Informatica

Informatica® Dynamic Data Masking
9.8.2

Stored Procedure Accelerator Guide for Microsoft SQL Server

Informatica Dynamic Data Masking Stored Procedure Accelerator Guide for Microsoft SQL Server
9.8.2
October 2016

© Copyright Informatica LLC 1993, 2018

This software and documentation contain proprietary information of Informatica LLC and are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright law. Reverse engineering of the software is prohibited. No part of this document may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording or otherwise) without prior consent of Informatica LLC. This Software may be protected by U.S. and/or international Patents and other Patents Pending.

Use, duplication, or disclosure of the Software by the U.S. Government is subject to the restrictions set forth in the applicable software license agreement and as provided in DFARS 227.7202-1(a) and 227.7702-3(a) (1995), DFARS 252.227-7013©(1)(ii) (OCT 1988), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14 (ALT III), as applicable.

The information in this product or documentation is subject to change without notice. If you find any problems in this product or documentation, please report them to us in writing.

Informatica, Informatica Platform, Informatica Data Services, PowerCenter, PowerCenterRT, PowerCenter Connect, PowerCenter Data Analyzer, PowerExchange, PowerMart, Metadata Manager, Informatica Data Quality, Informatica Data Explorer, Informatica B2B Data Transformation, Informatica B2B Data Exchange Informatica On Demand, Informatica Identity Resolution, Informatica Application Information Lifecycle Management, Informatica Complex Event Processing, Ultra Messaging, Informatica Master Data Management, and Live Data Map are trademarks or registered trademarks of Informatica LLC in the United States and in jurisdictions throughout the world. All other company and product names may be trade names or trademarks of their respective owners.

Portions of this software and/or documentation are subject to copyright held by third parties, including without limitation: Copyright DataDirect Technologies. All rights reserved. Copyright © Sun Microsystems. All rights reserved. Copyright © RSA Security Inc. All Rights Reserved. Copyright © Ordinal Technology Corp. All rights reserved. Copyright © Aandacht c.v. All rights reserved. Copyright Genivia, Inc. All rights reserved. Copyright Isomorphic Software. All rights reserved. Copyright © Meta Integration Technology, Inc. All rights reserved. Copyright © Intalio. All rights reserved. Copyright © Oracle. All rights reserved. Copyright © Adobe Systems Incorporated. All rights reserved. Copyright © DataArt, Inc. All rights reserved. Copyright © ComponentSource. All rights reserved. Copyright © Microsoft Corporation. All rights reserved. Copyright © Rogue Wave Software, Inc. All rights reserved. Copyright © Teradata Corporation. All rights reserved. Copyright © Yahoo! Inc. All rights reserved. Copyright © Glyph & Cog, LLC. All rights reserved. Copyright © Thinkmap, Inc. All rights reserved. Copyright © Clearpace Software Limited. All rights reserved. Copyright © Information Builders, Inc. All rights reserved. Copyright © OSS Nokalva, Inc. All rights reserved. Copyright Edifecs, Inc. All rights reserved. Copyright Cleo Communications, Inc. All rights reserved. Copyright © International Organization for Standardization 1986. All rights reserved. Copyright © ej-technologies GmbH. All rights reserved. Copyright © Jaspersoft Corporation. All rights reserved. Copyright © International Business Machines Corporation. All rights reserved. Copyright © yWorks GmbH. All rights reserved. Copyright © Lucent Technologies. All rights reserved. Copyright (c) University of Toronto. All rights reserved. Copyright © Daniel Veillard. All rights reserved. Copyright © Unicode, Inc. Copyright IBM Corp. All rights reserved. Copyright © MicroQuill Software Publishing, Inc. All rights reserved. Copyright © PassMark Software Pty Ltd. All rights reserved. Copyright © LogiXML, Inc. All rights reserved. Copyright © 2003-2010 Lorenzi Davide, All rights reserved. Copyright © Red Hat, Inc. All rights reserved. Copyright © The Board of Trustees of the Leland Stanford Junior University. All rights reserved. Copyright © EMC Corporation. All rights reserved. Copyright © Flexera Software. All rights reserved. Copyright © Jinfonet Software. All rights reserved. Copyright © Apple Inc. All rights reserved. Copyright © Telerik Inc. All rights reserved. Copyright © BEA Systems. All rights reserved. Copyright © PDFlib GmbH. All rights reserved. Copyright © Orientation in Objects GmbH. All rights reserved. Copyright © Tanuki Software, Ltd. All rights reserved. Copyright © Ricebridge. All rights reserved. Copyright © Sencha, Inc. All rights reserved. Copyright © Scalable Systems, Inc. All rights reserved. Copyright © jQWidgets. All rights reserved. Copyright © Tableau Software, Inc. All rights reserved. Copyright© MaxMind, Inc. All Rights Reserved. Copyright © TMate Software s.r.o. All rights reserved. Copyright © MapR Technologies Inc. All rights reserved. Copyright © Amazon Corporate LLC. All rights reserved. Copyright © Highsoft. All rights reserved. Copyright © Python Software Foundation. All rights reserved. Copyright © BeOpen.com. All rights reserved. Copyright © CNRI. All rights reserved.

This product includes software developed by the Apache Software Foundation (http://www.apache.org/), and/or other software which is licensed under various versions of the Apache License (the "License"). You may obtain a copy of these Licenses at http://www.apache.org/licenses/. Unless required by applicable law or agreed to in writing, software distributed under these Licenses is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the Licenses for the specific language governing permissions and limitations under the Licenses.

This product includes software which was developed by Mozilla (http://www.mozilla.org/), software copyright The JBoss Group, LLC, all rights reserved; software copyright © 1999-2006 by Bruno Lowagie and Paulo Soares and other software which is licensed under various versions of the GNU Lesser General Public License Agreement, which may be found at http:// www.gnu.org/licenses/lgpl.html. The materials are provided free of charge by Informatica, "as-is", without warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose.

The product includes ACE(TM) and TAO(TM) software copyrighted by Douglas C. Schmidt and his research group at Washington University, University of California, Irvine, and Vanderbilt University, Copyright (©) 1993-2006, all rights reserved.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (copyright The OpenSSL Project. All Rights Reserved) and redistribution of this software is subject to terms available at http://www.openssl.org and http://www.openssl.org/source/license.html.

This product includes Curl software which is Copyright 1996-2013, Daniel Stenberg, <daniel@haxx.se>. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at http://curl.haxx.se/docs/copyright.html. Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

The product includes software copyright 2001-2005 (©) MetaStuff, Ltd. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at http://www.dom4j.org/ license.html.

The product includes software copyright © 2004-2007, The Dojo Foundation. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at http://dojotoolkit.org/license.

This product includes ICU software which is copyright International Business Machines Corporation and others. All rights reserved. Permissions and limitations regarding this software are subject to terms available at http://source.icu-project.org/repos/icu/icu/trunk/license.html.

