Informatica

Informatica® Data Archive
6.5.1

Data Vault Administrator Guide

Informatica Data Archive Data Vault Administrator Guide
6.5.1
November 2021

© Copyright Informatica LLC 1996, 2021

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 © 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: 2021-12-21

Table of Contents

Preface 12

Informatica Resources 12

- Informatica Network 12
- Informatica Knowledge Base 12
- Informatica Documentation 12
- Informatica Product Availability Matrices 13
- Informatica Velocity 13
- Informatica Marketplace 13
- Informatica Global Customer Support 13

Chapter 1: Introduction to the Data Vault 14

Data Vault Overview 14

Data Vault Components 15

- Load Balancer 15
- Data Vault Loader 15
- Data Vault Agent 15
- Data Vault 15
- Data Vault Repository 15
- Data Vault Administration Tool and Command Line Program 16
- Data Vault Data Archive Plug-In 16

Architecture 16

Chapter 2: Data Vault Service Startup and Shutdown 18

Data Vault Service Startup and Shutdown Overview 18

Starting the Data Vault Components 18

- ssa_starter Syntax 19
- Startup Configuration Parameters 19

Starting the Data Vault Service 21

Starting the Data Vault Agent 21

Shutting Down the Data Vault Service and the Data Vault Agent 22

Chapter 3: Data Vault Configuration 23

Data Vault Configuration Overview 23

ssa.ini 23

- QUERY Section 24
- EXPORT Section 25
- SERVER Section 25
- AGENT Section 31
- META Section 32
- ADMIN Section 34

STARTER Section. . . . 34
Common Configuration. . . . 35
Amazon S3 Configuration. . . . 36
AWS S3 Configuration for Keyless Access. . . . 37
nucleus.ini . . . 37
CONNECTION Section. . . . 38
CLIENT Section . . . 38
Environment Variables. . . . 43
SSA_INI_DIR Environment Variable. . . . 43
SSA_STARTER_USER Environment Variable. . . . 43
NUCLEUS Environment Variable. . . . 43
Client Connections. . . . 44

Chapter 4: Data Vault SSL Setup. . . . 46

Data Vault SSL Setup Overview. . . . 46
Security Certificates. . . . 47
Generate Certificate Authority and Certificate Authority-Signed Certificates. . . . 47
Prerequisites. . . . 48
Step 1. Create the Certificate Authority Root Key and Certificate. . . . 48
Step 2. Generate a Host Certificate for the Server. . . . 49
Step 3. Sign the Certificate. . . . 50
Generate Self-Signed Certificates. . . . 51
Prerequisites. . . . 51
Step 1. Generate a Private Key. . . . 51
Step 2. Generate a Certificate Signing Request. . . . 52
Step 3. Generate a Self-Signed Certificate. . . . 52
SSL Configuration in Data Vault. . . . 52
Configuring the Ssa.ini and Nucleus.ini Files. . . . 52
Making JDBC Connections. . . . 53
Making ODBC Connections . . . 54
Using Perl Tools. . . . 55
Alternate Names. . . . 55
OpenSSL Configuration File for UNIX OS. . . . 56
OpenSSL Configuration File for Microsoft Windows OS. . . . 61
SSL Setup in Data Archive. . . . 65
Limitations. . . . 66
Troubleshooting. . . . 66

Chapter 5: Data Vault ODBC Setup 68

Data Vault ODBC Setup Overview. . . . 68
Install the Data Vault ODBC Driver Files. . . . 68
Verify the Target Connection in the Data Archive Workbench. . . . 69
Create the ODBC Data Source for the Data Vault. . . . 70

Creating an ODBC Data Source on Windows. . . . 70
Creating an ODBC Data Source on UNIX. . . . 72
ODBC Connection String Parameters. . . . 74

Chapter 6: Data Vault Administration. . . . 76

Data Vault Administration Overview. . . . 76
System Setup. . . . 77
Displaying a List of Archived Tables. . . . 77
Displaying Data Files in the Data Vault. . . . 77
Displaying Information About Files in the Data Vault. . . . 77
Displaying Information About Archived Tables. . . . 78
Setting Administration Parameters. . . . 79
Displaying Administration Parameter Values. . . . 79
Setting the Length of the Column Display. . . . 79
Configuring the Server Log. . . . 80
Enabling or Disabling the Trace Log. . . . 80
Displaying the Version Number of a Component. . . . 80
System Operations. . . . 81
Shutting Down the Load Balancer. . . . 81
Disconnecting from the Data Vault. . . . 82
Stopping an Agent Process. . . . 82
Killing an Agent Process. . . . 82
Changing the Priority Level for an Agent. . . . 82
Displaying the Current State. . . . 83
Displaying Host State. . . . 83
Displaying Agent State. . . . 83
Displaying Task State. . . . 84
Displaying Administrator State. . . . 84
Displaying Load Balancer Statistics. . . . 85
Resetting Load Balancer Statistics. . . . 85
Setting Host Preference and Priority Level. . . . 85
Removing Host Preferences. . . . 86
Displaying a List of Preferred Hosts with Priority Levels. . . . 86
Displaying Information About Metadata Database Connection. . . . 86
Metadata Administration. . . . 86
Checking Metadata Integrity. . . . 86
Remove Unreferenced Objects from the Data Vault Repository. . . . 87
Discarding Changes to the Data Vault Repository. . . . 87
Saving Changes to the Data Vault Repository. . . . 87
Displaying Column Metadata. . . . 87
Displaying Metadata Columns. . . . 88
Displaying Domain Metadata. . . . 88
Displaying Minimum and Maximum Column Values. . . . 88

Displaying the Minimum and Maximum Values for Columns in a Archived Table. 88
Accessing Catalog Tables. 88
Backup and Restore the Metadata Repository. 93
Data File Operations. 95
Checking Data File Integrity. 95
Batch Mode. 95
Data Indexing. 95
Rules and Guidelines for Data Indexing. 96
Data Indexing Example. 96
Create Index. 97
Renew Index. 98
Alter Index. 98
Drop Index. 99
Data Vault Performance Tuning. 100
Data Vault Cleanup. 101
Column Definitions. 102
Materialized Views. 103

