java command line, add the following Java agent argument before the `-cp` text:

      ```
      -javaagent:.\lib\GenericJDBC.jar= host:<ddm_host>,port:<ddm_generic_service_port>
      ```

Oracle SQL Developer Configuration

Complete the following steps to configure the Oracle SQL Developer client:

1. Copy the GenericJDBC.jar file. You can find the file in the following location:

   ```
   <Dynamic Data Masking installation>\Wrappers\jdbc\GenericJDBC.jar
   ```

2. Save the GenericJDBC.jar file in the following location:

   ```
   <SQL Developer installation>\sqldeveloper\lib
   ```

3. Find the sqldeveloper.conf file. You can find the file in the in the following location:

   ```
   <SQL Developer installation>\sqldeveloper\bin
   ```

4. Save a backup of the sqldeveloper.conf file.
5. Open the sqldeveloper.conf file in a text editor. To append the -javaagent argument to the Java command line, add the following text to the file:

   ```
   AddVMOption  -javaagent:..\lib\GenericJDBC.jar=
   host:<ddm_host>,port:<ddm_generic_service_port>
   ```

6. Save sqldeveloper.conf.
7. Use sqldeveloper.exe to launch Oracle SQL Developer.

SQuirreL Configuration

Complete the following steps to configure the SQuirreL SQL client:

1. Copy the GenericJDBC.jar file. You can find the file in the following location:

   ```
   <Dynamic Data Masking installation>\Wrappers\jdbc\GenericJDBC.jar
   ```

2. Save the GenericJDBC.jar file in the following location:

   ```
   <SQuirreL installation>\lib
   ```

3. Find the squirrel-sql.bat file. You can find the file in the SQuirreL installation directory.
4. Save a backup of the squirrel-sql.bat file.
5. Open the squirrel-sql.bat file in a text editor. To append the -javaagent argument to the Java command line, add the following text to the penultimate line of text in the file:

   ```
   -javaagent:.\lib\GenericJDBC.jar= host:<ddm_host>,port:<ddm_generic_service_port>
   ```

   The line that you modified is similar to the following text:

   ```
   "%LOCAL_JAVA%"  -verbose -Xmx256m -Dsun.java2d.noddraw=true -javaagent:.\lib
   \GenericJDBC.jar= host:<ddm_host>,port:<ddm_generic_service_port>  -cp %SQUIRREL_CP
   %  -splash:"%SQUIRREL_SQL_HOME%/icons/splash.jpg"
   net.sourceforge.squirrel_sql.client.Main %TMP_PARMS%
   ```

6. Save squirrel-sql.bat.

WebLogic Configuration

Follow the configuration steps for the WebLogic client based on the operating system.

Configure WebLogic for Windows

Complete the following steps to configure WebLogic for Windows:

1. Copy the GenericJDBC.jar file. You can find the file in the following location:

   ```
   <Dynamic Data Masking installation>\Wrappers\jdbc\GenericJDBC.jar
   ```

2. Save the GenericJDBC.jar file in the following location:

   ```
   ...\Middleware\user_projects\domains\base_domain\lib
   ```

   For example, you might save the file in the following location:

   ```
   C:\Oracle\Middleware\user_projects\domains\base_domain\lib
   ```

3. Find the startWebLogic.cmd file. You can find the file in the following directory:

   ```
   ...\Middleware\user_projects\domains\base_domain\bin
   ```

   For example, the file might be in the following location:

   ```
   C:\Oracle\Middleware\user_projects\domains\base_domain\bin\startWebLogic.cmd
   ```

4. Save a backup of the startWebLogic.cmd file.
5. Open the startWebLogic.cmd file in a text editor. To append the -javaagent argument to the Java command line, enter the following text below the comments section:

   ```
   set JAVA_OPTIONS=% JAVA_OPTIONS % -javaagent:..\lib
   \GenericJDBC.jar=host:<ddm_host>,port:<ddm_generic_service_port>
   ```

6. To set the classpath, add the following line after the classpath lines, but before the echo statements:

```
set CLASSPATH=%CLASSPATH%;..\lib\GenericJDBC.jar; ..\lib\informatica-jdbc-
db2-5.1.2.HF1.jar;<additional drivers>
```

7. Save startWebLogic.cmd.

Configure WebLogic for Linux

Complete the following steps to configure WebLogic for Linux:

1. Copy the GenericJDBC.jar file. You can find the file in the following location:

```
<Dynamic Data Masking installation>/Wrappers/jdbc/GenericJDBC.jar
```

2. Save the GenericJDBC.jar file in the following location:

```
.../Middleware/user_projects/domains/base_domain/lib
```

   For example, you might save the file in the following location:

```
C:/Oracle\Middleware/user_projects/domains/base_domain/lib
```

3. Find the startWebLogic.sh file. You can find the file in the following directory:

```
...\Middleware\user_projects\domains\base_domain\bin
```

   For example, the file might be in the following location:

```
C:\Oracle\Middleware\user_projects\domains\base_domain\bin\startWebLogic.cmd
```

4. Save a backup of the startWebLogic.cmd file.
5. Open the startWebLogic.cmd file in a text editor. To append the -javaagent argument to the Java command line, enter the following text below the comments section:

```
set JAVA_OPTIONS=% JAVA_OPTIONS % -javaagent:..\lib
\GenericJDBC.jar=host:<ddm_host>,port:<ddm_generic_service_port>
```

6. To set the classpath, add the following line after the classpath lines, but before the echo statements:

```
set CLASSPATH=%CLASSPATH%;..\lib\GenericJDBC.jar; ..\lib\informatica-jdbc-
db2-5.1.2.HF1.jar;<additional drivers>
```

7. Save startWebLogic.cmd.

CHAPTER 5

ODBC Client Configuration

This chapter includes the following topic:

ODBC Client Configuration Overview

Before you use the Dynamic Data Masking Generic ODBC approach to mask data for a database that uses an ODBC connection, you must configure the client machine.

The Dynamic Data Masking installation includes ODBC DLL files for 64-bit and 32-bit Windows. You must save the files in the Windows system directory and create Dynamic Data Masking host and port environment variables. Optionally, you can create a Data Source in the Windows ODBC Administrator to test the configuration.

1. Verify the requirements for ODBC client configuration.
2. Grant the required permissions to the user that performs the setup for the Windows Driver Manager files.
3. Set up the Dynamic Data Masking Driver Manager proxies.
4. Create the environment variables.
5. Optionally, create the Windows Data Source to test the client configuration.

Step 1. Verify the Requirements

Verify the following requirements before you configure the client:

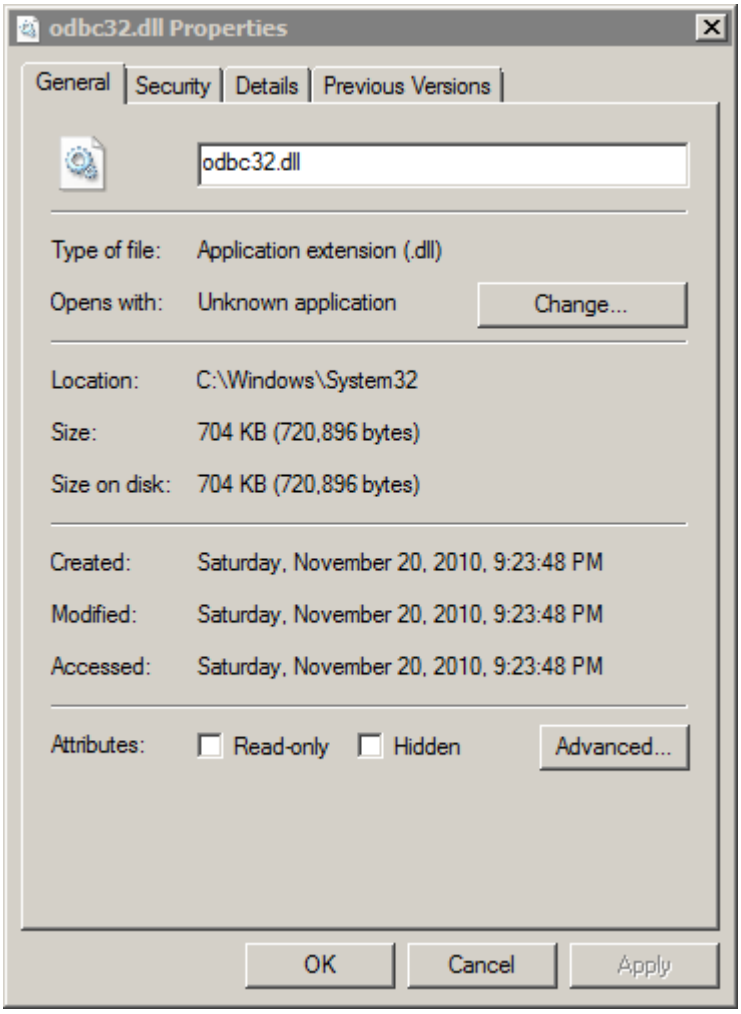
- The client must have a Windows 7 or Windows Server 2008 operating system.
- The Windows user that performs the setup must have administrator privileges to modify the ODBC DLLs or the user must have the rights to grant the required privileges or ownership.
- The Windows user that performs the setup must have the required permissions to rename the Windows ODBC Driver Manager (odbc32.dll).

Step 2. Grant File Permissions

Before you set up the Driver Manager proxies, you must identify the Driver Manager proxy that you want to install on the target application architecture and the ODBC 32-bit or 64-bit usage. Then you must change the owner of the Windows Driver Manager files and grant the required permissions.

1. Find the ODBC Driver Manager DLL file on the Windows machine based on the architecture of the target application. Complete the remaining steps for each of the required Driver Manager files on the machine.
   a. On a 64-bit Windows machine, you can find the following Windows Driver Manager files:
      - 64-bit Driver Manager: `<Windows installation>\System32\odbc32.dll`
      - 32-bit Driver Manager: `<Windows installation>\SysWOW64\odbc32.dll`
   b. On a 32-bit Windows machine, you can find the following Windows Driver Manager file:
      - 32-bit Driver Manager: `<Windows installation>\System32\odbc32.dll`
2. In Windows Explorer, right-click the file and click **Properties**.

   The **odbc32.dll Properties** window opens. The following image shows the window:



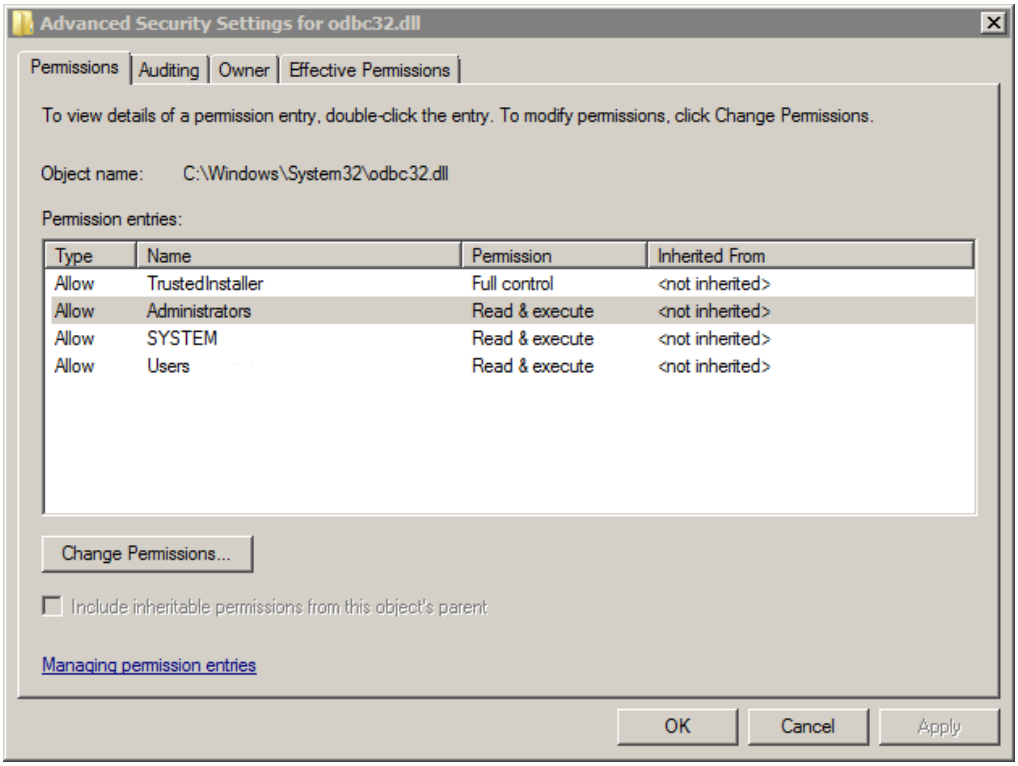
3. Select the **Security** tab and click **Advanced**.