This product includes software copyright © 1996-2006 Per Bothner. All rights reserved. Your right to use such materials is set forth in the license which may be found at http:// www.gnu.org/software/ kawa/Software-License.html.

This product includes OSSP UUID software which is Copyright © 2002 Ralf S. Engelschall, Copyright © 2002 The OSSP Project Copyright © 2002 Cable & Wireless Deutschland. Permissions and limitations regarding this software are subject to terms available at http://www.opensource.org/licenses/mit-license.php.

This product includes software developed by Boost (http://www.boost.org/) or under the Boost software license. Permissions and limitations regarding this software are subject to terms available at http:/ /www.boost.org/LICENSE_1_0.txt.

This product includes software copyright © 1997-2007 University of Cambridge. Permissions and limitations regarding this software are subject to terms available at http:// www.pcre.org/license.txt.

This product includes software copyright © 2007 The Eclipse Foundation. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at http:// www.eclipse.org/org/documents/epl-v10.php and at http://www.eclipse.org/org/documents/edl-v10.php.

This product includes software licensed under the terms at http://www.tcl.tk/software/tcltk/license.html, http://www.bosrup.com/web/overlib/?License, http://www.stlport.org/doc/ license.html, http://asm.ow2.org/license.html, http://www.cryptix.org/LICENSE.TXT, http://hsqldb.org/web/hsqlLicense.html, http://httpunit.sourceforge.net/doc/ license.html, http://jung.sourceforge.net/license.txt , http://www.gzip.org/zlib/zlib_license.html, http://www.openldap.org/software/release/license.html, http://www.libssh2.org, http://slf4j.org/license.html, http://www.sente.ch/software/OpenSourceLicense.html, http://fusesource.com/downloads/license-agreements/fuse-message-broker-v-5-3- license-agreement; http://antlr.org/license.html; http://aopalliance.sourceforge.net/; http://www.bouncycastle.org/licence.html; http://www.jgraph.com/jgraphdownload.html; http://www.jcraft.com/jsch/LICENSE.txt; http://jotm.objectweb.org/bsd_license.html; . http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231; http://www.slf4j.org/license.html; http://nanoxml.sourceforge.net/orig/copyright.html; http://www.json.org/license.html; http://forge.ow2.org/projects/javaservice/, http://www.postgresql.org/about/licence.html, http://www.sqlite.org/copyright.html, http://www.tcl.tk/software/tcltk/license.html, http://www.jaxen.org/faq.html, http://www.jdom.org/docs/faq.html, http://www.slf4j.org/license.html; http://www.iodbc.org/dataspace/iodbc/wiki/iODBC/License; http://www.keplerproject.org/md5/license.html; http://www.toedter.com/en/jcalendar/license.html; http://www.edankert.com/bounce/index.html; http://www.net-snmp.org/about/license.html; http://www.openmdx.org/#FAQ; http://www.php.net/license/3_01.txt; http://srp.stanford.edu/license.txt; http://www.schneier.com/blowfish.html; http://www.jmock.org/license.html; http://xsom.java.net; http://benalman.com/about/license/; https://github.com/CreateJS/EaselJS/blob/master/src/easeljs/display/Bitmap.js; http://www.h2database.com/html/license.html#summary; http://jsoncpp.sourceforge.net/LICENSE; http://jdbc.postgresql.org/license.html; http://protobuf.googlecode.com/svn/trunk/src/google/protobuf/descriptor.proto; https://github.com/rantav/hector/blob/master/LICENSE; http://web.mit.edu/Kerberos/krb5-current/doc/mitK5license.html; http://jibx.sourceforge.net/jibx-license.html; https://github.com/lyokato/libgeohash/blob/master/LICENSE; https://github.com/hjiang/jsonxx/blob/master/LICENSE; https://code.google.com/p/lz4/; https://github.com/jedisct1/libsodium/blob/master/LICENSE; http://one-jar.sourceforge.net/index.php?page=documents&file=license; https://github.com/EsotericSoftware/kryo/blob/master/license.txt; http://www.scala-lang.org/license.html; https://github.com/tinkerpop/blueprints/blob/master/LICENSE.txt; http://gee.cs.oswego.edu/dl/classes/EDU/oswego/cs/dl/util/concurrent/intro.html; https://aws.amazon.com/asl/; https://github.com/twbs/bootstrap/blob/master/LICENSE; https://sourceforge.net/p/xmlunit/code/HEAD/tree/trunk/LICENSE.txt; https://github.com/documentcloud/underscore-contrib/blob/master/LICENSE, and https://github.com/apache/hbase/blob/master/LICENSE.txt.

This product includes software licensed under the Academic Free License (http://www.opensource.org/licenses/afl-3.0.php), the Common Development and Distribution License (http://www.opensource.org/licenses/cddl1.php) the Common Public License (http://www.opensource.org/licenses/cpl1.0.php), the Sun Binary Code License Agreement Supplemental License Terms, the BSD License (http:// www.opensource.org/licenses/bsd-license.php), the new BSD License (http://opensource.org/licenses/BSD-3-Clause), the MIT License (http://www.opensource.org/licenses/mit-license.php), the Artistic License (http://www.opensource.org/licenses/artistic-license-1.0) and the Initial Developer's Public License Version 1.0 (http://www.firebirdsql.org/en/initial-developer-s-public-license-version-1-0/).

This product includes software copyright © 2003-2006 Joe Walnes, 2006-2007 XStream Committers. All rights reserved. Permissions and limitations regarding this software are subject to terms available at http://xstream.codehaus.org/license.html. This product includes software developed by the Indiana University Extreme! Lab. For further information please visit http://www.extreme.indiana.edu/.

This product includes software Copyright (c) 2013 Frank Balluffi and Markus Moeller. All rights reserved. Permissions and limitations regarding this software are subject to terms of the MIT license.

See patents at https://www.informatica.com/legal/patents.html.

DISCLAIMER: Informatica LLC provides this documentation "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of noninfringement, merchantability, or use for a particular purpose. Informatica LLC does not warrant that this software or documentation is error free. The information provided in this software or documentation may include technical inaccuracies or typographical errors. The information in this software and documentation is subject to change at any time without notice.

NOTICES

This Informatica product (the "Software") includes certain drivers (the "DataDirect Drivers") from DataDirect Technologies, an operating company of Progress Software Corporation ("DataDirect") which are subject to the following terms and conditions:

1. THE DATADIRECT DRIVERS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.
2. IN NO EVENT WILL DATADIRECT OR ITS THIRD PARTY SUPPLIERS BE LIABLE TO THE END-USER CUSTOMER FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR OTHER DAMAGES ARISING OUT OF THE USE OF THE ODBC DRIVERS, WHETHER OR NOT INFORMED OF THE POSSIBILITIES OF DAMAGES IN ADVANCE. THESE LIMITATIONS APPLY TO ALL CAUSES OF ACTION, INCLUDING, WITHOUT LIMITATION, BREACH OF CONTRACT, BREACH OF WARRANTY, NEGLIGENCE, STRICT LIABILITY, MISREPRESENTATION AND OTHER TORTS.