Chapter 7: Data Repartitioning. 105

Data Repartitioning Overview. 105
Data Repartitioning Use Cases. 105
Data Repartitioning Stages. 106
Data File Calculation. 106
Data File Creation. 107
Data File Creation Example. 108
Data File Registration. 109
Data File Registration for Archived Tables. 109
Data File Registration for New Tables. 109
Data Repartitioning Log Files 110
Data Repartitioning Example. 110
Before Data Repartitioning. 110
After Data Repartitioning. 110
ssapart. 111
ssapart Example. 112

Chapter 8: Partial Data Vault Copy. 113

Partial Data Vault Copy Overview. 113
Data Export. 114
Data Export Files. 114
TAR Archive File. 114
Data File Text File 115
Data Export Log File. 115
Automatic File Transfer. 116

FTP.ini File Parameters. . . . 116
Manual File Transfer. . . . 117
ssadbcopy for Data Export. . . . 117
ssadbcopy for Data Export Examples 118
Data Import. . . . 119
Data Import Log File. . . . 119
ssadbcopy for Data Import. . . . 120
ssadbcopy for Data Import Examples. . . . 120
Prerequisites for Partial Data Vault Copy. . . . 121
Prerequisites for Source and Target Data Vault Systems. . . . 121
Prerequisites for Automatic File Transfer. . . . 121
Known Limitations. . . . 122
Steps to Partially Copy Data Between Data Vault Systems. . . . 122
Partial Data Vault Copy Example. . . . 123
Source System Configuration. . . . 123
Target System Configuration. . . . 123
Step 1. Configure the FTP.ini File. . . . 124
Step 2. Export the Data From the Source System. . . . 124
Step 3. Import the Data to the Target System. . . . 125

Chapter 9: Archived Data Migration. . . . 127

Archived Data Migration Overview. . . . 127
Archived Data Migration Process. . . . 128
Prerequisites for Archived Data Migration. . . . 128
ssamigrate Known Limitations. . . . 129
ssamigrate Syntax. . . . 130
ssamigrate Commands for Supported Target Systems 131
ssamigrate Example Scenarios. . . . 132
Archived Data Vault Migration Example. . . . 133
Source System Configuration. . . . 133
Target System Configuration. . . . 133
Step 1. Migrate the Data from the Source to the Target System 133
Step 2. Verify the Data in the Target System 134

Chapter 10: Data Validation. . . . 136

Data Validation Overview. . . . 136
Running the SSA Data Validation Utility on Servers Connected Over a Network. . . . 136
Running the SSA Data Validation Utility on Unconnected Servers. . . . 138
Additional Options for the ssadatavalidation Utility. . . . 141

Chapter 11: Bulk File Uploader. . . . 144

Bulk File Uploader Overview. . . . 144
Master XML File. . . . 145

Source Data Flat Files. . . . 146
Bulk File Uploader Components. . . . 146
Ssaservice Component. . . . 146
Ssajob Component. . . . 147
Ssamon Component. . . . 147
Addjob Component. . . . 148
Logging. . . . 149
Configuration Parameters. . . . 149
Running the Bulk File Uploader. . . . 150
Step 1. Configure the Ssa.ini File. . . . 151
Step 2. Populate the Master XML File. . . . 151
Step 3. Start Ssaservice. . . . 155
Step 4. Submit the Source Data File for the Archive Job. . . . 155
Step 5. Monitor the Job Status. . . . 156
DDL Creation and Limited Alter Table Support. . . . 156
Discovery and Compliance from Data Archive. . . . 156
Data Type Mapping. . . . 157
System Requirements. . . . 158
Known Limitations. . . . 158

Chapter 12: Data Vault Administration Tool. . . . 159

Data Vault Administration Tool Overview. . . . 159
Using the Data Vault Administration Tool. . . . 160
Starting the Data Vault Administration Tool. . . . 160
Shutting Down the Data Vault Administration Tool. . . . 160
Data Vault Administration Tool Display. . . . 160
Data Vault Administration Tool Settings. . . . 161
Toolbar. . . . 161
Keyboard Shortcuts. . . . 162
Managing the Data Vault Repository. . . . 162
Checking Metadata Integrity. . . . 162
Viewing Database Information. . . . 163
Managing the Data Vault Clients. . . . 168
Viewing Clients. . . . 168
Viewing Queries. . . . 168
Closing the Client Connections. . . . 169
Managing the Data Vault Agents. . . . 170
Viewing the Data Vault Agents. . . . 170
Viewing the Agent Hosts. . . . 170
Viewing Tasks. . . . 171
Setting the Agent Priority Level. . . . 172
Shutting Down All Data Vault Agents. . . . 172
Shutting Down a Data Vault Agent. . . . 172

Viewing Administrator Account Information. . . . 172
Running a Command Line Program. . . . 173
Shutting Down the Data Vault Service. . . . 173

Chapter 13: Data Vault Logs. . . . 174

Overview of the Data Vault Logs. . . . 174
Standard Log. . . . 174
Trace Log. . . . 175
Query Statistics Log. . . . 176
Example of a Query Statistics Log. . . . 177
Query Statistics Fields. . . . 177
Loading Query Statistics into a Database. . . . 178

Chapter 14: User Account Privileges. . . . 180

User Account Privileges Overview. . . . 180
GRANT. . . . 181
Required Privileges. . . . 181
Syntax. . . . 182
Description to GRANT. . . . 183
OWNER Privileges. . . . 183
Examples of Owner Privileges. . . . 184
REVOKE. . . . 185
Required Privileges. . . . 185
Syntax. . . . 186
Description to REVOKE. . . . 187
ALTER AUTHORIZATION. . . . 187
DBA Privileges. . . . 188
Data Vault Passwords. . . . 189
Updating Passwords in the JReport Catalog. . . . 191
DBA User Passwords. . . . 191
Prerequisites. . . . 192
Changing the DBA User Password. . . . 192

Chapter 15: ssasql Command Line Program. . . . 193

ssasql Command Line Program Overview. . . . 193
ssasql Operating Modes 193
ssasql Example. . . . 194
ssasql Arguments. . . . 194
ssasql Session Mode 196
ssasql Session Mode Commands. . . . 196
ssasql SQL Mode 198
ssasql SQL Mode Command Syntax 198
ssasql SQL Mode Commands 199