   The **Advanced Security Settings for odbc32.dll** window opens.

4. Select the **Owner** tab and click **Edit**.
5. In the window that opens, select the current user and click **Apply**.
6. Click **OK** to close the window.
7. Click **OK** to close the **Advanced Security Settings for odbc32.dll** window.
8. In the **odbc32.dll Properties** window, click **OK** to close the window. You must close the properties window before you make additional changes to the file.
9. Reopen the **odbc32.dll Properties** window. Right-click the file and click **Properties**.
10. Select the **Security** tab and click **Advanced**.

    The **Advanced Security Settings for odbc32.dll** window opens.
11. In the **Permissions** tab, select the Administrators group and click **Change Permissions**.

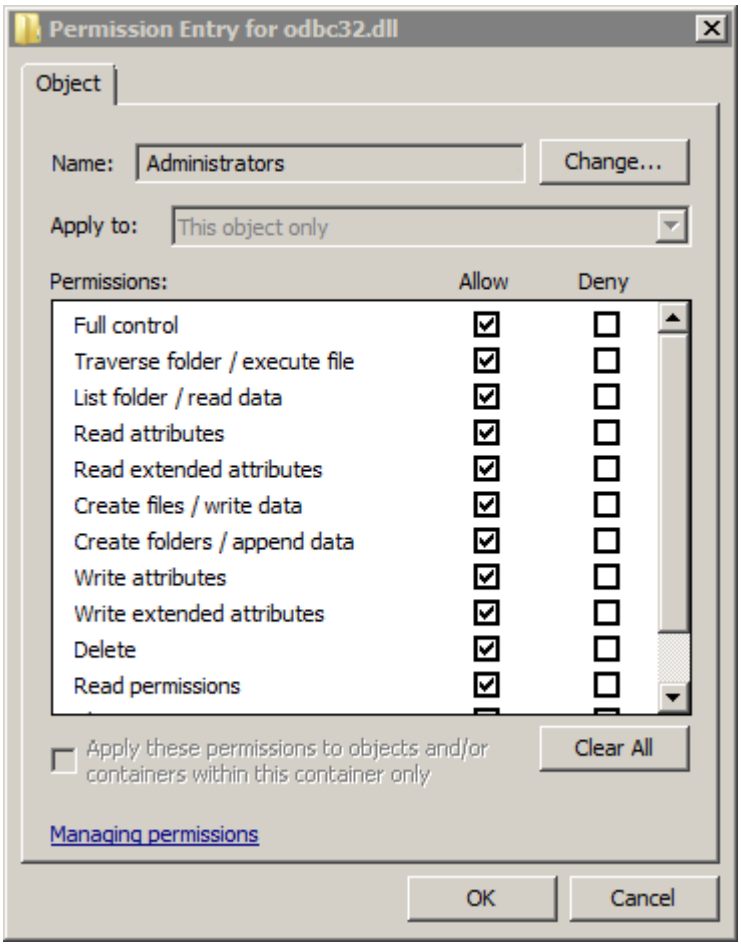
    The following image shows the **Permissions** tab:



12. In the window that opens, select the Administrators group again and click **Edit**.

    The **Permission Entry for odbc32.dll** window opens.
13. Click the **Allow** box in the **Full control** row to grant full permissions to the user.

    The following image shows the window with the **Full control** box selected:



14. Click **OK** to close the window.
15. Click **OK** to close the previous window. A dialog box asks if you want to continue. Click **Yes** to close the window.
16. Click **OK** to close the **Advanced Security Settings for odbc32.dll** window.
17. Click **OK** again to close the **odbc32.dll Properties** window.

    The permissions changes are saved.

Step 3. Set Up the Driver Manager Proxies

Save the Dynamic Data Masking Generic ODBC DLL files in the Windows system directory.

Complete the following steps for each Windows Driver Manager file that you edited in the previous step.

1. Rename the Windows Driver Manager file to odbc32.dll. Rename `odbc32.dll` to `odbc32o.dll`.

2. Find the Dynamic Data Masking Driver Manager proxy based on the architecture of the target client application.
   a. Find the following Dynamic Data Masking Generic ODBC DLL file for a 64-bit application:
      - `<Dynamic Data Masking installation>\Wrappers\odbc\Windows\odbc64\GenericOdbc64.dll`
   b. Find the following Dynamic Data Masking Generic ODBC DLL file for a 32-bit application:
      - `<Dynamic Data Masking installation>\Wrappers\odbc\Windows\odbc32\GenericOdbc32.dll`
3. Copy the Dynamic Data Masking Driver Manger file to the Windows system directory.
   a. Copy `GenericOdbc64.dll` to the following directory:

      `<Windows installation>\System32`
   b. Copy `GenericOdbc32.dll` to the following directory:
      - On a 64-bit machine: `<Windows installation>\SysWOW64`
      - On a 32-bit machine: `<Windows installation>\System32`
4. Rename the Dynamic Data Masking Generic ODBC DLL file to `odbc32.dll` for the 64-bit and 32-bit version.
   a. Rename `GenericOdbc64.dll` to `odbc32.dll`.
   b. Rename `GenericOdbc32.dll` to `odbc32.dll`.

Step 4. Create the Environment Variables

Create the Dynamic Data Masking host environment variables.

1. Open the Windows Start menu, right click **Computer**, and click **Properties**.

   The **Control Panel** opens.
2. On the right side of the Control Panel, click **Advanced system settings**.

   The **System Properties** window opens.
3. In the **System Properties** window, click the **Advanced** tab and click **Environment Variables** at the bottom of the window.

   The **Environment Variables** window opens.
4. Under the **System variables** box, click **New**.

   The **New System Variable** window opens.
5. In the **New System Variable** window, enter the following properties:

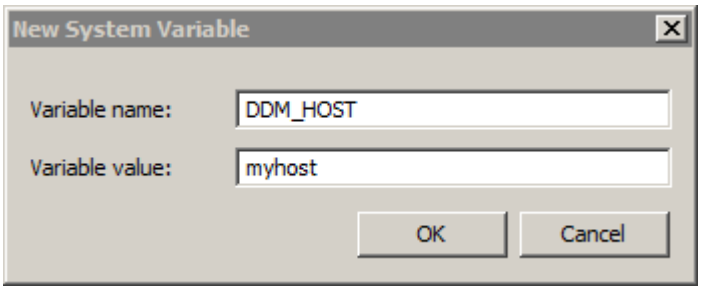
   Variable Name

   DDM_HOST

   Variable Value

   Dynamic Data Masking Server host name.

   The following image shows the **New System Variable** window:



6. Click **OK** to close the window.

   The DDM_HOST environment variable appears in the list of system variables.
7. In the System variables box, click New to add another variable.

   The **New System Variable** window opens.
8. In the **New System Variable** window, enter the following properties:

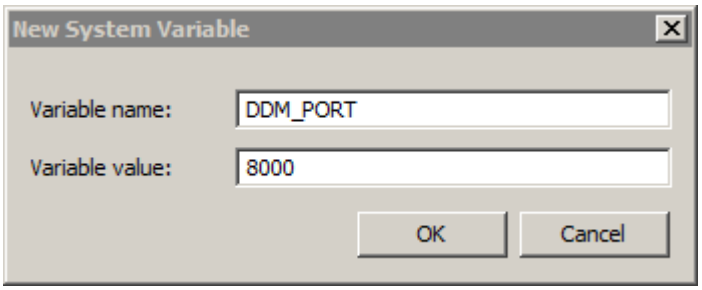
   Variable Name

   DDM_PORT

   Variable Value

   Dynamic Data Masking Generic ODBC service port.

   The following image shows the **New System Variable** window:



9. Click **OK** to close the window.

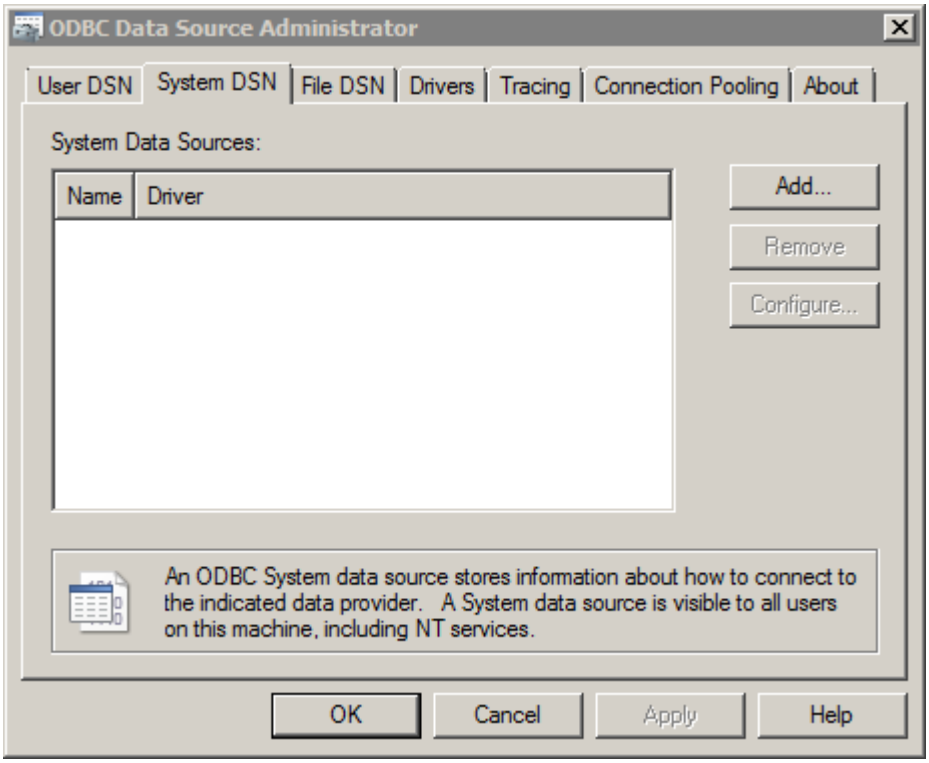
   The DDM_PORT environment variable appears in the list of system variables.
10. Click **OK** to close the **Environment Variables** window.
11. Click **OK** to close the **System Properties** window.

Step 5. Create a Windows Data Source (Test the Setup)

To test the client configuration, or if a System Data Source is not available on the client machine, you can create a System Data Source with the Windows ODBC Data Source Administrator, based on the architecture of the machine.

1. Open the Windows ODBC Data Source Administrator.
   a. On a 64-bit Windows machine, you can find the ODBC Data Source Administrator in the following locations:
      - 64-bit ODBC Administrator: `<Windows installation>\System32\odbcad32.exe`
      - 32-bit ODBC Administrator: `<Windows installation>\SysWOW64\odbcad32.exe`
   b. On a 32-bit Windows machine, you can find the ODBC Data Source Administrator in the following location:
      - 32-bit ODBC Administrator: `<Windows installation>\System32\odbcad32.exe`
2. To view the System Data Sources, click the **System DSN** tab at the top of the window.