Publication Date: 2018-06-08

Table of Contents

Preface 6

Informatica Resources 6

- Informatica Network 6
- Informatica Knowledge Base 6
- Informatica Documentation 6
- Informatica Product Availability Matrixes 7
- Informatica Velocity 7
- Informatica Marketplace 7
- Informatica Global Customer Support 7

Chapter 1: Introduction to the Stored Procedure Accelerator for Microsoft SQL Server 8

Stored Procedure Accelerator for Microsoft SQL Server Overview 8

Masking Stored Procedures and Table-Valued Functions 9

- Parameter Data Types 9

Stored Procedure Accelerator Components 10

Stored Procedure Accelerator Maintenance 11

Stored Procedure Accelerator Example 11

Chapter 2: Stored Procedure Accelerator Setup 13

Stored Procedure Accelerator Setup Overview 13

Step 1. Verify Requirements 13

Step 2. Create the Cleanup Procedure 14

- Creating the Cleanup Procedure 14
- Creating the Cleanup Procedure Job 14

Step 3. Create a Database Connection 16

Step 4. Create a Connection Rule 17

Step 5. Import the Security Rule Sets 18

- Importing the StoredProc Rule Set 19
- Configuring the StoredProc Rules 19
- Importing the StoredProcMasks Rule Set 21
- Configuring the StoredProcMasks Rules 22

Step 6. Define Masking Rules 23

Chapter 3: Stored Procedure Accelerator Rules 24

Stored Procedure Accelerator Rules Overview 24

Connection Rule 24

StoredProc Rule Set 25

- MatchProcNamesFolder Rule 25
- DefMaskRSSym Rule 26

MaskProcs Rule. 26
StoredProcMasks Rule Set. 27
ProcMasks1 Rule. 27
FuncMasks1 Rule. 28

Index. 29

Preface

The *Stored Procedure Accelerator Guide for Microsoft SQL Server* contains information to help administrators use the stored procedure accelerator to implement Dynamic Data Masking for stored procedures and table-valued functions in a Microsoft SQL Server database. This guide assumes that you have knowledge of Dynamic Data Masking.

Informatica Resources

Informatica Network

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

As a member, you can:

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

Informatica Knowledge Base

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

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

Informatica Documentation

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

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

Informatica Product Availability Matrixes

Product Availability Matrixes (PAMs) indicate the versions of operating systems, databases, and other types of data sources and targets that a product release supports. If you are an Informatica Network member, you can access PAMs at https://network.informatica.com/community/informatica-network/product-availability-matrices.

Informatica 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 http://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 the Stored Procedure Accelerator for Microsoft SQL Server

This chapter includes the following topics:

- Stored Procedure Accelerator for Microsoft SQL Server Overview, 8
- Masking Stored Procedures and Table-Valued Functions, 9
- Stored Procedure Accelerator Components, 10
- Stored Procedure Accelerator Maintenance, 11
- Stored Procedure Accelerator Example, 11

Stored Procedure Accelerator for Microsoft SQL Server Overview

Use the Stored Procedure Accelerator for Microsoft SQL Server to implement Dynamic Data Masking to mask stored procedures and table-valued functions in a Microsoft SQL Server database. The accelerator package contains predefined Dynamic Data Masking security rule templates and example rules.

The Stored Procedure Accelerator for Microsoft SQL Server is in the Dynamic Data Masking installation directory as an additional component that you can configure to mask stored procedures and table-valued functions. You can define masking rules based on the stored procedures and sensitive data in the Microsoft SQL Server database.

Masking Stored Procedures and Table-Valued Functions

Dynamic Data Masking uses a Java Action security rule to determine whether an incoming statement is a call to a stored procedure or a SELECT statement that includes a table-valued function that returns a result set. The Java Action rewrites the SQL statement so that the masking rules can use the result set structures.

If the stored routine is a table-valued function, the Rule Engine rewrites the SQL statement so that the Java Action receives the altered statement.

To mask stored procedure outputs, Dynamic Data Masking dynamically creates tables within the Microsoft SQL Server TempDb database based on the structure of the result set. It then populates the table with data in the TempDb database. Dynamic Data Masking names the tables in the TempDb database in the following way:

```
DDM_<Procedure Name>_<unique long number for call>_<result set sequence number>
```

For example, a table in the TempDb database might have the following name:

```
DDM_Proc_Dept_Emp_1381918459930_1
```

The unique long number for call and the result set sequence number are automatically generated numbers.

For each result set that the procedure returns, Dynamic Data Masking creates a new table in the TempDb database and populates the table with data. Dynamic Data Masking then applies masking rules to the SQL request and sends the modified SQL request to the table in the TempDb database. It then returns masked data to the client.

Parameter Data Types

The stored procedures that you mask with the Stored Procedure Accelerator for Microsoft SQL Server must have parameters with recognized data types.

You can use the accelerator to mask stored procedures that contain parameters with the following data types:

- bigint
- bit
- char
- date
- datetime
- datetime2
- decimal
- float
- int
- money
- nchar
- ntext
- numeric
- nvarchar
- nvarchar(max)

- real
- smalldatetime
- smallint
- smallmoney
- text
- time
- tinyint
- varchar
- varchar(max)
- xml

You cannot mask stored procedures that contain parameters with the following data types:

- binary
- datetimeoffset
- geography
- geometry
- hierarchyid
- image
- sql_variant
- timestamp
- uniqueidentifier
- varbinary
- varbinary(max)

Stored Procedure Accelerator Components

The accelerator contains .jar files and four security rule sets.

You can find the accelerator in the following directory:

```
<Dynamic Data Masking installation>\Accelerators\StoredProcedureMasking
```

Dynamic Data Masking uses the .jar files, MultipleStatementMasking.jar and StoredProcedureMasking.jar, in the Java Action security rule to mask stored procedures and table-valued functions.

You must configure two security rule sets to use the accelerator. The accelerator contains template rule sets and sample HR security rule sets.

Stored Procedure Accelerator Maintenance

Remove tables in the temporary database or use the cleanup script template to create a job that drops the temporary objects that the accelerator creates in the Microsoft SQL Server TempDb database as part of the masking process.

To mask stored procedures, Dynamic Data Masking dynamically creates tables in the Microsoft SQL Server TempDb database. After the Rule Engine applies masking rules to the SQL statement, it sends the request to the tables in the TempDb database.

When you set up the accelerator, you can create the CLEANUP_TEMP_OBJECTS procedure and schedule the procedure job to drop the tables. If you do not create the procedure, you must periodically manually drop the tables in the Microsoft SQL Server TempDb database.

Stored Procedure Accelerator Example

You want to mask data called by a stored procedure.

