Informatica

Informatica® Dynamic Data Masking
9.8.2

Stored Procedure Accelerator Guide for DB2

Informatica Dynamic Data Masking Stored Procedure Accelerator Guide for DB2
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: 2018-06-26

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 DB2. 8

Stored Procedure Accelerator for DB2 Overview. 8

Masking Stored Procedures and User-Defined Table Functions. 8

Parameter Data Types. 9

Stored Procedure Accelerator Components. 9

Stored Procedure Accelerator Maintenance. 10

Stored Procedure Accelerator Example. 10

Chapter 2: Stored Procedure Accelerator Setup. 12

Stored Procedure Accelerator Setup Overview. 12

Step 1. Verify Requirements. 13

Step 2. Add the JDBC Driver. 13

Step 3. Create the Cleanup Procedure. 13

Creating the Cleanup Procedure. 14

Creating the Cleanup Procedure Job. 14

Step 4. Create a Database Connection. 15

Step 5. Create a Connection Rule. 16

Step 6. Import the Security Rule Sets. 17

Importing the StoredProc Rule Set. 17

Configuring the StoredProc Rules. 18

Importing the StoredProcMasks Rule Set. 20

Configuring the StoredProcMasks Rules. 20

Step 7. Define Masking Rules. 22

Chapter 3: Stored Procedure Accelerator Rules. 23

Stored Procedure Accelerator Rules Overview. 23

Connection Rule. 23

StoredProc Rule Set. 24

MatchProcNamesFolder Rule. 24

DefMaskRSSym Rule. 25

MaskProcs Rule. 25
StoredProcMasks Rule Set. 26
ProcMasks1 Rule. 26
FuncMasks1 Rule. 27

Index. 28

Preface

The *Stored Procedure Accelerator Guide for DB2* contains information to help administrators use the stored procedure accelerator to implement Dynamic Data Masking for stored procedures and user-defined table functions in a DB2 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 DB2

This chapter includes the following topics:

- Stored Procedure Accelerator for DB2 Overview, 8
- Masking Stored Procedures and User-Defined Table Functions, 8
- Stored Procedure Accelerator Components, 9
- Stored Procedure Accelerator Maintenance, 10
- Stored Procedure Accelerator Example, 10

Stored Procedure Accelerator for DB2 Overview

Use the Stored Procedure Accelerator for DB2 to implement Dynamic Data Masking to mask stored procedures and user-defined table functions in a DB2 database. The accelerator package contains predefined Dynamic Data Masking security rule templates and example rules.

The Stored Procedure Accelerator for DB2 is in the Dynamic Data Masking installation directory as an additional component that you can configure to mask stored procedures and user-defined table functions. You can define masking rules based on the stored procedures and sensitive data in the DB2 database.

Masking Stored Procedures and User-Defined Table 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 user-defined table 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 user-defined table function, the Rule Engine rewrites the SQL statement so that the Java Action receives the altered statement.

If the stored object is an unqualified reference, it must either be in the client schema or on the path that is returned by the following statement from the Dynamic Data Masking administrator:

```
SELECT CURRENT_PATH FROM SYSIBM.SYSDUMMY1
```

To mask stored procedures that return result sets or that use the OUT CURSOR parameter, Dynamic Data Masking dynamically creates tables within the DB2 TempDb schema based on the structure of the result set that the stored procedure returns and the result set from the OUT CURSOR parameter of the stored procedure. It then populates the table with the result set data. Dynamic Data Masking names the tables in the TempDb schema in the following way:

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

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

```
DDM_Proc_Dept_Emp_1381918459930_1
```

In the example, the stored procedure name is Proc_Dept_Emp. The unique long number for call and the result set sequence number are automatically generated numbers.

Dynamic Data Masking then forms select queries for each table in the TempDb schema and masks the queries based on the security rules. Dynamic Data Masking then creates a temporary stored procedure in the TempDB schema with masked outputs. The client receives the results of the temporary stored procedure call.