   The following image shows the **System DSN** properties:



3. Click **Add**.

   The **Create New Data Source** window opens.
4. In the **Create New Data Source** window, click the name of the ODBC driver that you want to create a DSN for, based on the database that you want to connect to.
5. Click **Finish**.

   The ODBC setup window opens.

6. Provide the DSN details that the ODBC setup window requires and click **Test Connection**. When the successful connection dialog box appears, click **OK** to close the dialog box.
7. Click **OK** to close the ODBC setup window.

   The DSN appears in the **System DSN** tab of the **ODBC Data Source Administrator** window.
8. Note the System Data Source Name property value, which the client application requires to connect to the target database.
9. Click **OK** to close the ODBC Data Source Administrator.

CHAPTER 6

Access Control

This chapter includes the following topics:

Access Control Overview

Use Dynamic Data Masking access control to define the operations that a user can perform on Management Console tree nodes and Dynamic Data Masking configuration.

You can set permissions on domain, database, and security rule set nodes to define the users that can edit the nodes.

A Dynamic Data Masking user can be a privileged user or a non-privileged user. The type of user and type of permissions the user has on a node in the Management Console tree determines whether the user can view or make changes to the node.

Privileged User

A privileged user is a user that is connected to the the Dynamic Data Masking Server as the administrator or an LDAP user that belongs to the Dynamic Data Masking administration group.

Privileged users have full access control on the Management Console tree nodes. Privileged users can set privileges and perform any operation on Management Console tree nodes.

Non-Privileged User

A non-privileged user is a user that does not belong to the Dynamic Data Masking administration group.

In the Management Console tree, domain, database, and security rule set nodes have authorization properties. Authorization properties define which operations a non-privileged user can perform on Management Console tree nodes.

Non-privileged users cannot edit the Management Console Dynamic Data Masking Server node or the Server node children, such as service nodes, logger nodes, and appender nodes. A non-privileged user can have ownership, read, or read and write privileges on domain, database, and security rule set nodes in the Management Console tree.

The following table describes the authorizations a non-privileged user can have on a database, domain, or security rule set node in the Management Console tree:

Authorization	Description
Ownership	The LDAP user or group owns the node. A node owner has full access control to the node. A node owner can perform the following operations on the node: - Set Authorizations - Read operations - Write operations
Read	The LDAP user or group has read privileges on the node. A user with read privileges can perform the following operations on the node: - View the node details - View the children of the node Read authorizations are required on the source node for the copy node operation.
Read and Write	The LDAP user or group has read and write privileges on the node. A user with read and write privileges can perform the following operations: - View the node details - View the children of the node - The read privilege is required on the source node for the copy node operation. - Add - Edit database and security rule set node details - Copy (destination node) - Move (source and destination node) - Remove (parent and child nodes) - Edit domain, database, and security set names (parent and child nodes)

Authorization Properties of a Non-Privileged User

A non-privileged user must have authorizations to perform operations on Management Console tree nodes. Because the Management Console tree is a hierarchical tree, for some operations the user must have authorizations on multiple nodes.

The following table describes the operations a non-privileged user can perform on a database, domain, or security rule set node and the authorizations the user must have:

Operation	Authorizations
Add node	Ownership or read and write authorizations on the node. The user that creates the node is the owner of the node.
Copy node	Ownership or read authorizations on the source node and the descendants of the source node. Ownership or read and write privileges on the destination node. The user that creates the node is the owner of the node.
View authorizations	Ownership or read authorizations on the node.
Expand node	Ownership or read authorizations on the node.
View database details	Ownership or read authorizations on the node.
View security rule set details	Ownership or read authorizations on the node.
Move node	Ownership or read authorizations on the source node and the descendants of the source node. Ownership or read and write privileges on the destination node.
Remove node	Ownership or read and write authorizations on the parent node. Ownership or read and write authorizations on the child node.
Edit authorizations	Ownership.
Edit database details	Ownership or read and write authorizations on the node.
Edit security rule details	Ownership or read and write authorizations on the node.
Edit domain name	Ownership or read and write authorizations on the parent node. Ownership or read and write authorizations on the child node.
Edit security rule set name	Ownership or read and write authorizations on the parent node. Ownership or read and write authorizations on the child node.
Edit database name	Ownership or read and write authorizations on the parent node. Ownership or read and write authorizations on the child node.

Authorization Properties of a Newly Created Node

When you create a node in the Management Console tree, the node has default authorization properties.

You can add or copy a Management Console tree node to create a node.

The following table describes the default authorization properties:

Property	Default
Owner	User that creates the node. If the user is logged into Dynamic Data Masking as the administrator, the owner property is empty. The owner property is set to copied nodes and copied child nodes.
Read Privileges	Empty for an add operation. The read privileges property does not change for copied nodes and copied child nodes.
Read and Write Privileges	Empty for an add operation. The read and write privileges property does not change for copied nodes and copied child nodes.

Authorization Properties of a Moved Node

The authorization properties of moved Management Console tree nodes and child nodes do not change when you use the move operation.

Configuring Access Control

Configure access control on a node in the Management Console tree to allow users and LDAP groups to access the node.

1. In the Management Console, select a domain, database, or security rule set node.
2. Click **Tree** > **Authorization**.

   The **Authorize User or Group** window appears.
3. Define a node owner in the owner field and define read and write privileges for users and LDAP groups in the **Authorize User or Group** window.
4. Click **Ok**.

CHAPTER 7

Logs

This chapter includes the following topics:

Logs Overview

A log file maintains a history of events. Dynamic Data Masking creates log files and contains system loggers that record Dynamic Data Masking Server, service, and rule events. You can create custom loggers to log information that you specify in a security rule. Set log levels to determine which loggers send log information.

The Management Console Tree contains logger and appender nodes that create log files. The system loggers create the audit trail, rule, and server logs. You can add custom loggers to the Management Console tree that you use in security rules to specify events that you want to log. You can add appender nodes under loggers to specify how and where to log the event information. Set the Log Level property of the Dynamic Data Masking Server to specify the severity level of the events that you want to log.

You can use the log files to identify a problem with the Dynamic Data Masking Server, and to monitor and troubleshoot problems a Dynamic Data Masking service.

In addition to the system loggers, Dynamic Data Masking creates the following log files:

<year>_<month>.at

Detailed audit trail file. Contains detailed information about changes made within the Management Console. Dynamic Data Masking uses the year and the month that it creates the file to name the file.

DDMError.txt

Logs standard operating system process errors when they occur.

DDMOutput.txt

Logs standard operating system process output when they occur.

You can find log files in the following location: `<Dynamic Data Masking installation>/log`

Note: If you choose the Dynamic Data Masking Management Console installation without the Dynamic Data Masking Server, the installation does not create a log directory.

Audit Trail and Detailed Audit Trail

The Dynamic Data Masking general audit trail and detailed audit trail log files contain information that you use to verify whether a user made unauthorized modifications to the Dynamic Data Masking configuration.

The AuditTrail.log file contains general audit information about changes in the Management Console. The AT appender in the auditTrail logger creates the AuditTrail.log file. You can add appenders to the auditTrail logger to create additional audit trail output formats.

The detailed audit file contains comprehensive audit information that is not in the AuditTrail.log file. Dynamic Data Masking names the detailed audit file according to the year and month that it creates the file. For example, if Dynamic Data Masking creates a detailed audit file in April 2013, it names the file 2013_04.at.

The detailed audit file contains the following information on modifications to the Dynamic Data Masking configuration properties:

Property	Modification
Database	Database node changes. You can view the following actions: - Add - Remove - Copy - Move - Edit - Change database name
Security rule set	Security rule set changes. You can view the following actions: - Add - Remove - Copy - Move - Edit - Change security rule name
Service	Dynamic Data Masking service changes. You can view the following actions: - Add - Remove - Edit
Authorization	Authorization changes. You can view the following actions: - Set authorizations
Connection rule	Connection rule changes. You can view the following actions: - Edit
Server	Dynamic Data Masking Server changes. You can view the following actions: - Edit

The entries in the detailed audit file contain the following tags:

Tag	Description
operation	Audit trail operations. You can view the following operations: - Add - Change - Remove
date	Date and time of the modification.
clientIP	Client IP address or host name.
userName	Client user name. The entry does not contain the userName tag if the user is admin.
status	Status of the operation. You can view the following statuses: - Success - Failed
path	Element path in the Management Console tree.
authorizations	Element authorizations such as owner, read privileges, and read and write privileges. If the owner is admin, the authorizations tag is empty.
content	Content of the element.

Sample AuditTrail.log File

The following excerpt is from the general audit trail log:

```
05/14 19:54:43,199 INFO  - [127.0.0.1] [admin] LOGIN: Success!
05/14 20:00:17,093 INFO  - [127.0.0.1] [admin] REMOVE NODE (DatabaseNode) Oracle11g
(10.10.10.10): Success!
05/14 20:00:37,620 INFO  - [127.0.0.1] [admin] ADD NODE (DomainNode) Domain1: Success!
05/14 20:00:40,901 INFO  - [127.0.0.1] [admin] MOVE NODE (ServerNode) myServer: Success!
05/14 20:00:46,175 INFO  - [127.0.0.1] [admin] REMOVE NODE (DomainNode) Domain1: Cannot
remove node Domain1. DDM Server myServer is in branch.
05/14 20:00:49,441 INFO  - [127.0.0.1] [admin] MOVE NODE (ServerNode) myServer: Success!
05/14 20:00:58,838 INFO  - [127.0.0.1] [admin] EDIT NODE (DomainNode) domain2: Success!
05/14 20:01:06,603 INFO  - [127.0.0.1] [admin] EDIT NODE (RuleSetNode) MaskingRuleSet:
Success!
05/14 20:01:11,405 INFO  - [127.0.0.1] [admin] EDIT NODE (RuleSetNode) MaskingRuleSet:
Success!
05/14 20:01:21,679 INFO  - [127.0.0.1] [admin] SET SWITCHING RULES: ddmForOracle:
Success!
05/14 20:01:25,356 INFO  - [127.0.0.1] [admin] SET SWITCHING RULES: ddmForOracle:
Success!
05/14 20:01:36,912 INFO  - [127.0.0.1] [admin] LOGOUT
05/14 20:01:37,916 INFO  - [127.0.0.1] [admin] LOGIN: Authentication failed
05/14 20:01:43,440 INFO  - [127.0.0.1] [admin] LOGIN: Success!
05/14 20:02:01,397 INFO  - [127.0.0.1] [admin] LOGOUT05/14 20:20:34,104 INFO  -
[127.0.0.1] [admin] LOGIN: Success!
05/14 20:20:54,006 INFO  - [127.0.0.1] [admin] MOVE NODE (DomainNode) domain2: Success!
05/14 20:22:37,829 INFO  - [127.0.0.1] [admin] LOGIN: Success!
05/14 20:22:37,831 INFO  - [127.0.0.1] [admin] LOGOUT
05/14 20:23:09,070 INFO  - [127.0.0.1] [admin] LOGIN: Success!
05/14 20:25:51,918 INFO  - [127.0.0.1] [admin] LOGOUT
```

Sample Detailed Audit File

The following audit file is an example of the detailed audit file:

```
  <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- <XML type="database">
- <content type="ATContent">
- <newContent>
  <cpuCount>0</cpuCount>
- <services type="ArrayList">
- <entry>
  <serviceName>1</serviceName>
  </entry>
  </services>
  <oracleVersion />
  <dbaPassword>GNKEJHAPBJBBMECC</dbaPassword>
  <dbaUser>1</dbaUser>
  <name>1</name>
- <instances type="ArrayList">
- <entry class="com.activebase.db.oracle.OracleInstance" type="Configurable">
  <cpuCount>0</cpuCount>
  <oracleVersion />
  <dbaPassword>GNKEJHAPBJBBMECC</dbaPassword>
  <instanceName>1</instanceName>
  <dbaUser>1</dbaUser>
  <listenerAddress>1</listenerAddress>
  <hostName />
  <listenerPort>1</listenerPort>
  <infoTimestamp>0</infoTimestamp>
  </entry>
  </instances>
  <infoTimestamp>0</infoTimestamp>
  </newContent>
  </content>
  <operation>add</operation>
- <authorizations type="ATAuthorizations">
  <newAuthorizations class="com.activebase.configuration.Authorizations"
type="Configurable" />
  </authorizations>
  <status>success</status>
- <path type="ATPath">
  <newPath>Site/1</newPath>
  </path>
  <date>01/08/2013 19:26:57,429</date>
  <clientIP>127.0.0.1</clientIP>
  </XML>
```

Loggers

A logger is a Management Console tree node that uses Apache log4j to create a log of events.