You have a stored procedure named Proc_Dept_Emp. The stored procedure calls the following columns in the Dept_Emp table:

- EMPLOYEE_ID
- Last_Name
- FIRST_NAME
- JOB_ID
- Job_Title
- SALARY
- COMM
- DEPARTMENT_ID

You set up the stored procedure accelerator and create a masking rule set. You have a masking rule that masks the Proc_Dept_Emp stored procedure. The following table describes the columns and masking functions that you define in the masking rule:

Column	Masking Function
.*SALARY	999
.*COMM	substring(convert(varchar,\(col),0),1,1)
.*Last_Name	substring(\(col),1,2)

Note: When you create the masking rule, you must enter the column name preceded by .*, as in the table.

You send the following request to the database:

```
EXEC Proc_Dept_Emp @manager_id = 25 , @dept_id = 10
```

The Dynamic Data Masking Java Action converts the request to the following statement:

```
SELECT EMPLOYEE_ID,
Last_Name,FIRST_NAME,JOB_ID,Job_Title,SALARY,COMM,
DEPARTMENT_ID
FROM DDM_Proc_Dept_Emp_1381918459930_1
```

Note: The name of the table is the table that Dynamic Data Masking created in the TempDb database.

The masking rule then changes the request to the following statement:

```
SELECT EMPLOYEE_ID,
SUBSTRING(Last_Name,1,2) AS Last_Name, FIRST_NAME,JOB_ID,Job_Title,999 AS SALARY,
substring(convert(varchar,COMM,0),1,1) AS COMM,
DEPARTMENT_ID
FROM DDM_Proc_Dept_Emp_1381918459930_1
```

Dynamic Data Masking sends the modified statement to the TempDb database table and returns masked results.

CHAPTER 2

Stored Procedure Accelerator Setup

This chapter includes the following topics:

- Stored Procedure Accelerator Setup Overview, 13
- Step 1. Verify Requirements, 13
- Step 2. Create the Cleanup Procedure, 14
- Step 3. Create a Database Connection, 16
- Step 4. Create a Connection Rule, 17
- Step 5. Import the Security Rule Sets, 18
- Step 6. Define Masking Rules, 23

Stored Procedure Accelerator Setup Overview

Set up the Stored Procedure Accelerator for Microsoft SQL Server to mask stored procedure and table-valued function result sets in Microsoft SQL Server.

To set up the accelerator, perform the following tasks:

1. Verify the setup requirements.
2. Create the cleanup procedure and schedule the cleanup procedure job.
3. Create a Microsoft SQL Server database connection.
4. Create a connection rule.
5. Import the accelerator security rule sets and configure the rules.
6. Define masking rules for stored procedures and table-valued functions.

Step 1. Verify Requirements

Verify the following requirements before you use the Stored Procedure Accelerator for Microsoft SQL Server:

- You must have Dynamic Data Masking version 9.6.0 or later installed.

- You must have a Microsoft SQL Server database that contains stored procedures or table-valued functions. If the database does not contain a stored procedure, you can set up the accelerator and create rules, but the rules will not return a match and will not mask data.

Step 2. Create the Cleanup Procedure

To conserve space in the database, use the job creation template to create a Microsoft SQL Server job that runs periodically.

To mask stored procedures, Dynamic Data Masking dynamically creates tables in the Microsoft SQL Server TempDb database. After the Rule Engine applies masking rules to the SQL statement, it sends the request to the tables in the TempDb database. If you do not run the clean up procedure, you must manually drop the tables in the temporary TempDb database.

Creating the Cleanup Procedure

Create the CLEANUP_TEMP_OBJECTS procedure.

1. Open the following file in a text editor such as Notepad:

   ```
   <Dynamic Data Masking installation>\Accelerators\StoredProcedureMasking\sql\SQLServer
   \CLEANUP_TEMP_OBJECTS_TEMPLATE.sql
   ```

2. Edit the following attributes in the CLEANUP_TEMP_OBJECTS_TEMPLATE.sql file:

   DDM_Database_name

   The name of the database in the user environment in which you want to create the stored procedure.

   ddmadmin_schema

   The name of the schema in the user environment in which you want to create the stored procedure.

3. Log in to the database as an administrator.
4. Run the edited CLEANUP_TEMP_OBJECTS_TEMPLATE.sql file to create the CLEANUP_TEMP_OBJECTS procedure.

After you create the cleanup procedure, you can schedule a job to run the procedure. If you want to run the procedure manually, you must grant the user that runs the procedure the required permissions. You must map the user to the TEMPDB database and grant the ALTER dbo schema privilege. Grant the user the following permissions:

```
use tempdb
go
CREATE USER <user name>
FOR LOGIN <user name> WITH DEFAULT_SCHEMA=dbo
go
GRANT ALTER ON Schema ::  dbo to <user name>
go
```

Creating the Cleanup Procedure Job

Schedule the cleanup procedure job.

1. Open the following file in a text editor such as Notepad:

   ```
   <Dynamic Data Masking installation>\Accelerators\StoredProcedureMasking\sql\SQLServer
   \CLEAN_TEMP_OBJECTS_JOB_CREATION_TEMPLATE.sql
   ```

2. Search the text for the following keywords and edit the property values based on the user and client requirements:

   SQLServer_Job_Name

   The name of the job.

   JOB_OWNER_USER_LOGIN_ID

   The database username.

   DDMADMIN_DATABASE

   The Dynamic Data Masking database name.

   DDMADMIN_SCHEMA

   The name of the schema in which you want to run the stored procedure.

   CLEANUP_STORED_PROCEDURE_NAME

   The name of the stored procedure. Enter CLEANUP_TEMP_OBJECTS.

   CLEANUP_SCHEDULE_NAME

   The name of the schedule.

   The following properties are specific to SQL Server job scheduling:

   - @freq_type=< freq_type>
   - @freq_interval=<freq_interval>
   - @freq_subday_type=<freq_subday_type>
   - @freq_subday_interval=<freq_subday_interval>
   - @freq_relative_interval=<freq_relative_interval>
   - @freq_recurrence_factor=<freq_recurrence_factor>
   - @active_start_date=<active_start_date>
   - @active_end_date=<active_end_date>
   - @active_start_time=<active_start_time>
   - @active_end_time=<active_end_time>

   The following text is an example of parameter values that schedule the job to run every minute:

   ```
   @freq_type=4,
   @freq_interval=1,
   @freq_interval=1,
   @freq_subday_type=4,
   @freq_subday_interval=1,
   @freq_relative_interval=0,
   @freq_recurrence_factor=0,
   @active_start_date=20160225,
   @active_end_date=99991231,
   @active_start_time=0,
   @active_end_time=235959,
   ```

   For more information about parameter options, refer to the Microsoft documentation at the following URL: https://technet.microsoft.com/en-us/library/aa259582(v=sql.80).aspx
3. Log in to the database as an administrator.
4. Run the edited CLEAN_TEMP_OBJECTS_JOB_CREATION_TEMPLATE.sql file to create the job.