Parameter Data Types

The stored procedures that you mask with the Stored Procedure Accelerator for DB2 must have parameters with recognized data types.

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

Constant Arguments

- date
- integer
- string

Binding Arguments

- date
- integer
- out cursor

  Note: The OUT CURSOR parameter that Dynamic Data Masking returns has the same structure as the unmasked parameter. You cannot use the OUT CURSOR parameter for UPDATE, INSERT, or DELETE operations.
- string

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 MultipleStatementMasking.jar and StoredProcedureMasking.jar files in the Java Action security rule to mask stored procedures and user-defined table 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 DB2 TempDb schema as part of the masking process.

To mask stored procedures, Dynamic Data Masking dynamically creates tables and temporary stored procedures in the DB2 TempDb schema. After the Rule Engine applies masking rules to the SQL statement, it creates a temporary stored procedure and sends the request to the temporary stored procedure.

When you set up the accelerator, you can create the CLEANUP_TEMP_OBJECTS procedure and schedule the procedure job to drop the tables and stored procedures. If you do not create the procedure, you must periodically manually drop the tables and stored procedures in the DB2 TempDb schema.

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	\(col)-FLOOR(\(col) / 7)*7
.*Last_Name	substr(\(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:

```
CALL Proc_Dept_Emp(25,10)
```

The Dynamic Data Masking Java Action executes the stored procedure and forms the SELECT statements for the result sets:

```
SELECT   EMPLOYEE_ID , LAST_NAME LAST_NAME  , FIRST_NAME , MIDDLE_NAME , JOB_ID ,
MANAGER_ID , HIREDATE , SALARY  , COMM   , DEPARTMENT_ID , EMAIL
FROM TEMPDB . DDM_Proc_Dept_Emp_1420189677318_0
```

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

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

```
SELECT   EMPLOYEE_ID , substr(LAST_NAME ,1,2) LAST_NAME  , FIRST_NAME , MIDDLE_NAME ,
JOB_ID , MANAGER_ID , HIREDATE , 999 SALARY  , COMM -FLOOR(COMM  / 7)*7 COMM  ,
DEPARTMENT_ID , EMAIL
FROM TEMPDB . DDM_Proc_Dept_Emp_1420189677318_0 ;
```

Dynamic Data Masking creates a temporary stored procedure in the TempDb schema that uses the masked SELECT statement then and sends the call to the following temporary stored procedure:

```
CALL TEMPDB.DDM_MASKED_Proc_Dept_Emp_1420189677318()
```

CHAPTER 2

Stored Procedure Accelerator Setup

This chapter includes the following topics:

- Stored Procedure Accelerator Setup Overview, 12
- Step 1. Verify Requirements, 13
- Step 2. Add the JDBC Driver, 13
- Step 3. Create the Cleanup Procedure, 13
- Step 4. Create a Database Connection, 15
- Step 5. Create a Connection Rule, 16
- Step 6. Import the Security Rule Sets, 17
- Step 7. Define Masking Rules, 22

Stored Procedure Accelerator Setup Overview

Set up the Stored Procedure Accelerator for DB2 to mask stored procedure and user-defined table function result sets in a DB2 database.

To set up the accelerator, perform the following tasks:

1. Verify the setup requirements.
2. Add the IBM JDBC driver to the Dynamic Data Masking installation directory.
3. Create the cleanup procedure and schedule the cleanup procedure job.
4. Create a DB2 database connection.
5. Create a connection rule.
6. Import the accelerator security rule sets and configure the rules.
7. Define masking rules for stored procedures and user-defined table functions.

Step 1. Verify Requirements

Verify the following requirements before you use the Stored Procedure Accelerator for DB2:

- You must have Dynamic Data Masking version 9.6.1 or later installed.
- You must have a DB2 database that contains stored procedures or user-defined table 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. Add the JDBC Driver

Add the IBM JDBC driver to the Dynamic Data Masking installation directory.