You can add logger nodes in the Management Console tree under the Loggers node.

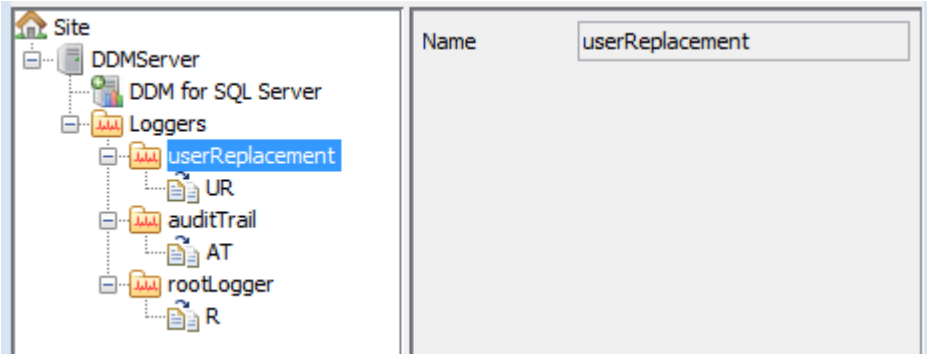
Use loggers to specify the events that you want to log and use appenders to define how to log the event. You can use pre-defined system loggers to log Dynamic Data Masking Server, service, and rule events. You can create custom loggers to log security rule events that you specify with the Log Message rule action.

Logger nodes can have multiple appender child nodes. When you use the logger in a security rule, the logger logs the event in each format specified by the child appender nodes.

Dynamic Data Masking contains pre-defined system loggers and appenders that log Dynamic Data Masking service events and rule events. You cannot edit or delete the system loggers.

The Loggers node is a child of the Dynamic Data Masking Server node in the Management Console tree. Because it is a child of the Dynamic Data Masking Server node, only Dynamic Data Masking administrators can edit and create child nodes of the Loggers node. Non-privileged users cannot edit or move logger and appender nodes and an administrator cannot delegate permissions to a non-privileged user.

The following image shows the Loggers node and the child system logger nodes:



System Loggers

A system logger is a pre-defined logger node in the Management Console tree that logs Dynamic Data Masking Server, service, and rule events.

The Management Console tree contains userReplacement, auditTrail, and rootLogger system loggers. The system loggers use Rolling File appenders to create the audit trail, rule, and server logs. You cannot delete or move the system loggers or the appenders. You can edit the Max File Size and Max Backups properties of the system logger appenders, but you cannot edit the Type, Name, and File properties. If you edit a system logger appender, Dynamic Data Masking immediately reconfigures config.properties and saves the file.

You can add appenders to the system loggers to log the same information in different formats.

The system loggers create the following log files:

auditTrail.log

Logs changes made within the Management Console. The AT appender of the auditTrail logger creates the auditTrail.log file.

rule.log

Logs rules that the Rule Engine applies to incoming requests. In the Management Console, you can use the Log When Rule is Applied box in the **Edit Rule** window to specify whether an occurrence of the rule is logged. The UR appender of the userReplacement logger creates the rule.log file.

If multiple rule log files exist, Dynamic Data Masking appends each file name with a version number, such as `rule.log1`. Dynamic Data Masking stores 10 rule log files by default. Rule logs update cyclically and restart on `rule.log1` when the logs are full.

By default, each rule log file stores up to 20 MB of data for a total of 200 MB. You can configure file size and the maximum number of files in the UR appender.

Note: In high transaction volume applications, specify additional rule logs carefully due to the increased overhead.

You can use the Log Loader utility to load rule.log data into an Oracle, DB2, Informix, or Microsoft SQL Server database. See *Informatica Dynamic Data Masking Log Loader* for information on the Log Loader utility.

server.log

Logs server records, events, and error messages for internal troubleshooting of the Dynamic Data Masking Server operations. The R appender of the rootLogger logger creates the server.log file.

If multiple server log files exist, Dynamic Data Masking appends each file name with a version number, such as `server.log1`. Dynamic Data Masking stores up to 10 server log files at a time. Server logs update cyclically and restart on `server.log1` when the logs are full.

By default, each rule log file stores up to 20 MB of data for a total of 200 MB. You can configure file size and the maximum number of files in the UR appender.

Sample Rule.log File

The following excerpt is from the rule log:

```
05/30 16:55:39,240 [MASK@ERP-1] INFO  - Blocking Rule: Identify
Blocked Statement (user message: This request has been blocked.)
select * from customer
Done by ClientInfo:[User=Admin, Host=__jdbc__, application=JDBC Thin Client] -
SessionID=74,2834 - SYSTEM - Instance 1

05/30 16:57:29,156 [sapiens@ERP-1] INFO  - None Rule: auditlog
BEGIN DBMS_OUTPUT.DISABLE; END;
Done by ClientInfo:[User=Admin, Host=ADMIN-THINK, application=C:\app\Admin\product
\11.2.0\dbhome_2\bin\sqlplus.exe] - SessionID=73,763 - system - Instance 1
```

Sample Server.log File

The following excerpt is from the server log:

```
05/23 13:16:00,075 [pool-1-thread-1] INFO  - Service started.
05/23 13:16:00,077 [main] INFO  - Service DDM for DB2 started
05/23 13:16:00,077 [main] WARN  - DDM for Oracle.configure: Invalid address provided:
null:0
05/23 13:16:00,079 [main] INFO  - Service DDM for Oracle started
05/23 13:16:00,116 [main] INFO  - Server started.
05/23 13:16:00,198 [Thread-3] INFO  - Service DDM for SQL Server started
05/23 18:36:43,989 [Thread-2] WARN  - ProcessService: ProcessService: restarting process.
05/23 18:36:44,005 [Thread-4] INFO  - Service DDM for SQL Server started
```

Custom Loggers

A custom logger is a logger that you create to use in a security rule to define events that you want to log.

Loggers have a Name property that you define when you create the logger. Logger names must be unique. When you create a security rule with the Log Message rule action, you specify the name of the logger in the rule and define the log level in the rule. The logger creates logs based on the appender child nodes of the logger. The log level that you define in the Send As parameter of the security rule must be equal to or higher than the log level that you define in the Dynamic Data Masking Server node. If the log level of the rule is a lower severity than the Dynamic Data Masking Server Log Level parameter, the logger will not log the event.

After you create a logger and add an appender, you must use the Log Message rule action in a security rule to define the events that you want the logger to log. If you do not use the logger in a security rule, it does not log events.

Because logger nodes are child nodes of the Dynamic Data Masking Server node, only a Dynamic Data Masking administrator can create or edit a logger node. An administrator cannot delegate privileges to a non-privileged user.

Note: Do not use system loggers with the Log Message rule action because you might not be able to perform log analysis on the logs if they contain information from security rules.

Creating a Custom Logger

To define the events that you want to log, create a custom logger that you use in a security rule .

1. In the Management Console tree, select the Loggers node and click **Tree** > **Add Logger**.

   The **Add Logger** window appears.
2. Enter the name of the logger. The logger name can contain alphabetic characters, numbers, and underscores (_), and must not exceed 60 characters.
3. Click **OK**.

   The logger appears in the Management Console tree.

After you create a logger, you must add appenders to the logger to specify the log output format. You must use the logger in a security rule with the Log Message rule action to define the events that you want to log.

Loggers Example

Your organization uses syslog to integrate log data from multiple types of systems. You want to add Dynamic Data Masking logs to the syslog repository.

You add appenders to the userReplacement, auditTrail, and rootLogger system loggers. The appenders use the syslog appender class. When Dynamic Data Masking writes to the auditTrail.log, rule.log, and server.log files, it also creates a syslog output.

You create custom loggers and you add syslog appenders to the loggers. You use the loggers in security rules based on the events that you want to log. Dynamic Data Masking sends the event data to the syslog repository.

Appenders

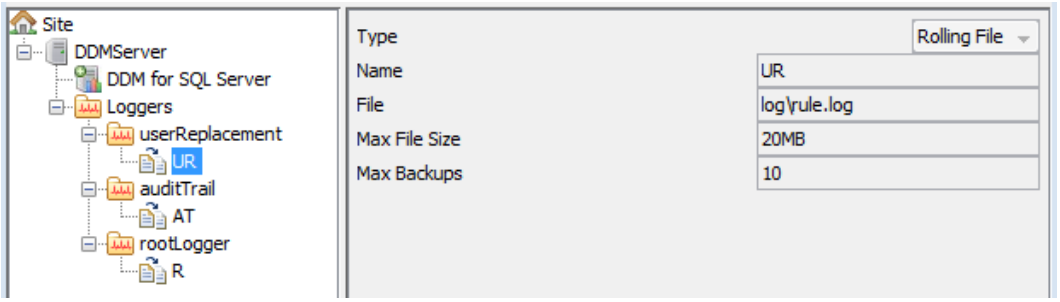
An appender is a node in the Management Console tree that uses log4j classes to define the output format of log information.

You can create appenders to log information in a format that is useful to your organization. You can use the built-in Rolling File, Syslog, SMTP, and SNMP appenders, or create a custom appender to store log information in any format. A logger can have multiple appenders.

Appenders are child nodes of logger nodes in the Management Console tree. Only administrators can create and edit appender nodes. An administrator cannot delegate appender permissions to a non-privileged user.

The Dynamic Data Masking system loggers use Rolling File appenders to create the rule.log, auditTrail.log, and server.log files. You cannot delete the system logger appenders. You can edit the system logger appender Max File Size and Max Backups properties, but you cannot edit the Type, Name, and File properties. You can add an appender to the system loggers to create an additional system log output.

The following image shows the system logger appenders in the Management Console tree and the UR appender properties:



Rolling File Appender

Create a Rolling File appender to log information to a text file.

Rolling File appenders create plain text output files. You can use any plain text extension, such as .log or .txt.

The File property contains the file path and name of the log file. If you do not specify a complete file path, the path originates in the Dynamic Data Masking installation directory. For example, the rule.log file has the following File property:

```
log\rule.log
```

The Max File Size and Max Backups properties prevent the log files from becoming too large. When the log reaches the Max File Size, the logger creates a new log file. When the number of files exceeds the Max Backups number, the logger overwrites the first log file that it created. You can specify Max File Size and Max Backups in megabytes or gigabytes by entering the number followed by MB or GB. Do not insert a space between the value and the unit of measure.

Rolling File Appender Properties

The following table describes the Rolling File appender properties:

Property	Description
Name	The name of the appender. The appender name can contain alphabetic characters, numbers, and underscores (_), and must not exceed 60 characters.
File	The file path and name of the log file. If you do not specify a complete file path, the path originates in the Dynamic Data Masking installation directory. The file path and name cannot exceed 60 characters.
Max File Size	The maximum size that the output file reaches before the appender creates a new file. Default is 10MB. **Note:** Do not enter a space between the value and the unit of measure.
Max Backups	The number of backup files the appender creates before it overwrites the oldest file. Max Backups must be a positive integer. Default is 20.

Syslog Appender

Create a Syslog appender to log information to a syslog repository.

If your organization uses a syslog system to store log information, you can use the Syslog appender to log events. You can add Syslog appenders to the system loggers to log standard Dynamic Data Masking output files to the syslog repository in addition to the Dynamic Data Masking log directory.