Step 3. Create a Database Connection

Add the Dynamic Data Masking service for Microsoft SQL Server in the Management Console and connect to the database.

1. Log in to the Dynamic Data Masking Management Console.
2. Select the Dynamic Data Masking Server in the Management Console tree and click **Tree** > **Add DDM Services**.

   The **Add DDM Services** window appears.
3. Select DDM for SQL Server and click **OK.**

   The DDM for SQL Server node appears in the Management Console tree.
4. Select the Management Console tree root node and click **Tree** > **Add Database**.

   The **Add Database** window appears.
5. Select the Microsoft SQL Server database type and configure the database connection parameters.

   The following table describes the connection parameters:

Parameters	Description
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 Port	TCP/IP listener port for the Microsoft SQL Server database.
Optional Parameters	Additional parameters for the Informatica driver for Microsoft SQL Server.
Database Name	Optional name of a Microsoft SQL Server database. Used for manual upgrade of Dynamic Data Masking for Microsoft SQL Server service.
Rule Set	Optional rule set you can define for a specific database in addition to the Default Rule Set. Used for manual upgrade of Dynamic Data Masking for Microsoft SQL Server service.
DDM Port	Dynamic Data Masking listener port. Use DDM Port for a manual upgrade of the Dynamic Data Masking for Microsoft SQL Server service.
Default Rule Set	Use for manual upgrade of Dynamic Data Masking for Microsoft SQL Server service. Leave blank to create a Microsoft SQL Server database connection.
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.

6. Click **Test Connection** and verify that Dynamic Data Masking is connected to the database.
7. Click **OK**.

   The database node appears in the Management Console tree.

Step 4. Create a Connection Rule

Create a connection rule that directs SQL requests to the StoredProcTmpl rule set.

1. Select the Dynamic Data Masking service node for Microsoft SQL Server that you created in the Management Console tree and click **Tree** > **Connection Rules**.

   The **Rule Editor** opens.

2. In the **Rule Editor**, select the DDM for SQL Server node in the tree and select **Action** > **Append Rule**.

   The **Append Rule** window opens.

3. In the **Append Rule** window, configure the following parameters:

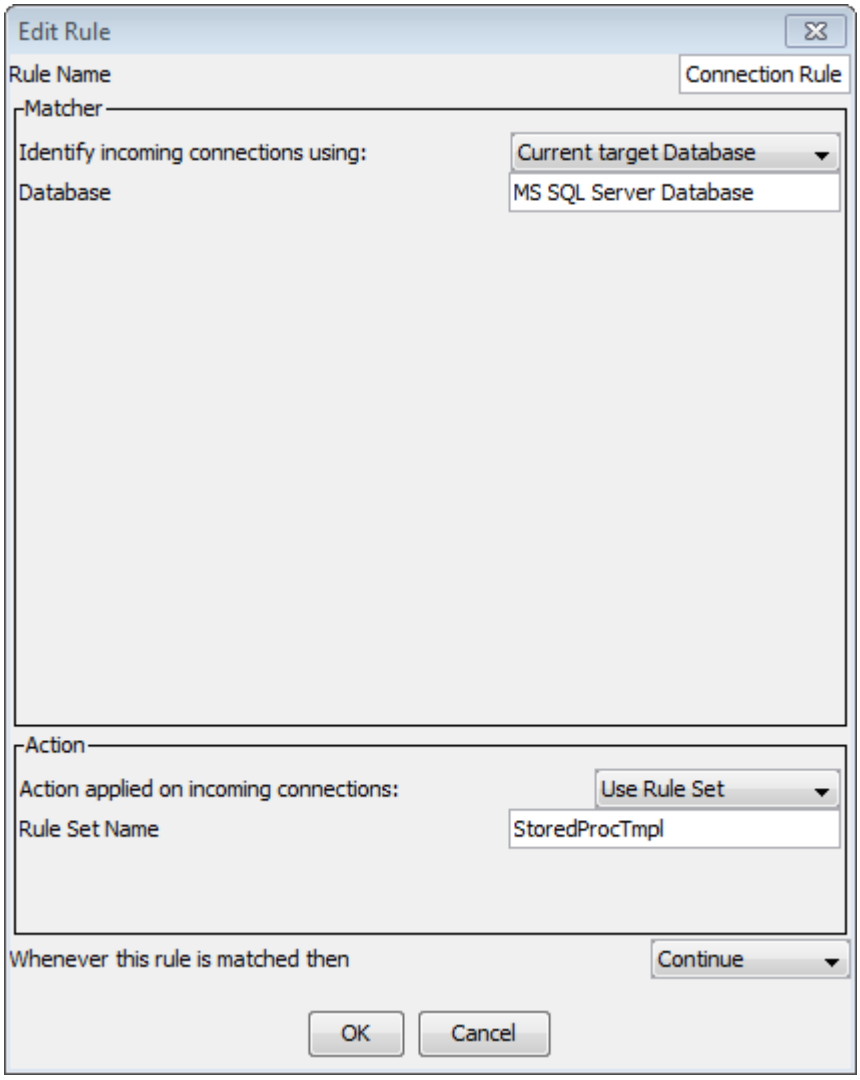
Parameter	Description
Rule Name	The logical name of the rule that appears in the rule tree. Enter the name of the connection rule.
Rule Matcher	Defines the criteria that the Rule Engine uses to identify a match. Select Current Target Database.
Database	Defines the target database name. Enter the name of the database.
Rule Action	Defines the action that the Rule Engine applies to incoming connections. Select Use Rule Set.
Rule Set Name	Defines the rule set that the Use Rule Set rule action sends the request to. Enter the name of the rule set that you want to direct the request to. To use the template rule sets, enter StoredProcTmpl. The Rule Set Name that you enter is the name of the rule set that contains the Define Symbol and Java Action security rules. If you want to use the HR example rule set, enter HRStoreProcRS. If you want to create your own rule set, enter the name of the rule set that you create.
Processing Action	Determines how the Rule Engine processes the connection after the Rule Engine applies the connection rule. Select Continue.

4. Click **OK**.

   The rule appears in the **Rule Editor**.

5. Select **File** > **Update Rules** to save the connection rule.
6. Select **File** > **Exit** to close the **Rule Editor**.

The following image shows the connection rule:



Step 5. Import the Security Rule Sets

Import the security rule sets and configure the rules to mask stored procedures and table-valued functions.

You can import the security rule set templates and alter them based on the stored procedures and table-valued functions in the database. You can import the HR rule sets to use as an example.

When you import the rule set that contains the masking rules, you must alter the rules based on the columns that the stored procedure or table-valued function accesses. Create a separate security rule in the rule set for each stored procedure or table-valued function.

Importing the StoredProc Rule Set

Import the StoredProcTmpl security rule set into the Management Console.