Display Formats. . . . 204
Limiting the Display of Results from a SELECT Statement. . . . 204
Date Picture Display Format. . . . 204
Time Picture Display Format. . . . 204

Chapter 16: Data Vault Audit Log. . . . 206

Data Vault Audit Log Overview . . . 206
Audit Log Process. . . . 207
Audit Log Configuration Process. . . . 207
Dynamic Data Masking Rules. . . . 208
Configuring the Data Vault SSA.ini File. . . . 208
Configuring the Data Vault Connection in the Management Console. . . . 210
Step 1. Add the Dynamic Data Masking Service for Data Vault. . . . 210
Step 2. Create a Connection to the Data Vault. . . . 210
Creating the Connection Rules. . . . 211
Step 1. Create the Connection Rule Folder. . . . 211
Step 2. Create a Rule to Switch to the Data Vault Database. . . . 212
Step 3. Create a Rule that Applies the Security Rule Set. . . . 212
Creating the Security Rules. . . . 213
Step 1. Create the Security Rule Set. . . . 213
Step 2. Create a Rule that Logs Incoming Requests to the Security Rule Set. . . . 214
Step 3. Create a Rule to Rewrite SQL Request Statements. . . . 214
Generate the Audit File. . . . 216
Sample Audit Log. . . . 217

Appendix A: Sample Configuration Files. . . . 218

ssa.ini. . . . 218
nucleus.ini. . . . 219

Index. . . . 220

Preface

Read the *Data Vault Administrator Guide* to learn how to configure and manage the data vault. The guide includes information about the Data Vault Administration Tool, user account privileges, and audit logs.

Informatica Resources

Informatica provides you with a range of product resources through the Informatica Network and other online portals. Use the resources to get the most from your Informatica products and solutions and to learn from other Informatica users and subject matter experts.

Informatica Network

The Informatica Network is the gateway to many resources, including the Informatica Knowledge Base and Informatica Global Customer Support. To enter the Informatica Network, visit https://network.informatica.com.

As an Informatica Network member, you have the following options:

- Search the Knowledge Base for product resources.
- View product availability information.
- Create and 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 find product resources such as how-to articles, best practices, video tutorials, and answers to frequently asked questions.

To search the Knowledge Base, visit https://search.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

Use the Informatica Documentation Portal to explore an extensive library of documentation for current and recent product releases. To explore the Documentation Portal, visit https://docs.informatica.com.

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

Informatica Product Availability Matrices

Product Availability Matrices (PAMs) indicate the versions of the operating systems, databases, and types of data sources and targets that a product release supports. You can browse the Informatica 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 and based on real-world experiences from hundreds of data management projects. Informatica Velocity represents the collective knowledge of Informatica consultants who work with organizations around the world to plan, develop, deploy, and maintain successful data management solutions.

You can find 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 extend and enhance your Informatica implementations. Leverage any of the hundreds of solutions from Informatica developers and partners on the Marketplace to improve your productivity and speed up time to implementation on your projects. You can find the Informatica Marketplace at https://marketplace.informatica.com.

Informatica Global Customer Support

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

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:
https://www.informatica.com/services-and-training/customer-success-services/contact-us.html.

To find online support resources on the Informatica Network, visit https://network.informatica.com and select the eSupport option.

CHAPTER 1

Introduction to the Data Vault

This chapter includes the following topics:

- Data Vault Overview, 14
- Data Vault Components, 15
- Architecture, 16

Data Vault Overview

Data Archive uses the Data Vault Service to move inactive data to a Data Vault.

When you use Data Archive, you can archive data in multiple ways. You can archive data to partitions in the same database, to another database, or to a Data Vault. The Data Vault is an immutable, highly compressed file structure that you can query using standard SQL syntax.

The Data Vault provides the following features and benefits:

- Storage efficiency, improved application performance, speed and ease of user access, and low overall cost of ownership.
- The Data Vault stores data in a fraction of the space that would be necessary using a relational format, offering order-of-magnitude savings in storage costs.
- Unlike tape backup archives, which require tape retrieval and loading as well as often complex and time-consuming searches, end users can run queries directly against the archive using the same Business Intelligence tools that they use for other analytical tasks.
- Unlike other compression technologies that require full uncompression of data before searching, you can run queries against the compressed files in Data Vault, and the result set is uncompressed.
- End users can access Data Vault directly, without the need to involve a DBA or archivist in the process.

Data can be loaded and compressed into archived table form from any relational or flat file data source. The Data Vault stores the compressed, immutable data in the form of SCT files. The compressed data can be extracted by an SQL routine from either an application script or through a query tool, loaded into any target data database, and joined there with existing tables if required.

The degree of data compression is a function of the type of data. Since tokenization is a key part of the process, the more duplication there is in the source data, the higher the compression ratio. This also implies that the greater the amount of data–and therefore the higher the probability of duplication–the higher the resulting compression ratio.

The Data Vault cannot be updated. However, once data has been retrieved from the Data Vault and loaded into another data structure, it can be manipulated in all the usual ways to perform operations such as "what-if" analyses.

Data Vault Components

The Data Vault consists of a number of components that interact to archive data and process queries to the Data Vault.

The Data Vault has the following components:

- Load Balancer
- Data Vault Loader
- Data Vault Agent
- Data Vault
- Data Vault repository
- Data Vault Administration Tool and command line program
- Data Vault Data Archive plug-In

Load Balancer

The load balancer manages client connections to the Data Vault repository and handles client requests to the Data Vault.

If you configure multiple Data Vault Agents for the Data Vault, the load balancer assigns a client request to the first available Data Vault Agent for processing.

Data Vault Loader

The Data Vault Loader loads data to the Data Vault. Data Archive calls the Data Vault Loader to move data from the source to the Data Vault.

Data Vault Agent

The load balancer calls the Data Vault Agent to handle client requests to the Data Vault.

You can configure multiple Data Vault Agents in on multiple machines. At least one agent must be running when the load balancer handles requests.

Data Vault

The Data Vault stores data in a highly compressed format. You can use ODBC- or JDBC-compliant query tools and standard SQL syntax to query the data in the Data Vault.

You can configure the Data Vault to use third-party storage platforms. You cannot modify data in the Data Vault.