Syslog Appender Properties

The following table describes the Syslog appender properties:

Property	Description
Name	The name of the appender. The appender name can contain alphabetic characters, numbers, and underscores (_), and must not exceed 60 characters.
Syslog Host	The host name or IP address of the Syslog server.
Facility	The Syslog facility. A system administrator can set Facility to one of the following strings: KERN, USER, MAIL, DAEMON, AUTH, SYSLOG, LPR, NEWS, UUCP, CRON, AUTHPRIV, FTP, LOCAL0, LOCAL1, LOCAL2, LOCAL3, LOCAL4, LOCAL5, LOCAL6, LOCAL7. Default is USER.

SMTP Appender

Create an SMTP appender to send log information as an email.

When you create an SMTP appender, the logger sends log events as an email. You can specify multiple email addresses that you want to receive the email. For example, you can use the SMTP appender to send an email when an error or fatal error occurs.

Before you use the SMTP appender, you must put the mail.jar file into the `<Dynamic Data Masking installation>/lib/ext` directory. On Windows, you must remove and restart the Dynamic Data Masking Server to update the Server configuration in the Windows Registry. On Unix, you must restart the Dynamic Data Masking Server.

SMTP Appender Properties

The following table describes the SMTP appender properties:

Property	Description
Name	The name of the appender. The appender name can contain alphabetic characters, numbers, and underscores (_), and must not exceed 60 characters.
From	The email address of the sender.
To	List of recipient email addresses, separated by commas.
Subject	The subject line of the email.
SMTP Host	The SMTP relay mail server to use to send the email.

Property	Description
SMTP Username	The username to authenticate the SMTP server.
SMTP Password	The password to authenticate the SMTP server.
Debug	Enable Debug to analyze the standard system output if an error occurs, such as an SMTP login failure.

Configuring the SMTP Appender

Before you use the SMTP appender, you must place the SMTP jar file into the Dynamic Data Masking directory. On Windows, remove and restart the Dynamic Data Masking Server. On Unix, restart the Dynamic Data Masking Server.

To configure the SMTP appender, you need the following file:

- mail.jar version 1.4.2

1. Place the mail.jar file into the following directory:

   ```
   <Dynamic Data Masking installation>/lib/ext
   ```

2. Close the Management Console.
3. Open a Server Control window.
   - If the Dynamic Data Masking Server runs on Windows, enter the following commands:

     ```
     - server stop
     - server remove
     - server start
     ```

   - If the Dynamic Data Masking Server runs on Unix, enter the following command:

     ```
     - server restart
     ```

4. Open the Management Console and create the SMTP appender.

Note: If you add, remove, or update the files in the `<Dynamic Data Masking installation>/lib/ext` directory, you must configure the SMTP appender again.

SNMP Appender

Create an SNMP appender to send log information by using SNMP protocol.

Simple Network Management Protocol (SNMP) is a standard protocol for managing devices on IP networks. Devices that typically support SNMP are routers, switches, servers, workstations, printers, and modem racks.

When you create an SNMP appender, you define the Generic Trap Type as a numeric value. You can find information on trap types at http://publib.boulder.ibm.com/infocenter/zvm/v5r4/index.jsp?topic=/com.ibm.zvm.v54.kijl0/trp.htm.

Before you use the SNMP appender, you must put the SNMPTrapAppender and OpenNMS jar files into the `<Dynamic Data Masking installation>/lib/ext` directory. On Windows, you must remove and restart the Dynamic Data Masking Server to update the Server configuration in the Windows Registry. On Unix, you must restart the Dynamic Data Masking Server.

SNMP Appender Properties

The following table describes the SNMP appender properties:

Property	Description
Name	The name of the appender. The appender name can contain alphabetic characters, numbers, and underscores (_), and must not exceed 60 characters.
Server Host	The host name or IP address of the SNMP server.
Server Trap Listen Port	The SNMP server port. Default is 161.
Enterprise OID	The object ID of the organization that sends the trap message. Set this parameter to any value to identify the message.
Local IP Address	The IP address of the local SNMP embedded agent. Default is 127.0.0.1.
Local Trap Send Port	The port of the local SNMP embedded agent.
Generic Trap Type	A number value that specifies the trap type. Set the Generic Trap Type to one of the following values: - 0. coldStart - 1. warmStart - 2. linkDown - 3. linkUp - 4. authenticationFailure - 5. egpNeighborLoss - 6. enterpriseSpecific
Specific Trap Type	The trap description.
SNMP Community	The name of the SNMP community. Default is public.
Forward Stack Trace with Trap	Specifies whether to include the stack trace as part of the log message.
Application Trap OID	The object ID of the application that sends the trap message. Enter the name of the application object.

Configuring the SNMP Appender

Before you use the SNMP appender, you must place the SNMP jar files into the Dynamic Data Masking directory. On Windows, remove and restart the Dynamic Data Masking Server. On Unix, restart the Dynamic Data Masking Server.

To configure the SNMP appender, you need the following files:

- SNMPTrapAppender_1_2_9.jar
- opennms-joesnmp-20031201-173122.jar

1. Place the SNMPTrapAppender and the OpenNMS jar file into the following directory:

   ```
   <Dynamic Data Masking installation>/lib/ext
   ```

2. Close the Management Console.

3. Open a Server Control window.
   - If the Dynamic Data Masking Server runs on Windows, enter the following commands:

     `- server stop`

     `- server remove`

     `- server start`
   - If the Dynamic Data Masking Server runs on Unix, enter the following command:

     `- server restart`
4. Open the Management Console and create the SNMP appender.

Note: If you add, remove, or update the files in the `<Dynamic Data Masking installation>/lib/ext` directory, you must configure the SNMP appender again.

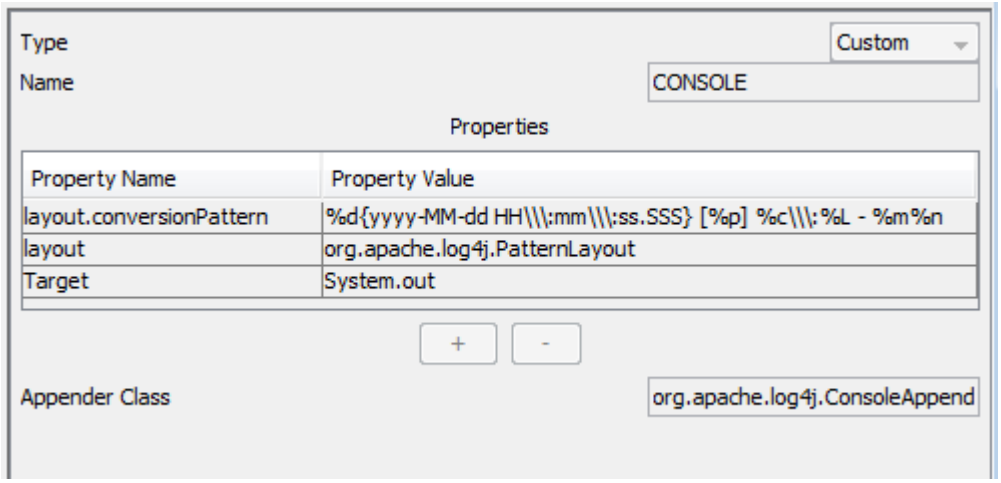
Custom Appender

Create a custom appender to log events in a custom format by using log4j classes.

A custom appender can use any log4j appender class and you can specify multiple properties for the appender.

The built-in appenders have hidden properties that you cannot modify. A custom appender is an appender that either has a class that does not match a built-in appender class or a property that does not match the hidden properties of the built-in class. For example, the Rolling File appender has a hidden encoding property set to UTF-8. You can create a custom rolling file appender that has a different encoding property.

The following image shows a custom appender that uses the console appender class:



Custom Appender Properties

The following table describes the Custom appender properties:

Property	Description
Name	The name of the appender. The appender name can contain alphabetic characters, numbers, and underscores (_), and must not exceed 60 characters.
Property Name	The name of the property.
Property Value	The value of the property. Property Value can be any value that is represented as a string.
Appender Class	The log4j appender class and package.

Creating an Appender

Create an appender to specify the format of the log information.

1. In the Management Console tree, select a logger node and click **Tree** > **Add Appender**. You can add an appender to a system logger or to a custom logger.

   The **Add Appender** window appears.
2. Select the type of appender that you want to create. The appender properties change based on the appender that you choose.
3. Enter the appender properties and click **OK**.

   The appender appears as a child node of the logger node that you selected.

Log Levels

The log level that you define in the Dynamic Data Masking Server node determines the severity of the event that you want to log. You specify a log level in a security rule to define the severity of individual events.

The Dynamic Data Masking Server node contains a Log Level property that the Dynamic Data Masking administrator can set to Information, Warning, or Error. The log level in the Dynamic Data Masking Server node corresponds to the Send As property of the Log Message security rule action. If the Send As property is an equal or greater severity than the Log Level property, the logger logs the event. If the Send As property is a lower severity than the Log Level property, the logger does not log the event.

For example, you have the Log Level property of the Dynamic Data Masking Server set to Warning. You have three security rules that use the Log Message rule action. The Send As property of Rule_1 is set to Information. Rule_2 has the property set to Warning, and Rule_3 has the property set to Error. The loggers will create logs for Rule_2 and Rule_3, but not for Rule_1.

The following table describes the log levels:

Log Level	Description
Information	Provides information about the normal behavior of the Dynamic Data Masking Server. Information logs can include information about when a service starts or stops and when a user logs in or if the log in fails.
Warning	Provides information that you can use to find and analyze non-fatal abnormal states in the Dynamic Data Masking Server. Warning logs can include information about a Dynamic Data Masking service start or stop failure, or an error that occurs when you add a node in the Management Console tree.
Error	Provides only error messages. Use the Error log level in production because it provides the best Dynamic Data Masking performance.

Setting the Log Level

Set the log level to specify the severity of the events that you want to log.

1. In the Management Console tree, select the Dynamic Data Masking Server node and click **Tree** > **Edit**.

   The **Edit** window appears.
2. Configure the **Log Level** property to the level that you want to log.
3. Click **OK**.

CHAPTER 8

High Availability

This chapter includes the following topics:

High Availability Overview

High availability refers to the uninterrupted availability of computer system resources. When you configure Dynamic Data Masking, you might configure high availability for the Dynamic Data Masking Server, high availability for the database that the Dynamic Data Masking Server connects to, or both.

Informatica recommends that you use standard vendor software to implement high availability. However, if you do not have high availability configured for your database, you can configure Dynamic Data Masking high availability. To send requests to a secondary database if the primary database is unavailable, configure high availability for the database. To send requests through a secondary Dynamic Data Masking Server if the primary Dynamic Data Masking Server is unavailable, configure high availability for the Dynamic Data Masking Server.

Database High Availability

Database high availability sends requests to a database based on whether a connection to the database exists. If you do not have a standard database high availability solution configured for the database, you can configure high availability in Dynamic Data Masking.

Create connection rules that use the Check Database Connection matcher to send requests to a primary database if a connection exists, and to a secondary database if the primary database connection does not exist. When you create the connection rule, you specify a database in the rule matcher and Dynamic Data Masking verifies whether a connection to the database exists.

For more information about the Check Database Connection Matcher, see the *Dynamic Data Masking User Guide*.

Configuring Database High Availability

Configure database high availability to send requests to a secondary database if the primary database is unavailable.

1. In the Management Console, create a database node with a connection to the primary database and a database node with a connection to the secondary database.
2. Add the Dynamic Data Masking service for the database type.
3. Open the connection rule set for the Dynamic Data Masking service and create a connection rule that sends requests to the primary database if a connection to the database exists. Configure the following properties for the connection rule:

   Matcher

   Select the Check Database Connection matcher. The Check Database Connection matcher verifies whether a connection to the database exists.

   Database

   Enter the name of database node for the primary database.

   Rule Action

   Select the Switch to Database action. The Switch to Database action sends the request to the database if the matcher identified a connection.

   Database

   Enter the name of database node for the primary database.