1. Select the Management Console tree root node and click **Tree** > **Add Rule Set**.

   The **Add Rule Set** window opens.
2. Enter StoredProcTmpl as the rule set name and click **OK**.

   The StoredProcTmpl rule set node appears in the Management Console tree.

   Note: If you want to use the HR example rule sets and you directed the connection rule to the HRStoredProcRS rule set, you must enter HRStoredProcRS as the rule set name.
3. Select the StoredProcTmpl node and click **Tree** > **Security Rule Set**.

   The **Rule Editor** opens.
4. In the **Rule Editor**, click **Action** > **Import**.

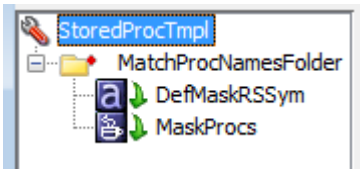
   The **Import** window opens.
5. Navigate to the following directory:

   ```
   <Dynamic Data Masking installation>\Accelerators\StoredProcedureMasking\rules\SQLServer
   ```
6. Select the SQLServer_StoredProcRSTmpl.xml file and click **Import**.

   The MatchProcNamesFolder rule folder appears in the **Rule Editor**.

   Note: If you want to use the HR example rule sets, select the SQLServer_HRStoredProcRS.xml file.
7. Expand the MatchProcNamesFolder rule folder to view the DefMaskRSSym and MasProcs rules.

   The following image shows the security rule set rules:



8. Click **File** > **Update Rules** to save the security rules.

Configuring the StoredProc Rules

Configure the StoredProc rules in the **Rule Editor**.

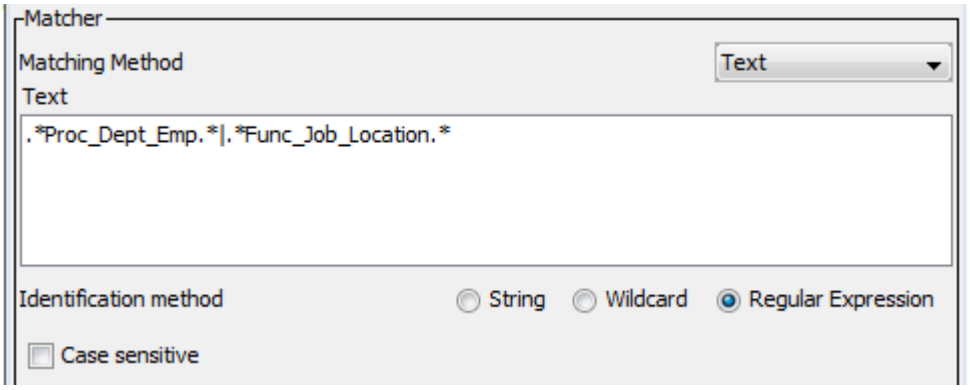
1. Select the MatchProcNamesFolder rule and click **Action** > **Edit**.

   The **Edit Rule** window opens.
2. In the Text field of the matcher, replace "proc name 1," "proc name 2," and "proc name 3" with stored procedure or table-valued function names.

   For example, you might enter the following text in the Text field:

   ```
   .*Proc_Dept_Emp.*|.*Func_Job_Location.*
   ```

   The following image shows the MatchProcsNamesFolder rule matcher parameters:



3. Click **OK**.

   The **Rule Editor** closes.

4. Click **File** > **Update Rules** to save the security rule.
5. Select the DefMaskRSSym rule and click **Action** > **Edit**.

   The **Edit Rule** window opens.

6. In the Value field of the rule action, enter the name of the rule set that contains the masking rules. For the template rule sets, enter StoredProcMasksTmpl. If you want to use the HR example rule set, enter HRStoredProcMasksRS.

   Note: The Symbol Name must be ExplicitRuleSet.

   The following image shows the DefMaskRSSym rule action properties:

7. Click **OK**.

   The **Rule Editor** closes.

8. Click **File** > **Update Rules** to save the security rule.
9. Select the MaskProcs rule and click **Action** > **Edit**.

   The **Edit Rule** window opens.

10. In the rule action Class Path field, enter the file path of the stored procedure accelerator .jar files, separated by a semicolon (;).

    You can find the .jar files in the following directory:

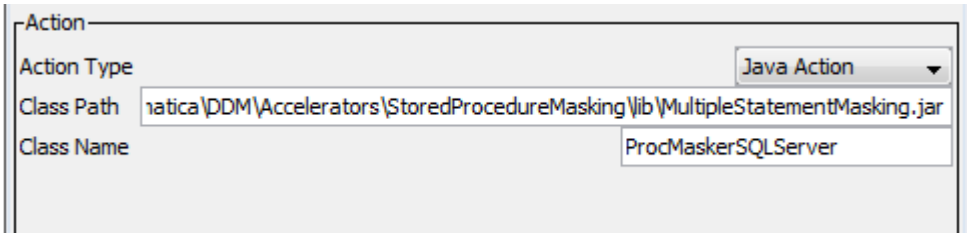
    ```
    <Dynamic Data Masking installation>\Accelerators\StoredProcedureMasking\lib
    ```

For example, you might enter:

```
C:\Program Files\Informatica\DDM\Accelerators\StoredProcedureMasking\lib
\StoredProcedureMasking.jar;C:\Program Files\Informatica\DDM\Accelerators
\StoredProcedureMasking\lib\MultipleStatementMasking.jar
```

Note: The Class Name must be ProcMaskerSQLServer.

The following image shows the MaskProcs rule action parameters:



11. Click **OK**.

    The **Rule Editor** closes.
12. Click **File** > **Update Rules** to save the security rules.
13. Click **File** > **Exit** to close the **Rule Editor**.

Importing the StoredProcMasks Rule Set

Import the StoredProcMasks masking rule set into the Management Console.

1. Select the Management Console tree root node and click **Tree** > **Add Rule Set**.

   The **Add Rule Set** window opens.
2. Enter StoredProcMasksTmpl as the rule set name and click **OK**.

   The StoredProcMasksTmpl rule set node appears in the Management Console tree.

   Note: If you want to use the HR example rule sets, enter HRStoredProcMasksRS as the rule set name.
3. Select the StoredProcMasksTmpl node and click **Tree** > **Security Rule Set**.

   The **Rule Editor** opens.
4. In the **Rule Editor**, click **Action** > **Import**.

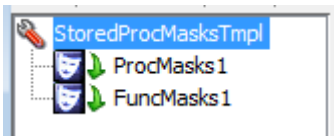
   The **Import** window opens.
5. Navigate to the following directory:

   ```
   <Dynamic Data Masking installation>\Accelerators\StoredProcedureMasking\rules\SQLServer
   ```
6. Select the SQLServer_StoredProcMasksRSTmpl.xml file and click **Import**.

   The ProcMasks1 and FuncMasks1 rules appear in the **Rule Editor**.