Data Vault Repository

The Data Vault repository contains the Data Vault catalog. The catalog describes the tables that are archived in the Data Vault and stores the data structures for the Data Vault.

The load balancer gets information about the archived data from the Data Vault repository when it processes client requests.

Data Vault Administration Tool and Command Line Program

You use the Data Vault Administration tool or command line program to configure and manage the Data Vault.

The Data Vault Administration Tool and command line program is installed with the Data Vault. On Windows, you can install the Data Vault Administration Tool on the same machine or on a separate machine. If the Data Vault Administration Tool is on a separate machine, it uses TCP/IP to communicate with the Data Vault.

Data Vault Data Archive Plug-In

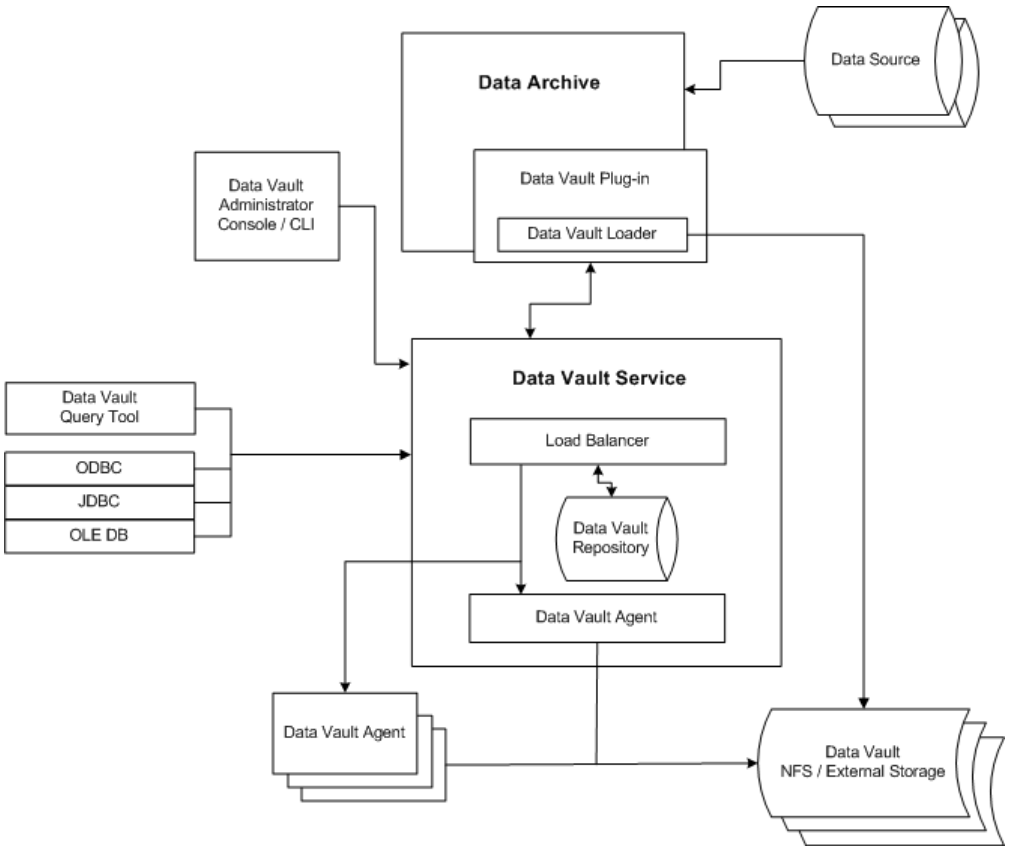
The Data Vault Data Archive plug-in adds Data Vault functionality to Data Archive.

When you install the plug-in, you can set up and run jobs in Data Archive to archive data from various sources to the Data Vault. The Data Vault Data Archive plug-in must be installed in the Data Archive directory.

Architecture

Data Archive uses the Data Vault Service to move data to the Data Vault.

The following diagram illustrates the Data Vault architecture:



Set up a job in Data Archive to archive source data to the Data Vault. When you run the archive job, Data Archive calls the Data Vault Loader to move data to the Data Vault. You can set up the Data Vault to use

third-party storage systems such as EMC Centera, Hadoop Distributed File System (HDFS), or Hitachi Content Platform (HCP).

The Data Vault load balancer generates metadata in the Data Vault repository for the archived data. The load balancer calls a Data Vault Agent to access the Data Vault and process client requests.

Use the Data Vault Administration Tool or command line program to monitor and administer the Data Vault components.

You can use ODBC- or JDBC-compliant query or business intelligence tools to access the data in the Data Vault. On Windows, you can also use the Data Vault SQL Tool that ships with the Data Vault to query the data in the Data Vault.

Use the Data Archive options and tools to access the Data Vault. For example, use the Data Validation option to verify that the data moved to the Data Vault is valid. You can also use the Data Discovery portal to search the data in the Data Vault.

CHAPTER 2

Data Vault Service Startup and Shutdown

This chapter includes the following topics:

- Data Vault Service Startup and Shutdown Overview, 18
- Starting the Data Vault Components, 18
- Starting the Data Vault Service, 21
- Starting the Data Vault Agent, 21
- Shutting Down the Data Vault Service and the Data Vault Agent, 22

Data Vault Service Startup and Shutdown Overview

When you install the Data Vault, the installer creates a startup program named *ssa_starter* to automate the startup and shutdown of the Data Vault Service and the Data Vault Agent.

The ssa_starter uses the parameters set in the STARTER section of the ssa.ini file to determine the configuration settings for the Data Vault components.

You can also start the Data Vault Service and the Data Vault Agent separately. If you start the components separately, you must start the Data Vault Service before you start the agent. At startup, each component uses the parameters in the ssa.ini file to determine the connection and configuration settings.

Starting the Data Vault Components

You can use ssa_starter startup program created during installation to start and stop the Data Vault components, and to restart the components if they fail.

On UNIX, the ssa_starter program runs as a daemon. On Windows, you can configure the ssa_starter program as a Windows service.

ssa_starter Syntax

The ssa_starter options you can use depend on the platform where you install the Data Vault.

```
ssa_starter option
```

The following table describes syntax options for Windows and UNIX:

Option	Description
-i	Installs the ssa_starter service. Only for Windows.
-u	Uninstalls the ssa_starter service. Only for Windows.
-r	Runs ssa_starter manually, not as a service.
-s	Sends SIGHUP to the running daemon, which causes it to re-read the ssa.ini file. This option should be used after making changes to the configuration. Only for UNIX.
-o	Sends the log to the standard output stream, which is usually the console.
-h	Displays the online help.