   Processing Action

   Select Stop if Applied. The Rule Engine does not continue to the next rule in the tree if Dynamic Data Masking applied the Switch to Database rule action.
4. In the connection rule set, create a rule that sends the request to the secondary database if a connection to the primary database does not exist. Configure the following properties for the connection rule:

   Matcher

   Select the Check Database Connection matcher.

   Database

   Enter the name of the database node for the secondary database.

   Rule Action

   Select the Switch to Database action.

   Database

   Enter the name of the database node for the secondary database.

   Processing Action

   Select Stop if Applied.
5. Save the connection rule set.

If a connection to the primary database exists, Dynamic Data Masking sends the request to the primary database. If a connection to the primary database does not exist, Dynamic Data Masking sends the request to the secondary database.

Dynamic Data Masking Server High Availability

Dynamic Data Masking Server high availability sends requests through a secondary Dynamic Data Masking Server if the primary Dynamic Data Masking Server is unavailable.

Informatica recommends that you use standard vendor solutions to provide Dynamic Data Masking high availability. For example, you might use failover clustering for Microsoft SQL Server.

To configure Dynamic Data Masking Server high availability for DB2, you can create connection rules that use the Load Control rule action. You must have at least two Dynamic Data Masking Server installations and the database client must use the IBM JDBC driver for DB2. If the Dynamic Data Masking Server installations are on the same machine, they must use different listener ports. After you configure Dynamic Data Masking Server high availability, you can connect to the database through either of the Dynamic Data Masking Servers. If one of the servers is unavailable, the request goes through the other server.

Configuring Dynamic Data Masking Server High Availability for DB2

Configure Dynamic Data Masking Server high availability to send requests through a secondary Dynamic Data Masking Server if the primary Dynamic Data Masking Server is unavailable.

Verify the following prerequisites before you configure Dynamic Data Masking Server high availability for DB2:

- You must have a DB2 database.
- You must have two Dynamic Data Masking Servers installed.
- You must have the IBM JDBC driver.

1. In the Management Console for the primary Dynamic Data Masking Server, create a connection to the DB2 database. The database client must use the IBM JDBC driver to connect to the listener port that you define for the connection.

   Note: If the Dynamic Data Masking Server installations are on the same machine, you must configure different listener port numbers for the primary and secondary servers.
2. Add the Dynamic Data Masking service for DB2.
3. Open the connection rule set for the Dynamic Data Masking service and create a rule folder that identifies requests to the database. Configure the following properties for the rule folder:

   Matcher

   Select the Incoming DDM Listener Port matcher. The Incoming DDM Listener Port matcher identifies requests based on the incoming listener port.

   Incoming Port

   Enter the listener port number for the primary Dynamic Data Masking Server.

   Rule Action

   Select the Folder rule action. The Folder rule action creates a rule folder.

   Processing Action

   Select the Stop if Matched processing action to process only the connection rules in the rule folder.
4. In the rule folder, create a connection rule that sets the priority levels of the Dynamic Data Masking Servers. Configure the following properties for the connection rule:

Matcher

Select the All Incoming Connections matcher. The All Incoming Connections matcher applies the rule action to all SQL requests.

Rule Action

Select the Load Control rule action. The Load Control rule action identifies the Dynamic Data Masking Servers and port numbers, and sets the server priority level. Configure the following Load Control properties:

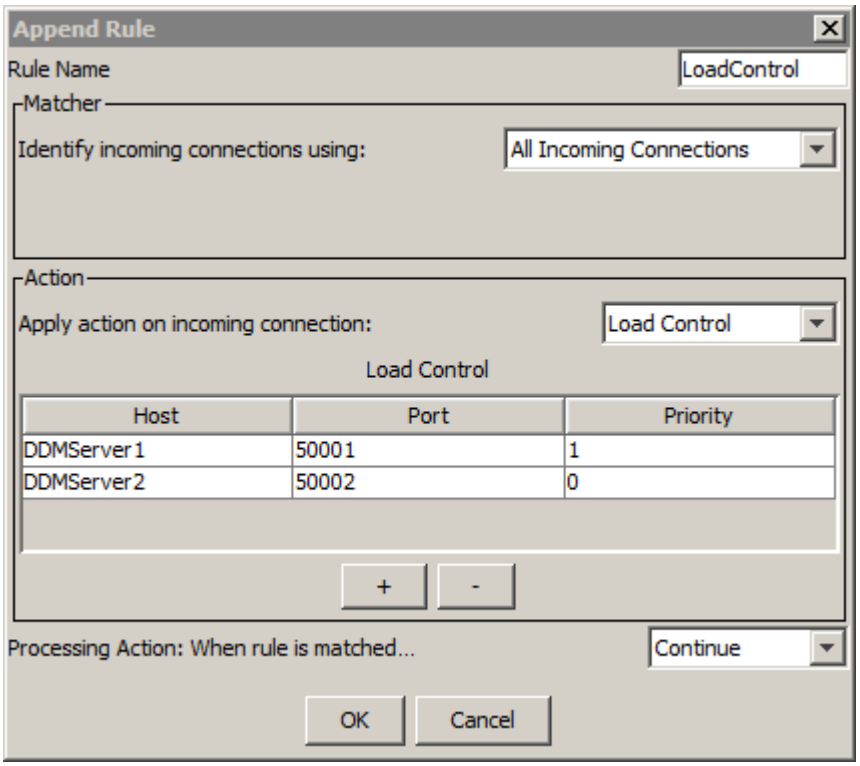
- **Host**. Enter the names of the Dynamic Data Masking Servers. Click the plus sign (+) to add additional servers.
- **Port**. Enter the port number for each of the Dynamic Data Masking Servers.
- **Priority**. Enter a priority number for each of the Dynamic Data Masking Servers. The value of the Priority property corresponds to the frequency that the client sends the request through the Dynamic Data Masking Server.

Note: For more information about the Load Control action and how to set priority levels, see the *Dynamic Data Masking User Guide*.

Processing Action

Select the Continue processing action. The Continue processing action sends the request to the next rule in the tree.

The following image shows an example of the connection rule:



5. In the connection rule folder, create another connection rule that sends the request to the database. Configure the following properties for the connection rule:

Matcher

Select the All Incoming Connections Matcher.

Rule Action

Select the Switch to Database action. The Switch to Database action sends the request to the database that you specify.

Database

Enter the name of the database node in the Management Console tree.

Processing Action

Select the Continue processing action.

6. Save the connection rule set for the primary Dynamic Data Masking Server.
7. In the Management Console for the secondary Dynamic Data Masking Server, create a connection to the DB2 database. The database client must use the IBM JDBC driver to connect to the listener port that you define for the connection.

   Note: If the Dynamic Data Masking Server installations are on the same machine, you must configure different listener port numbers for the primary and secondary servers.
8. Add the Dynamic Data Masking service for DB2.
9. Open the connection rule set for the Dynamic Data Masking service and create connection rules that are identical to the connection rules that you created for the primary Dynamic Data Masking Server except that the Load Control action priority levels are switched.
10. Configure the following properties for the rule folder:

    Matcher

    Select the Incoming DDM Listener Port matcher.

    Incoming Port

    Enter the listener port number for the secondary Dynamic Data Masking Server.

    Rule Action

    Select the Folder rule action.

    Processing Action

    Select the Stop if Matched processing action to process only the connection rules in the rule folder.
11. Configure the following properties for the first rule in the rule folder:

    Matcher

    Select the All Incoming Connections matcher.

    Rule Action

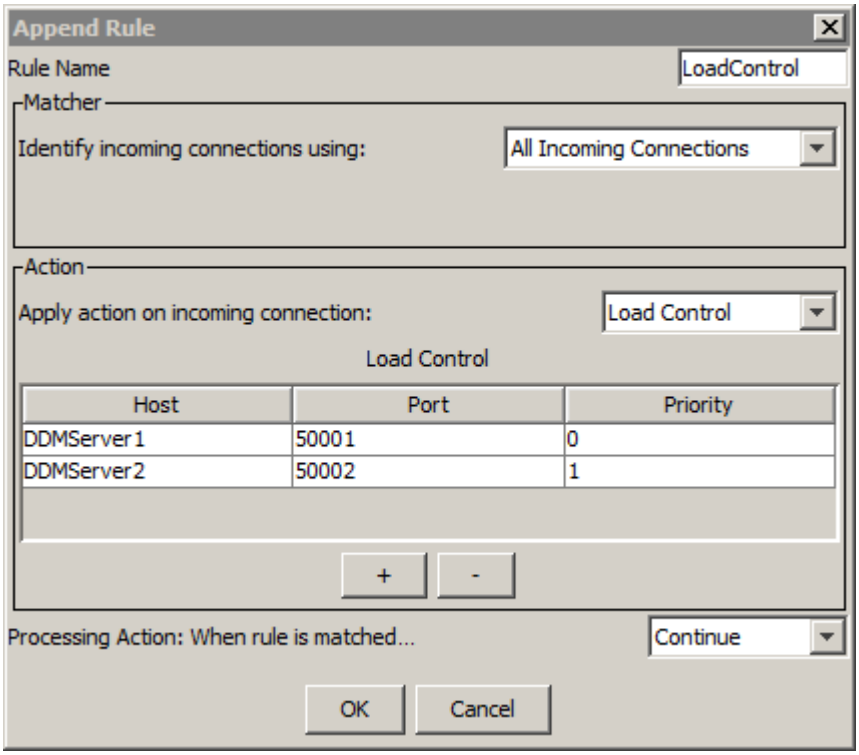
    Select the Load Control rule action. Configure the following Load Control properties:

    - **Host**. Enter the names of the Dynamic Data Masking Servers. Click the plus sign (+) to add additional servers.
    - **Port**. Enter the port number for each of the Dynamic Data Masking Servers.
    - **Priority**. Enter a priority number for each of the Dynamic Data Masking Servers. Enter one (1) for the secondary Dynamic Data Masking Server and zero (0) for the primary Dynamic Data Masking Server.

Processing Action

Select the Continue processing action.

The following image shows an example of the connection rule:



12. Configure the following properties for the second rule in the rule folder:

    Matcher

    Select the All Incoming Connections Matcher.

    Rule Action

    Select the Switch to Database action.

    Database

    Enter the name of the database node in the Management Console tree.

    Processing Action

    Select the Continue processing action.

13. Save the connection rule set for the secondary Dynamic Data Masking Server.

You can connect to the DB2 database through either of the Dynamic Data Masking Servers. If one of the Dynamic Data Masking Servers is unavailable, the request goes through the other Dynamic Data Masking Server.

To verify which Dynamic Data Masking Server receives requests, you can run Dynamic Data Masking in debug mode and check the log files to see which server provides debug information. You can also define database nodes for different databases on each server and check which database Dynamic Data Masking sends the request to.

CHAPTER 9

Server Control

This chapter includes the following topics:

Server Control Overview

The Dynamic Data Masking Server Control program is a command line interface that you use to manage Dynamic Data Masking Servers and services. Server Control reads administrative requests and writes the results of the requests to the standard system output and error streams.

Server Control has a set of commands that simplify management and configuration of local and remote Dynamic Data Masking Servers. Server Control is installed with the Dynamic Data Masking Server. Run Server Control on the machine where you installed the Dynamic Data Masking Server.

You can run the following types of commands from Server Control:

server

Use `server` commands to configure the local Dynamic Data Masking Server and Dynamic Data Masking services. For example, you can start and stop the Dynamic Data Masking Server and services, set the Dynamic Data Masking listener port, and view the Dynamic Data Masking Server version.

You can run the equivalents of the following Server Control server commands from shell scripts:

- start
- stop
- restart
- startDDMService
- stopDDMService
- restartDDMService
- status

server config

Use `server config` commands to perform configuration tasks on local and remote Dynamic Data Masking Servers. For example, you can set Dynamic Data Masking database passwords, synchronize Dynamic Data Masking Server configurations, and export and import security rule sets and Dynamic Data Masking databases.

server service

Use `server service` commands to manage Dynamic Data Masking services. For example, you can import and export Dynamic Data Masking services.