   Note: If you want to use the HR example rules, select the SQLServer_HRStoredProcMasksRS.xml file.

   The following image shows the StoredProcMasksTmpl rules:



7. Click **File** > **Update Rules** to save the security rules.

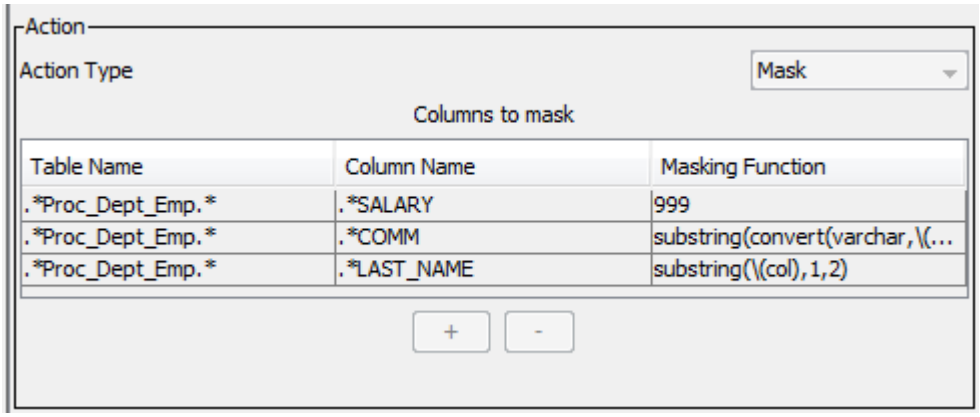
Configuring the StoredProcMasks Rules

Configure the StoredProcMasks rules in the Rule Editor.

1. Select the ProcMasks1 rule and click **Action** > **Edit**.

   The **Edit Rule** window opens.
2. Edit the rule properties based on the stored procedure that you want to mask. For example, the HR rule set has a MaskProcDeptEmp rule with masking functions defined for each column.

   The following image shows the rule action parameters in the HR MaskProcDeptEmp example rule:

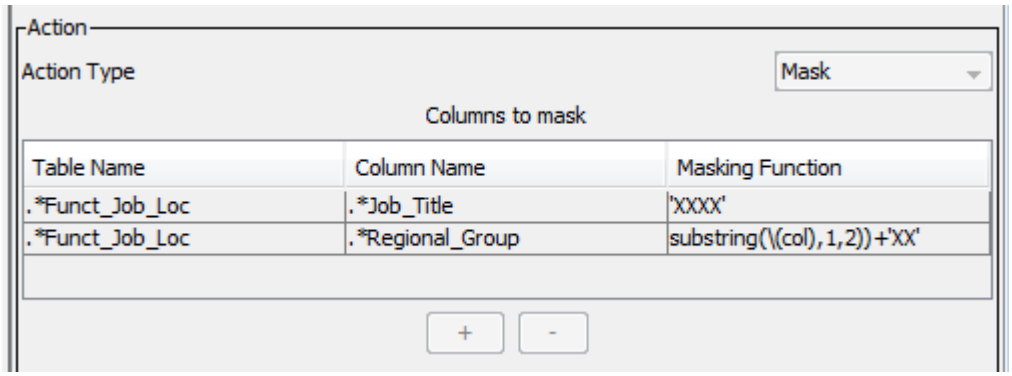


3. Click **OK**.

   The **Rule Editor** closes.
4. Click **File** > **Update Rules** to save the security rule.
5. Select the FuncMasks1 rule and click **Action** > **Edit**.

   The **Edit Rule** window opens.
6. Edit the rule properties based on the table-valued function that you want to mask. For example, the HR rule set has a MaskFuncJobLoc rule with masking functions defined for each column.

   The following image shows the rule action parameters in the HR MaskFuncJobLoc example rule:



7. Click **OK**.

   The **Rule Editor** closes.
8. Click **File** > **Update Rules** to save the security rules.
9. Click **File** > **Exit** to close the **Rule Editor**.

Step 6. Define Masking Rules

Define masking rules for each stored procedure and table-valued function.

For each stored procedure and table-valued function in the database, create a masking rule in the StoredProcMasks rule set. In the rule, you can add a row in the Mask rule action for each column that the stored procedure or table-valued function outputs. For information about masking rules, see the *Dynamic Data Masking User Guide*.

CHAPTER 3

Stored Procedure Accelerator Rules

This chapter includes the following topics:

- Stored Procedure Accelerator Rules Overview, 24
- Connection Rule, 24
- StoredProc Rule Set, 25
- StoredProcMasks Rule Set, 27

Stored Procedure Accelerator Rules Overview

The Stored Procedure Accelerator for Microsoft SQL Server contains template rules and sample rules that you can edit or copy to create rules based on the stored procedures and table-valued functions in the database.

To use the accelerator, you must create a connection rule and two rule sets. The connection rule directs requests to the first rule set. The first rule set contains rules to set up the accelerator. The second rule set contains masking rules. You can import the accelerator rule sets and configure the rules based on the stored routines in the database.

You can find the .xml files that contain the accelerator rule sets in the following location:

```
<Dynamic Data Masking installation>\Accelerators\StoredProcedureMasking\rules\SQLServer
```

The SQLServer_StoredPRocMasksRSTmpl.xml and SQLServer_StoredProcRSTmpl.xml files contain the template rule sets. You can import the template rule sets and configure the rules to mask stored procedures and table-valued functions.

The SQLServer_HRStoredProcRS.xml and SQLServer_HRStoredPRocMasksRS.xml files contain the example HR rule sets. You can view the example rules to see how a complete rule set looks.

Connection Rule

A Dynamic Data Masking connection rule directs the SQL request to the StoredProcTmpl security rule set.

You must create a connection rule in the Dynamic Data Masking Management Console to use the accelerator. Configure the connection rule to identify the incoming connection. For example, you can identify

the incoming connection by the database type. Select the Use Rule Set action and define the StoredProcTmpl rule set name or the name of the rule set that you create that contains the accelerator setup rules. Select the Stop if Applied processing action. If the client makes a request to the database, the Rule Engine applies the rule set.

StoredProc Rule Set

The StoredProc rule set contains three rules that identify requests, define the ExplicitRuleSet symbol, and call the accelerator .jar files.

The StoredProc template rule set is the SQLServer_StoredPRocRSTmpl.xml file in the accelerator directory. To use the template rules, create a rule set with the name StoredProcTmpl and import the rules into the rule set.

The StoredProc example HR rule set is the SQLServer_HRStoredProcRS.xml file in the accelerator directory. To use the example rules, create a rule set with the name HRStoredProcRS and import the rules into the rule set.

You can also create a rule set with any name and create rules in the rule set based on the template rules and example rules.

MatchProcNamesFolder Rule

The MatchProcNamesFolder rule identifies requests that call stored procedures and table-valued functions and directs them through the rule set.