Startup Configuration Parameters

The ssa_starter startup program uses configuration settings in the [STARTER] section of the ssa.ini file.

The following table describes the parameters used by the startup service:

Parameter	Description
SERVER_CMD	The command to start the Data Vault Service. Default is ssaserver.
AGENT_CMD	The command to start the Data Vault Agent. Default is ssaagent. Do not include the -u parameter to specify a numeric ID for the agent process. The startup service assigns an agent ID.
AGENT_COUNT	The number of Data Vault Agents processes to start. Default is 2.
LOGDIR	The directory where the log files will be written. On UNIX, the daemon log is named ssa_starter. On Windows the service activities are logged by the operating system, which can be viewed through the Event Viewer.
AGENT_CONTROL	Enables (1) or disables (0) automatic restart if the Data Vault Agent fails. Default is 1.
SERVER_CONTROL	Enables (1) or disables (0) automatic restart if the Data Vault Service fails. Default is 1.
USER	UNIX only. The user ID under which all processes will be started by the ssa_starter daemon. It is usually important to set this parameter, since the default is the root user, which can cause problems. For example, if a process running as root creates a file, only another root process can access that file. This parameter can also be set using the SSA_STARTER_USER environment variable. If both the ssa.ini file and the environment variable define the user ID, the environment variable takes precedence.

Parameter	Description
VERBOSE	The level of detail for the messages written to the log files. The following are the possible options: - 0 : Includes all error messages and the most important messages - 1 : Level 0 plus more detailed messages - 2 : Level 1 plus system trace information Default is 2.
EXE[0-9]	These 10 optional parameters (EXE0, EXE1, EXE2, ..., EXE9) refer to custom commands, scripts, or executables that will run in conjunction with Data Vault Service on startup. Note that the numbering of the parameters has no relationship to the order in which they are processed. These parameters are run asynchronously, so there is no guarantee about the execution order.

The ssa_starter program reads the [STARTER] section in the ssa.ini file to determine the settings at initialization. The [STARTER] section in the ssa.ini file must specify at least the following parameters:

- SERVER_CMD. Contains the command to start the Data Vault Service.
- AGENT_CMD. Contains the command to start the Data Vault Agent.

The AGENT_COUNT parameter indicates the number of agents to start and must be set to a value of 1 or higher, otherwise no agent will be started.

The AUTOSTARTUP parameter must be set to 1 (enabled) in the [META] section of the ssa.ini file so that the startup service or daemon works properly. This allows the startup service to start the Data Vault repository database before the Data Vault Service. A related parameter, STARTUPCOMMAND, which defines the command line for starting the Data Vault repository database, must also be configured in the same section of the ssa.ini file. If the Data Vault repository database is not on the local machine, the STARTUPHOST parameter must specify the name of the remote computer where the Data Vault repository database is located.

If the Data Vault server shuts down unexpectedly, enabling AUTOSTARTUP might prevent it from restarting. This is because after a server shuts down unexpectedly, the child metadata process remains running, so that the port is busy.

To prevent this behavior, you can also manage the metadata server with ssa_starter and the EXE commands. If you do not enable AUTOSTARTUP, you must add the following EXE parameters in the [STARTER] section of the ssa.ini file:

For UNIX:

```
EXE0=fb_inet_server -m

EXE0_CONTROL=1
```

For Microsoft Windows:

```
EXE0=fb_inet_server -a

EXE0_CONTROL=1
```

These parameters allow ssa_starter to manage the metadata server process when the AUTOSTARTUP option is set to 0. The name of the metadata server process is `fb_inet_server.exe`. Formerly, this process, which runs on the application server, was called `ssaeng.exe` on Microsoft Windows and `ssaeng` on Linux.

On UNIX, if any changes are made in the ssa.ini file to parameters related to the Data Vault Service daemon, ssa_starter can be made to read the changes in the ssa.ini file using the ssa_starter -s command. On Windows, the service can be made to re-read the configuration by stopping, then restarting the service.

Starting the Data Vault Service

You can start the Data Vault Service separately. If you start the components separately, start the Data Vault Service before you start the agent.

Ensure that the configuration settings for Data Vault Service are set properly in the ssa.ini file. The startup parameters in the [META] section must point to the Data Vault repository database. You must start the Data Vault Service before you start the agent.

Start Data Vault Service with the following command:

```
ssaserver
```

On UNIX, you can use the & option to run the Data Vault Service in the background.

If the Data Vault Service has successfully connected to the Data Vault repository database, the following text will be displayed at the console:

```
META starting up........ok
COM starting up...ok
EXEC starting up....ok
SSA Server successfully started.
```

Starting the Data Vault Agent

You can start each Data Vault Agent separately. The Data Vault Service must be running when the agent starts. If you start the components separately, start the Data Vault Service before you start the agent.

When the Data Vault Agent process starts, it reads the local ssa.ini file and uses information in the [SERVER] section to connect to the Data Vault Service. If an agent is started on a remote machine, there must be an ssa.ini file on that machine containing the same HOST and EXEPORT parameters and values as the ssa.ini file on the machine that hosts the Data Vault Service. Parameters that are specific to the agent can also be found in the [AGENT] section of the ssa.ini file.

You can start the Data Vault Agent with the following command:

```
ssaagent [ -u AID ][ -p priority ]
```

where:

- *AID* is an optional user-defined numeric ID for the ssaagent process. Ssaagent processes viewed through the Data Vault Service administration command line program are identified by this Agent ID ("AID"), or if the AID was not set, by the operating system process ID (PID). Each ssaagent process started on the same machine must be given a unique AID value. A space must separate the -u flag from its value.
- *priority* specifies an optional priority level for the ssaagent process. This must be a value from 0 to 10 inclusive, where a larger number means higher priority. A priority level of 0 effectively "freezes" the agent, making it unavailable for task execution after it completes its current task (if applicable). A priority level of 1 reserves the ssaagent process for administrator tasks, such as compressed data file registration. A space must separate the -p flag from its value.