1. Copy the IBM JDBC driver and save the driver in the following location:

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

2. Open a Server Control window.
   - On Windows, run the following commands:

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

   - On Linux, run the following commands:

     ```
     - ./server stop
     - ./server start
     ```

Step 3. Create the Cleanup Procedure

To conserve space in the database, use the job creation template to create a DB2 job that runs periodically.

Dynamic Data Masking dynamically creates tables and temporary stored procedures in the DB2 TempDb schema. After the Rule Engine applies masking rules to the SQL statement, it creates a temporary stored procedure and sends the request to the temporary stored procedure. If you do not run the clean up procedure, you must manually drop the tables and stored procedures in the temporary TempDb schema.

In DB2, the administrative task scheduler is disabled by default. Before you set up the cleanup procedure, set up the administrative task scheduler. Configure the DB2_ATS_ENABLE registry variable and create the SYSTOOLSPACE table space. You can use the following command to enable the DB2_ATS_EANBLE registry variable:

```
db2set DB2_ATS_ENABLE=YES
```

Creating the Cleanup Procedure

Create the CLEANUP_TEMP_OBJECTS procedure.

1. Open the following file:

   ```
   <Dynamic Data Masking installation>\Accelerators\StoredProcedureMasking\sql
   \DB2\CLEANUP_TEMP_OBJECTS_SP.sql
   ```

2. Use an SQL client to log in to the database as an administrator.
3. Paste the text of the CLEANUP_TEMP_OBJECTS_SP.sql file into the SQL client.
4. Run the script to create the CLEANUP_TEMP_OBJECTS procedure.

The stored procedure contains the following attributes:

created_before_seconds

Objects that have a creation time that is before the value of the created_before_seconds attribute are considered for deletion. Specify the time in seconds.

tempdb_schema

The name of the temporary schema where the accelerator creates the temporary objects.

For example, if you want to delete the temporary objects in the DDMTEMPDB schema that the accelerator created more than 300 seconds ago, you would use the following attribute values:

- `created_before_seconds : 300`
- `tempdb_schema : 'DDMTEMPDB'`

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
   \DB2\CLEANUP_TEMP_TASK_TEMPLATE.sql
   ```

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

   name

   The name of the task.

   schedule

   An input argument that is the VARCHAR(1024) datatype. The argument specifies a task execution schedule at fixed points in time. If the argument is NULL, the task is not scheduled at fixed points in time. You must write the `schedule` string in Cron format.

   procedure_schema

   The name of the schema in which you created the CLEANUP_TEMP_OBJECTS stored procedure.

   procedure_input

   The arguments that you want to pass to the CLEANUP_TEMP_OBJECTS stored procedure. For example, you might enter the following text:

   ```
   'VALUES(300,''TEMPDB'')'
   ```

   For more information about parameter options, refer to the DB2 LUW documentation at the following URL:
   https://www.ibm.com/support/knowledgecenter/SSEPGG_9.7.0/com.ibm.db2.luw.sql.rtn.doc/doc/r0054371.html

3. Use an SQL client to log in to the database as an administrator.
4. Paste the text of the edited CLEANUP_TEMP_TASK_TEMPLATE.sql file into the SQL client.
5. Run the script to create the job.
6. Use the following SQL command to verify that the job was created successfully:

```
SELECT * FROM systools.admin_task_list WHERE name LIKE <name>
```

The following text is an example execution script with the edited keyword values in bold. The task performs the cleanup of temporary objects daily at 12:00 AM, with an immediate effect:

```
CALL SYSPROC.ADMIN_TASK_ADD
  ( 'DELETETEMPOBJECTS',
    NULL,
    NULL,
    NULL,
    '0 0 * * *',
    'db2admin',
    'cleanup_temp_objects',
    'VALUES(60,''TEMPDB'')',
    NULL,
    NULL )
```

Step 4. Create a Database Connection

Add the Dynamic Data Masking service for DB2 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 DB2 and click **OK.**

   The DDM for DB2 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 DB2 database type and configure the following database connection parameters:

   DDM Database Name

   The name of 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.

   Optional Parameters

   Additional parameters for the Informatica driver for DB2.

DBA Username

User name for the database user account to log in to the DB2 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 5. 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 DB2 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 DB2 node in the tree and select **Action** > **Append Rule**.

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

   Rule Name

   Enter the name of the connection rule.

   Rule Matcher

   Select the Current Target Database matcher.

   Database

   Enter the name of the DB2 database.

   Rule Action

   Select the Use Rule Set action. The Use Rule Set action send the request to a security rule set.

   Rule Set Name

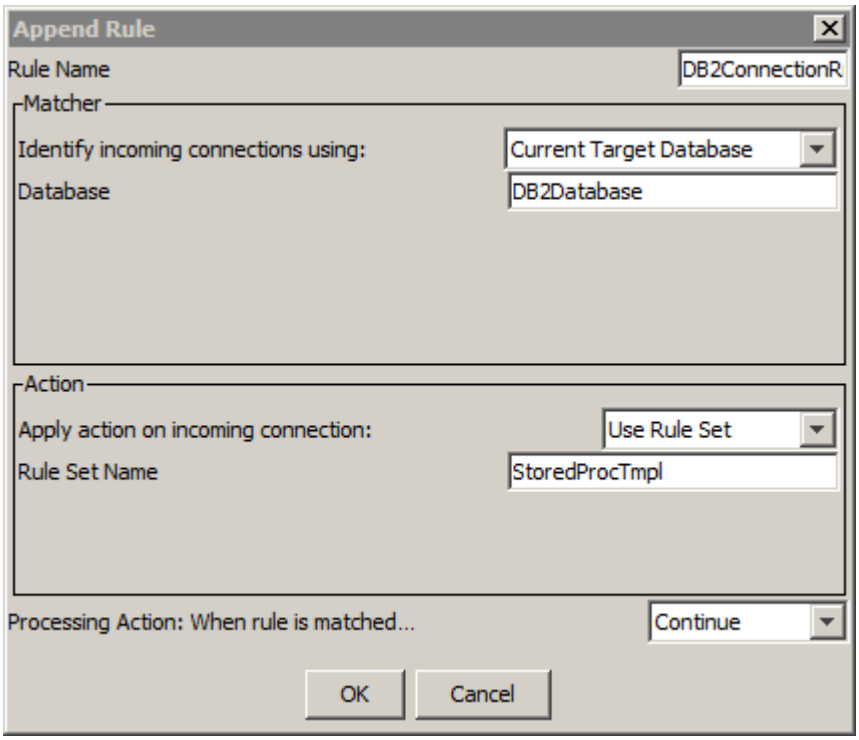
   Enter the name of the rule set that you want to direct the request to. To use the template rule set, enter StoredProcTmpl.

   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 HRStoredProcRS. If you want to create your own rule set, enter the name of the rule set that you create.

   Processing Action

   Select the Continue processing action.

   The following image shows the connection rule:



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**.

Step 6. Import the Security Rule Sets

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

You can import the security rule set templates and alter them based on the stored procedures and user-defined table 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 user-defined table function accesses. Create a separate security rule in the rule set for each stored procedure or user-defined table 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\DB2
   ```

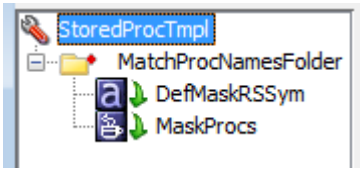
6. Select the DB2_StoredProcRSTmpl.xml file and click **Import**.

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

   Note: If you want to use the HR example rule sets, select the DB2_HRStoredProcRS.xml file.