The first rule in the StoredProc rule set is the MatchProcNamesFolder rule. The rule identifies requests to restrict the number of requests that go through the masking rules, which improves performance. The database may have stored procedures and table-valued functions that you do not want Dynamic Data Masking to affect the results of, such as maintenance stored procedures. Configure the MatchProcNamesFolder rule so that those stored procedures are not a match for the rule.

You can use any matcher to identify incoming requests that call stored procedures or table-valued functions. For example, if the names of the stored procedures in the database begin with Proc_, you can use a regular expression to identify requests that contain Proc_.

The MatchProcNamesFolder has the following parameters:

Rule Name

The rule name is MatchProcNamesFolder.

Matcher

The rule uses the Text matcher. You can change the matcher based on how you want to identify stored procedures.

Text

You can use regular expressions in the Text parameter to identify stored procedures.

Rule Action

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

Processing Action

The rule uses the Stop if Matched action.

DefMaskRSSym Rule

The DefMaskRSSym rule defines the ExplicitRuleSet symbol and directs the request to the masking rule set.

The DefMaskRSSym rule uses the Define Symbol action to define the ExplicitRuleSet symbol value. You must use the ExplicitRuleSet symbol name.

The symbol value is the name of the rule set that contains the masking rules. For the template rules, you can enter StoredProcMasksTmpl. If you want to use the HR example rule sets, enter HRStoredProcMasksRS.

The DefMaskRSSym rule has the following properties:

Name

The name of the rule is DefMaskRSSym.

Matcher

The rule uses the Any matcher.

Rule Action

The rule uses the Define Symbol rule action. Do not change the rule action.

Symbol Name

The symbol name is ExplicitRuleSet. Do not change the symbol name.

Symbol Value

Enter the name of the rule set that contains the masking rules. For the template rules, enter StoredProcMasksTmpl. For the HR example rule set, enter HRStoredProcMasksRS.

Keep Per Session.

Enter No to create a symbol that only exists during the current SQL request.

MaskProcs Rule

The MaskProcs rule specifies the location of the stored procedure accelerator .jar files.

The MaskProcs rule uses the Any matcher so that the rule applies to all requests. It uses the Java Action rule action to identify the Stored Procedure .jar files. The Class Name must be ProcMaskerSQLServer.

The MaskProcs rule has the following parameters:

Name

The name of the rule is MaskProcs.

Matcher

The rule uses the Any matcher so that the rule applies to all requests.

Rule Action

The rule uses the Java Action rule action. Do not change the rule action.

Class Path

Enter the file paths of the accelerator .jar files, separated by a semicolon (;). You can find the .jar files in the following location:

```
<Dynamic Data Masking installation>\Accelerators\StoredProcedureMasking\lib
```

For example, you might enter:

```
C:\Program Files\Informatica\DDM\Accelerators\StoredProcedureMasking\lib
\StoredProcedureMasking.jar;C:\Program Files\Informatica\DDM\Accelerators
\StoredProcedureMasking\lib\MultipleStatementMasking.jar
```

Class Name

The class name is ProcMaskerSQLServer. Do not change the class name.

Processing Action

The rule uses the Continue processing action.

StoredProcMasks Rule Set

The StoredProcMasks rule set contains masking rules for stored procedures and table-valued functions.

You can add as many masking rules to the StoredProcMasks rule set as required. You must create a separate rule for each stored procedure and table-valued function. The predefined rule set contains two rules that show you how to mask a stored procedure and a table-valued function.

The StoredProcMasks template rule set is the SQLServer_StoredPRocMasksRSTmpl.xml file in the accelerator directory. To use the template rules, create a rule set with the name StoredProcMasksTmpl and import the rules into the rule set.

The StoredProcMasks example HR rule set is the SQLServer_HRStoredProcRS.xml file in the accelerator directory. To use the example rules, create a rule set with the name HRStoredProcMasksRS and import the rules into the rule set.

You can also create a rule set with any name and create masking rules in the rule set based on the template rules and example rules.

Note: You must identify the StoredProcMasks rule set name in the DefMaskRSSym rule when you define the symbol value. If you change the name of the StoredProcMasks rule set, update the Value field in the DefMaskRSSym rule.

ProcMasks1 Rule

The ProcMasks1 rule masks a stored procedure.

You can configure the ProcMasks1 rule to mask a stored procedure. Add a row in the rule action for each column that the procedure outputs. Create a rule for each stored procedure.

The ProcMasks1 rule has the following values:

Name

The name of the rule is ProcMasks1. You can change the name of the rule based on the name of the stored procedure.

Matcher

The rule uses the Any matcher.

Rule Action

The rule uses the Mask rule action.

Table Name

Enter the name of the stored procedure, preceded and followed by .*. For example, you might enter:

```
.*Proc_EMPLOYEE.*
```

Column Name

The name of the column that you want to mask, preceded by .*. For example, you might enter:

```
.*SALARY
```

Masking Function

The masking function that you want to use to mask the column data.

Processing Action

The rule uses the Continue processing action.

FuncMasks1 Rule

The FuncMasks1 rule masks a table-valued function.

You can configure the FuncMasks1 to mask a table-valued function. Add a row in the rule action for each column that the function outputs. Create a rule for each table-valued function.

The FuncMasks1 rule has the following values:

Name

The name of the rule is FuncMasks1. You can change the name of the rule based on the name of the table-valued function.

Matcher

The rule uses the Any matcher.

Rule Action

The rule uses the Mask rule action.

Table Name

Enter the name of the table-valued function.

Column Name

Enter the name of the column that you want to mask, preceded by .*. For example, you might enter:

```
.*SALARY
```

Masking Function

Enter the masking function that you want to use to mask the column data.

Processing Action

The rule uses the Continue processing action.

INDEX

A

accelerator
  components 10
  example 11
  maintenance 11
  masking 9
  overview 8
  rules 24
  setup 13

C

cleanup procedure
  create 14
components 10
connection rule
  creating 17

D

DefMaskRSSym
  rule 26

F

FuncMasks1
  rule 28

M

maintenance 11
masking 9
MaskProcs
  rule 26
MatchProcNamesFolder
  rule 25

P

procedure
  cleanup 14
ProcMasks1
  rule 27

R

requirements 13
rule set
  importing 18
  StoredProcMasks 27
  StoreProc 25
rules
  connection rule 24
  DefMaskRSSym 26
  FuncMasks1 28
  masking rules 23
  MaskProcs 26
  MatchProcNamesFolder 25
  overview 24
  ProcMasks1 27
  StoredProc 19
  StoredProcMasks 22

S

set up
  create the cleanup procedure 14
setup
  database connection 16
  StoredProc rule set 19
  StoredProcMasks rule set 21
  verify requirements 13
StoredProc
  configuring 19
  importing 19
  rule set 25
StoredProcMasks
  configuring 22
  importing 21
  rule set 27