You can start as many Data Vault Agents as required. At minimum, there must be one Data Vault Agent running. On UNIX, you can use the & option to run the agent in the background.

If the agent starts up successfully, it displays a message similar to the following (server names and port numbers will vary):

```
New connection 1 to the <Host: ALPHA1, PN: 6500> has been established
LOG>Info : The agent: <host: alpha1, AID: 1368>, priority: 10 was successfully registered
```

Shutting Down the Data Vault Service and the Data Vault Agent

Use the Data Vault Administration Tool to shut down the Data Vault Service. Then run a command line program to shut down the Data Vault Agent.

To shut down the Data Vault Service, log in to the Data Vault Administration Tool and click **Action** > **Shutdown**. If the Data Vault repository database is configured to start with the Data Vault Service, it shuts down when the Data Vault Service shuts down.

To shut down the Data Vault Agent, go to the command line and run the following command:

```
Stop <HostName> <AgentID>
```

The <HostName> parameter is the host machine where the Data Vault Agent process is running, and the <AgentID> parameter refers to the numeric value that identifies the agent process.

CHAPTER 3

Data Vault Configuration

This chapter includes the following topics:

- Data Vault Configuration Overview, 23
- ssa.ini, 23
- nucleus.ini , 37
- Environment Variables, 43
- Client Connections, 44

Data Vault Configuration Overview

The Data Vault stores setup information in configuration files. When you configure the Data Vault during installation, the installer writes the settings to the configuration files. After installation, you can modify the configuration files to change the Data Vault configuration.

The Data Vault stores configuration information in the following files:

- ssa.ini
- nucleus.ini

The configuration files are organized in sections. Each section contains a set of parameters that are relevant to a component or aspect of the Data Vault.

You can also use the environment variables to store specific Data Vault settings, such as the directory where the ssa.ini file is located.

ssa.ini

The `ssa.ini` contains configuration settings for the Data Vault load balancer and agent.

The `ssa.ini` file contains the following sections:

- QUERY
- EXPORT
- MERGE
- SERVER

- AGENT
- META
- ADMIN
- STARTER
- COMMON

QUERY Section

The QUERY section of the ssa.ini file contains the parameters for queries sent to the Data Vault.

The following table describes the parameters in the QUERY section of the ssa.ini file:

Column	Description
MEMORY	The total amount of memory (in MB) to be used for query result sets. Memory is allocated dynamically up to this amount as required. Note that the specified value must be greater than 0. Default is 1024
THREADS	The number of threads used when querying the compressed data file. Default is 2.
TEMPDIR	The directory for Data Vault temporary data files. To improve performance, use a local directory. Default is <DataVaultServiceDirectory>/temp.
VFS	The directory location of a virtual file system. Instead of creating a number of temporary data files, the Data Vault Service stores the temporary files in a single file container. If this parameter is absent or empty, the Data Vault Service uses the standard file system in the location specified by the TEMPDIR parameter.
SHAREDIR	The network location where temporary files for GROUP BY and ORDER BY queries are stored. This location must be accessible to any machine that runs the Data Vault Agent. Default is <DataVaultServiceDirectory>/share.
INDEXDIR	The directory where the Data Vault stores index files when you create indexes for archived tables. Default is <DataVaultServiceDirectory>/index.

EXPORT Section

The EXPORT section of the ssa.ini file contains the parameters for data export from the Data Vault.

The following table describes the parameters in the EXPORT section of the ssa.ini file:

Column	Description
BUFFSIZE	The size (in MB) of the internal buffer used for an export operation. Default is 64.
DELIMITER	The character used to delimit columns in the exported flat file. This parameter should only be set when MODE=1 (variable-length mode). Default is | (vertical bar).
EOL	The end-of-line indicator in the exported flat file. This parameter is used only in fixed-length mode (MODE=0). In variable-length mode (MODE=1), rows are always separated by the carriage return and line feed character combination (\r\n). Default is \n.
MAXTEMPSIZE	The maximum size (in GB) for the temporary virtual file system created during the export operation. Default is 8.
MODE	The way in which the exported data is written. There are two options: 0 (fixed-length mode) and 1 (variable-length mode). In fixed-length mode, each column has a fixed width; column values less than that width are padded with blank spaces. In variable-length mode, column values are separated from each other by a delimiter (set with the DELIMITER parameter); there is no space padding. Default is 0 (fixed-length mode).
NULL	The character string that represents a null value in the exported flat file. Default is NULL.
TEMPDIR	The directory where a temporary virtual file system will be created during the export operation. Default is <DataVaultServiceDirectory>/temp.

SERVER Section

The SERVER section of the ssa.ini file contains the configuration parameters for the Data Vault Service. It also contains configuration parameters for log files.

The following table describes the [SERVER] section startup parameters for the load balancer:

Column	Description
HOST	The name of the machine on which Data Vault Service load balancer is running.
PORT	Port number used by the ssasql command line program and other clients such as the Data Vault SQL Tool and ODBC applications to connect to the Data Vault. Default is 8500.
EXEPORT	Port number used by the Data Vault Agent and the Data Vault Administration Tool to connect to the Data Vault. Default is 8600.

The following table describes the optional parameters for the load balancer. The default value is used if the parameter is missing or empty, and there is no [COMMON] section value:

Column	Description
TEMPDIR	The directory where Data Vault Service temporary files (vfs*xxxxx*.tmp) are created. Default value is <DataVaultServiceDirectory>/temp.
SHAREDIR	The network location where temporary files for GROUP BY and ORDER BY queries are stored. This location must be accessible to all machines that have Data Vault Agent processes running. Default value is <DataVaultServiceDirectory>/share.
INDEXDIR	The directory where the Data Vault stores index files when you create indexes for archived tables. Default is <DataVaultServiceDirectory>/index.
BULKLOADDIR	The directory where the compressed data files generated through bulkload operations are created. Default is **.** (current directory).
THREADS	The number of threads used by load balancer to handle client requests. Default. 10 Minimum. 1 Maximum. (depends on machine resources)
ROWSETBUFFER	The size of the result set row buffer (in KB). This value should be at least the size of a fetched row. If the value is set to less than the size of a fetched row, an error message will be returned. Note also that if the buffer is too large, memory errors can occur. Default. 64 Minimum. (size of a fetched row) Maximum. (depends on machine resources)
BLOCKFETCHBUFFER	The size of the array (in KB) for fetching multiple records when the ODBC block fetching option is enabled. Increasing the size of this buffer can often improve performance. Default. 64 Minimum. 64 Maximum. (depends on machine resources)
ERHF	Determines how expression evaluation errors are handled for archived table queries. The options are: - STOP. Query execution is halted when the first error condition is encountered; an error message is returned to the client that submitted the query. - REJECT. The entire compressed data file record that produced the error is rejected. - REPLACE. Each value that generates an error is replaced with a null in the result set. If the client application supports ODBC-level warning messages, a numerical summary of rejected or replaced values will be displayed at the end of the query results, but only in cases where the number (*n*) is greater than 0: *n* rejected rows -or- *n* replaced values Default is STOP.