7. Expand the MatchProcNamesFolder rule folder to view the DefMaskRSSym and MaskProcs 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**.

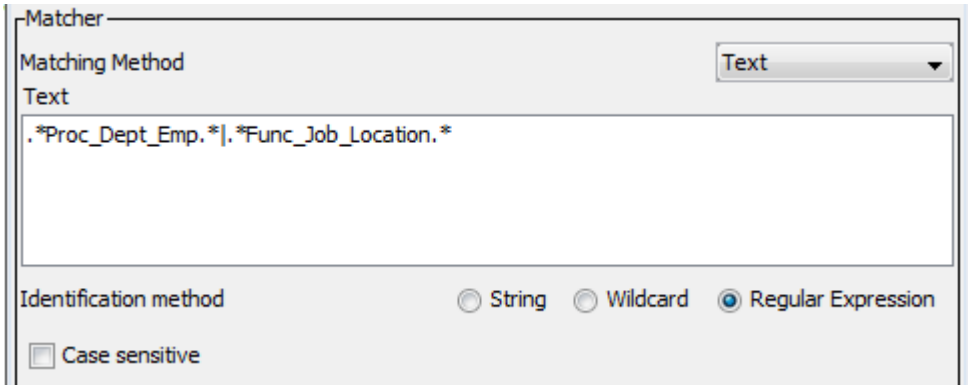
   The **Edit Rule** window opens.

2. The Text field of the matcher contains a template that you can use to enter the stored procedure names. Replace "proc name 1," "proc name 2," and "proc name 3" with stored procedure or user-defined table 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 ProcMaskerDB2.

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:

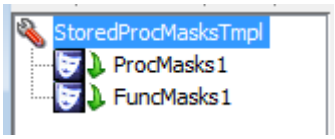
   ```
   <Dynamic Data Masking installation>\Accelerators\StoredProcedureMasking\rules\DB2
   ```

6. Select the DB2_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 DB2_HRStoredProcMasksRS.xml file.

   The following image shows the StoredProcMasksTmpl rules:

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

Configuring the StoredProcMasks Rules

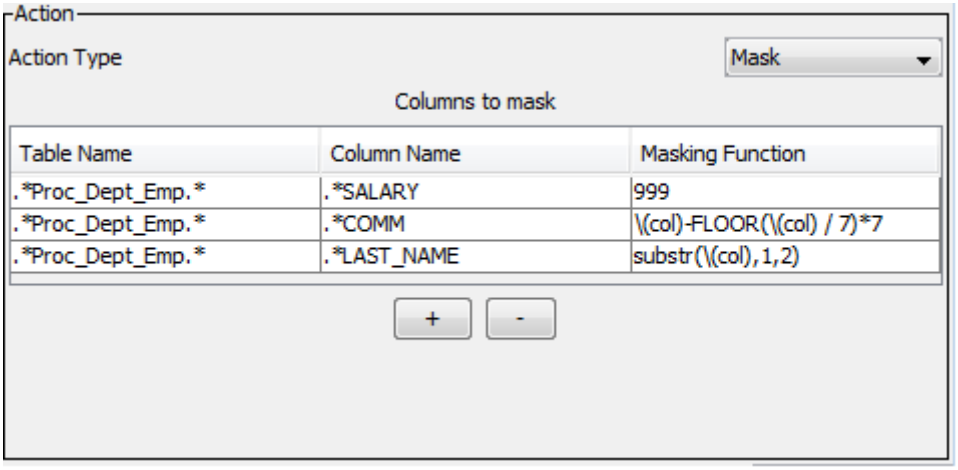
Configure the StoredProcMasks rules in the Rule Editor.

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.