Important: Server Control stores information in the config.properties file. You must not modify the config.properties file.

Running Server Control

Run Server Control to manage the Dynamic Data Masking Server from a command line interface.

If you run the Dynamic Data Masking Server on Windows, Server Control runs as a batch file with a .bat extension.

If you run the Dynamic Data Masking Server on Linux, Server Control runs as a shell script with no extension. You can run the `server` shell script to use the Server Control commands. Alternatively, you can run a subset of the Server Control commands from individual shell scripts. For example, to start the Dynamic Data Masking Server, you can run the `server` shell script with the start command or you can run the `start` shell script directly.

Running Server Control on Windows

On Windows, run the Server Control command line program from the Start Menu.

1. Select **Start** > **Informatica** > **Dynamic Data Masking** > **Server Control**.
2. At the command line, enter commands with the following syntax:

```
server <command name> <parameter>
```

For example, the following command sets the port for the Dynamic Data Masking Server to 8195:

```
server setPort 8195
```

Running Server Control on Linux or UNIX

On Linux, use the `server` shell script to run Server Control commands. Alternatively, run Server Control commands from individual shell scripts.

1. Open a terminal, and navigate to the Dynamic Data Masking Server installation directory.

   For example, you might navigate to the following directory:

   ```
   /home/Informatica/DDM
   ```

2. Run the shell script for the Server Control command that you want to use.
   - If you run the `server` shell script, enter commands with the following syntax:

     ```
     ./server <command name> <parameter>
     ```

   - If you run a shell script for a specific command, run the shell script with the following syntax:

     ```
     ./<shell script name> <parameter>
     ```

For example, the following command uses the `server` shell script to start the DDM for Oracle service:

```
./server startDDMService "DDM for Oracle"
```

Alternatively, the following command uses the `startDDMService` shell script to start the DDM for Oracle service:

```
./startDDMService "DDM for Oracle"
```

Server Control Commands

Enter commands and parameters in the Server Control command line program to manage the Dynamic Data Masking Servers and services.

Use the following rules when you enter commands and parameters:

- The first word after a command is the parameter.
- If a parameter contains spaces, enclose the parameter in double quotes.
- Server Control commands are not case sensitive.

Syntax Notation

Before you use Server Control, review the syntax notation.

The following table describes the Server Control syntax notation:

Convention	Description
<x>	Required parameter. If you omit a required parameter, Server Control returns an error message.
[x]	Optional parameter. The command runs whether or not you enter optional parameters.
-x	Parameter placed before an argument that designates the argument that you enter. For example, to enter a target Dynamic Data Masking Server, type `-targets` followed by the Dynamic Data Masking Server name.

Server Commands

Use Server Control `server` commands to configure the local Dynamic Data Masking Server and services.

Server Control has the following `server` commands:

- checkPort
- help
- log
- remove

- rename
- restart
- restartDDMService
- services
- setInternalPassword
- setPort
- start
- startDDMService
- status
- stop
- stopDDMService
- version

CheckPort

Checks if a port is available or locked. Use the command to identify ports that you can use for the Dynamic Data Masking Server.

The `checkPort` command uses the `port` parameter. The value of the `port` parameter is the port number that you want to check for the Dynamic Data Masking Server. The command uses the following syntax:

```
server
  checkPort <port>
```

For example, you might enter the following command:

```
server checkPort 6002
```

Help

Displays descriptions and parameters for each Server Control command.

The command uses the following syntax:

```
server
  help
```

Log

Sets and displays the Dynamic Data Masking Server log level.

If you do not include a `Log_Level` parameter in the command, Server Control displays the current log level. You can set the following log levels:

- INFO
- DEBUG
- WARN
- ERROR

The command uses the following syntax:

```
server
  log <Log_Level>
```

For example, you might enter the following command:

```
server log warn
```

Remove

Removes the service or daemon for the Dynamic Data Masking Server. The Dynamic Data Masking Server must be stopped to run the command.

The command uses the following syntax:

```
server
  remove
```

Rename

Renames the Dynamic Data Masking service on Windows or the daemon on Linux. The Dynamic Data Masking Server must be shut down to run the command.

The `rename` command uses the `name` parameter. The value of the `name` parameter is the name that you want to set for the Dynamic Data Masking Server. The command uses the following syntax:

```
server
  rename <name>
```

For example, you might enter the following command:

```
server rename Informatica_DDM
```

On Linux and UNIX, Dynamic Data Masking does not check to verify that the Dynamic Data Masking Server name you choose is not in use. Before you change the Server name, make sure that there is not a Dynamic Data Masking Server in the same environment with the same name.

Restart

Restarts the Dynamic Data Masking Server.

The command uses the following syntax:

```
server
  restart
```

On Linux, you can also run the `restart` shell script to restart the Dynamic Data Masking Server.

RestartDDMService

Restarts a Dynamic Data Masking service configured in the Management Console. The Dynamic Data Masking Server must be running.

The `restartDDMService` command uses the `DDMService_Name` parameter. The value of the `DDMService_Name` parameter is the name of the Dynamic Data Masking service that you want to restart.

The command uses the following syntax:

```
server
  restartDDMService <DDMService_Name>
```

For example, you might enter the following command:

```
server restartDDMService "DDM for SQL Server"
```

On Linux, you can also run the `restartDDMService` shell script to restart a Dynamic Data Masking service.

Services

Lists Dynamic Data Masking services in the Management Console Tree. The Dynamic Data Masking Server must be running.

The command uses the following syntax:

```
server
  services
```

SetInternalPassword

Sets a value for the Dynamic Data Masking Server internal password. The Dynamic Data Masking Server must be shut down to execute the command.

The `setInternalPassword` command uses the `password` parameter. The value for the `password` parameter is the internal password that you want to set for the Dynamic Data Masking Server.

The command uses the following syntax:

```
server
  setInternalPassword <password>
```

For example, you might enter the following command:

```
server setInternalPassword SpF_dPHx
```

SetPort

Sets the value of the server management port. The Dynamic Data Masking Server must be shut down to run the command.

The `setPort` command uses the `port` parameter. The value of the `port` parameter is the port number that you want to set for the Dynamic Data Masking Server.

The command uses the following syntax:

```
server
  setPort <port>
```

For example, you might enter the following command:

```
server setPort 6002
```

Start

Starts the Dynamic Data Masking Server. Creates an operating system service in Windows if the service does not exist. The Dynamic Data Masking Server must be shut down to run the command.

The command uses the following syntax:

```
server
  start
```

On Linux, you can also run the `start` shell script to start the Dynamic Data Masking Server.

StartDDMService

Starts a Dynamic Data Masking service configured on the Dynamic Data Masking Server. The Dynamic Data Masking Server must be running.

The `startDDMService` command uses the `DDMService_Name` parameter. The value of the `DDMService_Name` parameter is the name of the Dynamic Data Masking service that you want to start.

The command uses the following syntax:

```
server
  startDDMService <DDMService_Name>
```

For example, you might enter the following command:

```
server startDDMService "DDM for Oracle"
```

On Linux, you can also run the `startDDMService` shell script to start a Dynamic Data Masking service.

Status

Shows whether the Dynamic Data Masking Server is running. The command also displays the name of the Dynamic Data Masking Server and the number of the server management port.

The command uses the following syntax:

```
server
  status
```

On Linux, you can also run the `isStarted` shell script to show the status of the Dynamic Data Masking Server.

Stop

Stops the Dynamic Data Masking Server.

The command uses the following syntax:

```
server
  stop
```

On Linux, you can also run the `stop` shell script to stop the Dynamic Data Masking Server.

StopDDMService

Stops a Dynamic Data Masking service configured on the Dynamic Data Masking Server. The Dynamic Data Masking Server must be running.

The `stopDDMService` command uses the `DDMService_Name` parameter. The value of the `DDMService_Name` parameter is the name of the Dynamic Data Masking service that you want to stop.

The command uses the following syntax:

```
server
  stopDDMService <DDMService_Name>
```

For example, you might enter the following command:

```
server stopDDMService "DDM for DB2"
```

On Linux, you can also run the `stopDDMService` shell script to stop a Dynamic Data Masking service.

Version

Displays the name and version of the Dynamic Data Masking Server. The Dynamic Data Masking Server must be running.

The command uses the following syntax:

```
server
  version
```

Server Config Commands

Use Server Control `server config` commands to perform configuration tasks on local and remote Dynamic Data Masking Servers.

Server Control has the following `server config` commands:

- export
- import
- setDBPassword
- sync

Export

Exports a Dynamic Data Masking database or security rule set. The Dynamic Data Masking Server must be running.

Use the `export` command to export a Dynamic Data Masking database or security rule set from the Dynamic Data Masking Server. You can then import the file into one or more Dynamic Data Masking Servers.

If you do not specify a source Dynamic Data Masking Server, the `export` command exports the object from the local Dynamic Data Masking Server.

The command uses the following syntax:

```
server config
  export [/y] <object full path> <file>
  [[-source] user/pwd@host:port]
```

For example, if you want to export a database named DEV_SYBASE_DB from the local Dynamic Data Masking Server and name the file DEV_SYBASE_DB_1, you might enter the following command:

```
server config export "Site\DEV_SYBASE_DB" "DEV_SYBASE_DB_1"
```

If you want to export the DEV_SYBASE_DB_1 database from a remote Dynamic Data Masking Server, you might enter the following command:

```
server config export "Site\DEV_SYBASE_DB" "DEV_SYBASE_DB_1" -source admin/
admin@ABC12345:7000
```

The `export` command uses the following parameters:

/y

Forces the command to continue without user confirmation. If you use the `/y` parameter, Server Control does not require confirmation to overwrite an existing file. For example, you might use the `/y` parameter in a script.

object full path

The path in the Management Console of the object that you want to export.

file

The name of the file that the object exports to.

-source

Indicates that the following argument is a source.

user

The Dynamic Data Masking user name that you use to log in to the Dynamic Data Masking Server.

pwd

The password for the Dynamic Data Masking user.

host

The Dynamic Data Masking Server host name or IP address.

port

The Dynamic Data Masking Server listener port.

Import

Imports a Dynamic Data Masking database or security rule set. The Dynamic Data Masking Server must be running.

Use the `import` command to import a Dynamic Data Masking database or security rule set into one or more Dynamic Data Masking Servers. The type of object that you import must be the same type of object that you specify as the location to import the object. For example, you can import a security rule set into a security rule set. You can import an Oracle database into an Oracle database node, but you cannot import an Oracle database into a Sybase database node.

If you do not specify a target Dynamic Data Masking Server, the `import` command imports the object into the local Dynamic Data Masking Server. Use the `-targets` parameter to specify multiple target Dynamic Data Masking Servers.

The command uses the following syntax:

```
server config
  import <object full path> <file>
  [[-targets] user/pwd@host:port [user/pwd@host:port user/pwd@host:port]]
```

For example, if you exported a database named DEV_SYBASE_DB and you named the file DEV_SYBASE_DB_1, and you want to import the database into the local Dynamic Data Masking Server, you might enter the following command:

```
server config import "Site\DEV_SYBASE_DB" "DEV_SYBASE_DB_1"
```

If you want to import the DEV_SYBASE_DB_1 file into multiple Dynamic Data Masking Servers, you might enter the following command:

```
server config import "Site\DEV_SYBASE_DB" "DEV_SYBASE_DB_1" -targets admin/
admin@ABC12345:1111 admin/admin@XYZ98765:2222
```

The `import` command uses the following parameters:

object full path

The full path in the Management Console where you want to import the object. If the parent path does not exist in the Management Console tree, the command returns an error. If the object does not exist in the path, Dynamic Data Masking creates the object. If the object exists in the path, the object must be the same object type as the object that you want to import.

For example, you want to import a database into the following location in the Management Console tree:

```
Site\backup\SQL_SERVER_DB
```

The parent path is Site\backup and the object is SQL_SERVER_DB. The parent path must exist in the Management Console tree. If the object does not exist, Dynamic Data Masking creates a database node named SQL_SERVER_DB.

file

The file name of the object that you want to import.