The following table describes the optional parameters that relate to log file configuration:

Column	Description
VERBOSE	Enables (1) or disables (0) the writing of extra details to the Data Vault Service log files. These extra details include the list of Data Vault data files affected by a query (by default, this list is limited to the first 4 KB of textual data), as well as debug information. Default. 0 (off) Note that the verbose log can also be turned on by executing the Data Vault Service Administration console command "SET DETAILED_LOG ON."
LOGDIR	The directory where the Data Vault Service log files are stored. Default: **.** (current directory)
LOG	Enables or disables the standard log file. The possible values are ON or ENABLE to turn on standard logging; OFF or DISABLE to turn off standard logging. By default, standard logging is enabled. Default. ON (or ENABLE)
LOGVERBOSE	Sets the level of detail that will be written to the standard log file. The following levels are available: 0 or LOW. all errors and other important messages (the default level) 1 or HIGH. level **0** + more detailed messages 2 or TRACE. level **1** + tracing messages Note that the higher the logging level, the larger the log file will be, though the log file can be limited to a particular size (LOGLIMIT_FILESIZE) or amount of time (LOGLIMIT_TIME). Also note that level **2** produces the same information as the independent trace log (TRACELOG). This trace information is intended mainly for internal use by Informatica Global Customer Support, and may not have much practical value to end users. Default: 0 (or LOW)
LOGLIMIT_FILESIZE	Sets the maximum size (in bytes) for the standard log file. Once the log file grows to this size, one of two possibilities will occur: either new log information will "push out" the oldest log entries, keeping the log file below or at the specified maximum size; or, if LOGLIMIT_MAXFILES is set to a value greater than **1**, a new log file will be created to store the latest log information, leaving the previous log file intact. Setting LOGLIMIT_FILESIZE to **0** is equivalent to omitting this parameter from the ssa.ini file. The LOGLIMIT_TIME parameter will be used instead, unless that parameter is also missing, in which case the default value for LOGLIMIT_FILESIZE will be used to limit the log file. The LOGLIMIT_FILESIZE parameter takes precedence over LOGLIMIT_TIME, so if both are set in the ssa.ini file, the log file will be limited by size rather than time period. Default is 1048576 (1 MB).

Column	Description
LOGLIMIT_TIME	Sets a time limit on the standard log file. When this option is enabled, entries are added to the current log file until the timestamp interval between the first message and the latest one exceeds the amount of time specified by LOGLIMIT_TIME. At that point, a new log file is created with the most recent message to start. The old log file is left intact, though an earlier log file may be removed, depending on the LOGLIMIT_MAXFILES setting. The possible values for LOGLIMIT_TIME parameter are the following (where *n* is a positive integer): *n* MINUTE (or *n* MINUTES) *n* HOUR (or *n* HOURS) *n* DAY (or *n* DAYS) *n* MONTH (or *n* MONTHS) If the unit of time is not specified, it is assumed to be DAYS. The LOGLIMIT_TIME and LOGLIMIT_FILESIZE parameters are mutually exclusive; they both set limits on the standard log file, but one constrains by time, while the other constrains by file size. If both parameters are present in the ssa.ini file, the LOGLIMIT_FILESIZE parameter takes precedence. Default is DAYS.
LOGLIMIT_MAXFILES	Specifies the maximum number of standard log files. There are two options for limiting the standard log file: by size (LOGLIMIT_FILESIZE) and by time period (LOGLIMIT_TIME). Once the configured limit for the log file is reached, a new log file is created to store the latest messages. The old log file is not deleted, however. After the first log file has reached its limit and "rolled over", there will always be at least two log files--the current log file and the previous one--although, depending on the value for LOGLIMIT_MAXFILES, earlier log files may be deleted by the system. The LOGLIMIT_MAXFILES parameter determines the total number of log files, past and present, that will be retained. If 0 is specified for LOGLIMIT_MAXFILES, the maximum number of log files is set to unlimited. If **1** is specified, the default (**2**) is used instead. Default is 2.
STATLOGLIMIT_FILESIZE	Sets the maximum size of each query statistics log file in bytes. When the size limit for the current log file is reached, it is renamed, and the query statistics are written to a new log file. Default is 1048576 (1 MB).
TRACELOG	Enables (**ON** or **ENABLE**) or disables (**OFF** or **DISABLE**) the trace log file, which contains a lot of internal diagnostic information returned by different parts of Data Vault Service. This information might not have much practical value to end users, but can be analyzed by Informatica Global Customer Support to help diagnose problems or improve the system. By default, trace logging is enabled. Note that when LOGVERBOSE=**2** or **TRACE**, the information written to the standard log is the same as what is written to the trace file. Default is ON or ENABLE.