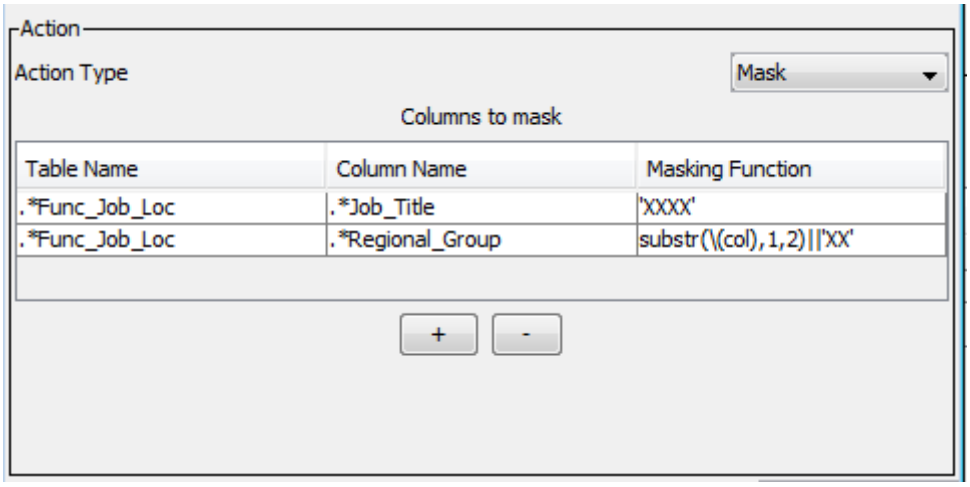
4. Click **File** > **Update Rules** to save the security rule.

5. Select the FuncMasks1 rule and click **Action** > **Edit**.

   The **Edit Rule** window opens.

6. Edit the rule properties based on the user-defined table 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 7. Define Masking Rules

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

For each stored procedure and user-defined table 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 user-defined table 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, 23
- Connection Rule, 23
- StoredProc Rule Set, 24
- StoredProcMasks Rule Set, 26

Stored Procedure Accelerator Rules Overview

The Stored Procedure Accelerator for DB2 contains template rules and sample rules that you can edit or copy to create rules based on the stored procedures and user-defined table 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\DB2
```

The DB2_StoredPRocMasksRSTmpl.xml and DB2_StoredProcRSTmpl.xml files contain the template rule sets. You can import the template rule sets and configure the rules to mask stored procedures and user-defined table functions.

The DB2_HRStoredProcRS.xml and DB2_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 DB2_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 DB2_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 user-defined table 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 might have stored procedures and user-defined table 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 user-defined table 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 ProcMaskerDB2.

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 ProcMaskerDB2. 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 user-defined table 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 user-defined table function. The predefined rule sets contain two rules that show you how to mask a stored procedure and a user-defined table function.

The StoredProcMasks template rule set is the DB2_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 DB2_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 user-defined table function.

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

Matcher

The rule uses the Any matcher.

Rule Action

The rule uses the Mask rule action.

Table Name

Enter the name of the user-defined table 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 9
- example 10
- maintenance 10
- masking 8
- overview 8
- rules 23
- setup 12

C

cleanup procedure
- create 13

components 9
connection rule
- creating 16

D

DefMaskRSSym
- rule 25

F

FuncMasks1
- rule 27

M

maintenance 10
masking 8
MaskProcs
- rule 25

MatchProcNamesFolder
- rule 24

P

procedure
- cleanup 13

ProcMasks1
- rule 26

R

requirements 13
rule set
- importing 17
- StoredProcMasks 26
- StoreProc 24

rules
- connection rule 23
- DefMaskRSSym 25
- FuncMasks1 27
- masking rules 22
- MaskProcs 25
- MatchProcNamesFolder 24
- overview 23
- ProcMasks1 26
- StoredProc 18
- StoredProcMasks 20

S

set up
- create the cleanup procedure 13

setup
- database connection 15
- StoredProc rule set 17
- StoredProcMasks rule set 20
- verify requirements 13

StoredProc
- configuring 18
- importing 17
- rule set 24

StoredProcMasks
- configuring 20
- importing 20
- rule set 26