-targets

Indicates that the following arguments are targets.

user

The Dynamic Data Masking user name that you use to log in to the Dynamic Data Masking Server.

pwd

The password for the Dynamic Data Masking user.

host

The Dynamic Data Masking Server host name or IP address.

port

The Dynamic Data Masking Server listener port.

SetDBPassword

Sets the password for the Dynamic Data Masking database. The Dynamic Data Masking Server must be running.

If you do not specify a target Dynamic Data Masking Server, the `setDBPassword` command updates the password of the database on the local Dynamic Data Masking Server. Use the `-targets` parameter to specify multiple target Dynamic Data Masking Servers.

The command uses the following syntax:

```
server config
  setDBPassword <dbpath> <oldDBPassword> <newDBPassword>
  [[-targets] user/pwd@host:port [user/pwd@host:port user/pwd@host:port]]
```

For example, if you have a database named DEV_SYBASE_DB_1 under a root node named Site on the local Dynamic Data Masking Server, you might enter the following command:

```
server config setDBPassword "Site\DEV_SYBASE_DB_1" ddmadmin ddmadmin1
```

If you want to change the password for the database on multiple Dynamic Data Masking Servers, you might enter the following command:

```
server config setDBPassword "Site\DEV_SYBASE_DB_1" ilmuser ilmuser1 -targets admin/
admin@ABC12345:1111 admin/admin@XYZ9876:2222
```

The `setDBPassword` command uses the following parameters:

dbpath

The path to the database in the Management Console tree.

oldDBPassword

The password that Dynamic Data Masking uses to connect to the database.

newDBPassword

The new password that you want Dynamic Data Masking to use to connect to the database.

-targets

Indicates that the following arguments are targets.

user

The Dynamic Data Masking user name that you use to log in to the Dynamic Data Masking Server.

pwd

The password for the Dynamic Data Masking user name.

host

The Dynamic Data Masking Server host name or IP address.

port

The Dynamic Data Masking Server listener port.

Sync

Synchronizes the Management Console tree of one or more target Dynamic Data Masking Servers with the Management Console tree of a source Dynamic Data Masking Server. The source and target Dynamic Data Masking Servers must be running.

The `sync` command synchronizes databases, and security rule sets. You cannot synchronize the Dynamic Data Masking Server, Dynamic Data Masking services, connection rules, and loggers. To copy a Dynamic Data Masking service, you must export the service from the source location and import the service into the target location.

You can specify one source Dynamic Data Masking Server and one or more target Dynamic Data Masking Servers. You must specify at least one target or source. If you specify a source and not a target, the `sync` command uses the local Dynamic Data Masking Server as the target. If you specify a target and not a source, the `sync` command uses the local Dynamic Data masking Server as the source.

The command uses the following syntax:

```
server config
  sync [[-source] user/pwd@host:port]
  [-targets user/pwd@host:port [user/pwd@host:port user/pwd@host:port]]
```

For example, if you want to copy the Management Console tree of a remote Dynamic Data Masking Server into the local Dynamic Data Masking Server, you might enter the following command:

```
server config sync -source admin/admin@ABC12345:1111
```

If you want to copy the Management Console tree of a remote Dynamic Data Masking Server into multiple remote Dynamic Data Masking Servers, you might enter the following command:

```
server config sync -source admin/admin@ABC12345:1111 -targets admin/admin@XYZ9876:2222
admin/admin@DEF1234:3333
```

The `sync` command uses the following parameters:

-source

Indicates that the following argument is a source.

user

The Dynamic Data Masking user name that you use to log in to the Dynamic Data Masking Server.

pwd

The password for the Dynamic Data Masking user name.

host

The Dynamic Data Masking Server host name or IP address.

port

The Dynamic Data Masking Server listener port.

-targets

Indicates that the following arguments are targets.

Server Service Commands

Use Server Control `server service` commands to manage Dynamic Data Masking services.

Server Control has the following `server service` commands:

- export
- import

Export

Exports a Dynamic Data Masking service. The Dynamic Data Masking Server must be running.

Use the `export` command to export a Dynamic Data Masking service and the connection rules associated with the service into a file. You can then import the file into one or more Dynamic Data Masking Servers.

If you do not specify a source Dynamic Data Masking Server, the `export` command exports the service from the local Dynamic Data Masking Server.

The command uses the following syntax:

```
server service
  export [/y] <DDM Service> <file>
  [[-source] user/pwd@host:port]
```

For example, if you want to export the DDM for Oracle service from the local Dynamic Data Masking Server and name the file DDM_for_Oracle, you might enter the following command:

```
server service export "DDM for Oracle" "DDM_for_Oracle"
```

If you want to export the DDM_for_Oracle service from a remote Dynamic Data Masking Server, you might enter the following command:

```
server service export /y "DDM for Oracle" "DDM_for_Oracle" -source admin/
admin@ABC12345:1111
```

The `export` command uses the following parameters:

/y

Forces the command to continue without user confirmation. If you use the `/y` parameter, Server Control does not require confirmation to overwrite an existing file. For example, you might use the `/y` parameter in a script.

DDM Service

The Dynamic Data Masking service that you want to export.

file

The name of the file that the Dynamic Data Masking service exports to.

-source

Indicates that the following argument is a source.

user

The Dynamic Data Masking user name that you use to log in to the Dynamic Data Masking Server.

pwd

The password for the Dynamic Data Masking user.

host

The Dynamic Data Masking Server host name or IP address.

port

The Dynamic Data Masking Server listener port.

Note: You cannot use the `sync` command to synchronize Dynamic Data Masking services. To copy a service, you must export the service from the source location and import the service into the target location.

Import

Imports a Dynamic Data Masking service. The Dynamic Data Masking Server must be running.

Use the `import` command to import a Dynamic Data Masking service and the connection rules associated with the service into a Dynamic Data Masking Server. The type of service that you import must be the same type of service that you specify as the location to import the service. For example, you can import a DDM for Oracle service into a DDM for Oracle service, but you cannot import a DDM for Oracle service into a DDM for Sybase service.

If you do not specify a target Dynamic Data Masking Server, the `import` command imports the service into the local Dynamic Data Masking Server. Use the `-targets` parameter to specify multiple target Dynamic Data Masking Servers.

The command uses the following syntax:

```
server service
     import <DDM Service> <file>
  [[-targets] user/pwd@host:port[user/pwd@host:port user/pwd@host:port]]
```

For example, if you exported a DDM for Oracle service and named the file DDM_for_Oracle, and you want to import the service into the local Dynamic Data Masking Server, you might enter the following command:

```
server service import "DDM for Oracle" "DDM_for_Oracle"
```

If you want to import the DDM_for_Oracle file into multiple Dynamic Data Masking Servers, you might enter the following command:

```
server service import "DDM for Oracle" "DDM_for_Oracle" -targets admin/
admin@ABC12345:1111 admin/admin@XYZ9876:2222
```

The `import` command uses the following parameters:

DDM Service

The name of the Dynamic Data Masking service that you want to import. You must enter a valid name for the service, such as "DDM for Oracle." If the service that you want to import does not exist, Dynamic Data Masking creates the service in the Management Console tree. If you specify the name of an existing service, the type of service that you import must be the same type as the service that you import into.

file

The file name of the Dynamic Data Masking service that you want to import.

-targets

Indicates that the following arguments are targets.

user

The Dynamic Data Masking user name that you use to log in to the Dynamic Data Masking Server.

pwd

The password for the Dynamic Data Masking user.

host

The Dynamic Data Masking Server host name or IP address.

port

The Dynamic Data Masking Server listener port.

Note: You cannot use the `sync` command to synchronize Dynamic Data Masking services. To copy a service, you must export the service from the source location and import the service into the target location.

CHAPTER 10

Performance Tuning

This chapter includes the following topics:

Performance Tuning Overview

Performance tuning efficiently allocates resources to optimize the performance of Dynamic Data Masking in production and non-production environments. When you implement performance tuning techniques, you help to ensure that the Dynamic Data Masking opertions do not affect database and application performance. To tune Dynamic Data Masking performance, you can change the log levels, reduce parsing errors, and define an efficient user stack limit.

Dynamic Data Masking Resource Consumption

Dynamic Data Masking consumes CPU resources to process connections, rewrite requests, and store logs. The CPU consumption is linearly proportional to the SQL *Net traffic that the Dynamic Data Masking service routes. You can use the SQL *Net and DBlinks traffic values to estimate the amount of CPU that the Dynamic Data Masking service consumes.

The Dynamic Data Masking resource consumption is approximately 1% of the server CPU and has no I/O overhead. The Dynamic Data Masking service requires approximately 1 GB for memory and some disk space for logs.

To calculate the CPU consumption, you must determine the round-trip value, which is the total packet traffic that a client and server sends and receives each second. The total round-trip value includes SQL *Net traffic and DBlinks.

Use the following variables and equations to calculate the CPU consumption:

Variables

X1 = SQL *Net round-trip value

X2 = DBlinks round-trip value

PR = Total round-trip packets for each second

Equations

PR = (X1+X2)*2

CPU Consumption = PR/10,000

Log Performance

Log files use server resources to record service information and events. The tracing level determines the amount of information that the log stores.

Logs consume system resources and can slow down performance. To improve log performance, you can change the tracing level to reduce the amount of information that the log stores. By default, each log uses the information tracing level. The information tracing level is a high impact tracing level and uses the most server resources.

Note: If you change the tracing level, it does not affect the tracing level for SQL Server.

The following table describes the different tracing levels:

Tracing Level	Description
Information	The default log level. Logs all information messages and provides comprehensive information about the Dynamic Data Masking service. The information that the log provides is useful if you encounter an issue with Dynamic Data Masking operations. You can refer to the logs to troubleshoot the problem. Performance impact is high.
Warn	Logs warning messages from the Dynamic Data Masking service. Performance impact is moderate.
Error	Logs error messages and instances when a user connection is force closed. Performance impact is low.

Configuring the Tracing Level

Configure the tracing level in the Management Console with a low impact tracing level to reduce resource consumption.

1. In the Management Console, click the Dynamic Data Masking Server.
2. Select **Tree** > **Edit**.

   The Dynamic Data Masking Server configuration window appears.
3. Select the log level from the Log Level menu.
4. Click **OK**.

Dynamic Data Masking Latency

Latency measures the amount of time it takes a SQL request to travel from its source to its destination. Latency varies between platforms and ranges between 75-150 microseconds for each packet.

In addition to latency, the processing time impacts performance. The processing time depends on the matcher or action type that you use. The types of matcher and action classifications are low, medium, and high.

The following table describes the processing time for each matcher type:

Performance Impact	Matcher/Action	Latency
Low	- Any matcher - Text matcher - Time of day matcher - Block action - Rewrite action - Search and Replace action - Define Symbol action - Nothing action	Less than 1 microsecond
Medium	- SQL Syntax matcher - From Clause Object matcher	1-100 microseconds
High	PL/SQL Function matcher	The processing time depends on the PL/SQL processing time.

User Stack Limit

High user stack limits cause high CPU consumption and connection refusal.

Set the user stack limit, or ulimit, to 1,024 kilobytes to ensure that system can create threads. To view the user stack limit, enter `#ulimit -s` in the command shell. The system returns the value in kilobytes.

Calculating the User Stack Limit

An insufficient number of available threads cause high CPU load and client request delays. Before you change the user stack limit, verify the minimum number of required threads necessary for the system.

Use the following formula to calculate the minimum number of threads:

```
<number of concurrent sessions> x 10
```

Insufficient Thread Error

The following error appears in server.log if the number of open files is insufficient:

```
12/16 13:24:51,597 [dnr-1] WARN - Service dnr.createClientPeer: received IOException
while listening:
java.io.IOException: Too many open files
```

INDEX

G

H

I

J

L

M

N

O

P

R

S

T

U

W