Column	Description
TRACELOGLIMIT_FILESIZE	Sets the maximum size (in bytes) for the trace log file. Once the log file grows to this size, one of two possibilities will occur: either new log information will "push out" the oldest log entries, keeping the log file below or at the specified maximum size; or, if TRACELOGLIMIT_MAXFILES is set to a value greater than 1, a new log file will be created to store the latest log information, leaving the previous log file intact. Setting TRACELOGLIMIT_FILESIZE to **0** is equivalent to omitting this parameter from the ssa.ini file. The TRACELOGLIMIT_TIME parameter will be used instead, unless that parameter is also missing, in which case the default value for TRACELOGLIMIT_FILESIZE will be used to limit the trace log file. The TRACELOGLIMIT_FILESIZE parameter takes precedence over TRACELOGLIMIT_TIME, so if both are set in the ssa.ini file, the tracelog file will be limited by size rather than time period. Default is 1048576 (1 MB).
TRACELOGLIMIT_TIME	Sets a time limit on the trace log file. When this option is enabled, entries are added to the current trace log until the timestamp interval between the first message and the latest one exceeds the amount of time specified by TRACELOGLIMIT_TIME. At that point, a new trace log file is created with the most recent message to start. The old trace log file is left intact, though an earlier log file may be removed, depending on the TRACELOGLIMIT_MAXFILES setting. The possible values for TRACELOGLIMIT_TIME parameter are the following (where *n* is a positive integer): *n* MINUTE (or *n* MINUTES) *n* HOUR (or *n* HOURS) *n* DAY (or *n* DAYS) *n* MONTH (or *n* MONTHS) If the unit of time is not specified, it is assumed to be DAY(S). The TRACELOGLIMIT_TIME and TRACELOGLIMIT_FILESIZE parameters are mutually exclusive; they both set limits on the trace log file, but one constrains by time, while the other constrains by file size. If both parameters are present in the ssa.ini file, the TRACELOGLIMIT_FILESIZE parameter takes precedence. Default is DAY(S).
TRACELOGLIMIT_MAXFILES	Specifies the maximum number of trace log files. There are two options for limiting the trace log: by size (TRACELOGLIMIT_FILESIZE) and by time period (TRACELOGLIMIT_TIME). Once the configured limit for the trace log file is reached, a new log file is created to store the latest messages. The old trace log is not deleted, however. There are always at least two trace log files--the current trace log file and the previous one--but depending on the value for TRACELOGLIMIT_MAXFILES, earlier log files may be deleted by the system when the current trace log reaches its limit and "rolls over". The TRACELOGLIMIT_MAXFILES parameter determines the total number of trace log files, past and present, that will be retained. If **0** is specified for TRACELOGLIMIT_MAXFILES, the maximum number of trace log files is set to unlimited. If **1** is specified, the default (**2**) is used instead. Default is 2.

The following table describes the optional parameters that enforce Data Vault user password policies:

Column	Description
COMPLEXPASSWORD	Enables the Data Vault password complexity policy. When the parameter is enabled, the password used to access Data Vault must contain at least one upper-case letter and at least one of the following special characters: @ # _ = , . ? ~ { } The following values are valid for the COMPLEXPASSWORD parameter: - ON - TRUE - ENABLE - 1 If you set the parameter value anything other than ON, TRUE, ENABLE, or 1, you receive an error during server startup. After you configure the property, restart the Data Vault server to enable the policy. When COMPLEXPASSWORD is enabled, Data Vault returns an error message if a user tries to update their Data Vault password to one that does not contain an upper-case letter and a special character. If you set COMPLEXPASSWORD=0, or do not give the parameter a value, then the COMPLEXPASSWORD policy is disabled and users can use passwords that do not contain any complexity control. To change the password for the DBA user with the COMPLEXPASSWORD policy, see the chapter "User Account Privileges."
MAXNUMPASSWORDS	Enables the policy that prevents the re-use of a specified number of Data Vault passwords that have been used before. When enabled, the parameter prevents the user from setting a Data Vault password that they have used before, up to a maximum of 12 previous passwords. For example, when you set MAXNUMPASSWORDS=10, the user cannot re-use their previous ten passwords when they update their password. Valid inputs are 0-12. If you give the parameter a value out of the 0-12 range, the user receives an error message when they update their password. After you configure the property, restart the Data Vault server to enable the policy. If you set MAXNUMPASSWORDS=0, or do not give the parameter a value, then the MAXNUMPASSWORDS policy is disabled and users can reuse any older password.

Column	Description
MINPASSWORDLENGTH	Enables the minimum password length policy. Use the parameter to specify the minimum number of characters that a user's Data Vault password must contain. When a user tries to update their password to one that has less than the number of characters specified, the Data Vault returns an error. For example: MINPASSWORDLENGTH=8 After you configure the property, restart the Data Vault server to enable the policy. In this scenario, if a user creates or updates a password that has less than 8 characters, the command fails with an "invalid password" error. If you give the parameter a value out of the 0-128 range, the Data Vault also returns an error when a user tries to update their password. If you set MINPASSWORDLENGTH=0, or do not give the parameter a value, then the MINPASSWORDLENGTH policy is disabled and users can create or update passwords of any length, up to 128 characters.
PASSWORDEXPIRYINDAYS	Enables the policy that requires a user to change their Data Vault password after a certain number of days, for example, 90 or 180 days. When enabled, Data Vault returns a warning message upon every user login starting 10 days before the current password is set to expire. Once the password expires, a user can no longer login through any ODBC or JDBC clients and must change their password through the Data Vault SQL Tool. Alternatively, the DBA can change a password for any user. For example: PASSWORDEXPIRYINDAYS=10 After you configure the property, restart the Data Vault server to enable the policy. In this scenario, since the password will expire within 10 days, the user is warned upon login that their password will expire in the remaining amount of days, hours, and minutes. They are asked to update their password. Once the password expires, the user is prompted to change their password upon login. If you give the parameter a value out of the 0-2,147,483,647 range, the user receives an error message upon login. If you set PASSWORDEXPIRYINDAYS=0, or do not give the parameter a value, then the PASSWORDEXPIRYINDAYS policy is disabled and user passwords will never expire.

AGENT Section

The AGENT section of the ssa.ini file contains the startup parameters for the Data Vault Agent.

The following table describes the parameters in the AGENT section of the ssa.ini file:

Parameter	Description
EXEHOST	The name of the machine on which Data Vault Service load balancer is running. This is identical to the HOST parameter in the [SERVER] section. If [SERVER] HOST and [AGENT] EXEHOST parameters are both present in the ssa.ini file and have different values, the value in the [SERVER] section takes precedence.
EXEPORT	The port through which Data Vault Service load balancer and Data Vault Agent processes communicate. This is identical to the EXEPORT parameter in the [SERVER] section. If [SERVER] EXEPORT and [AGENT] EXEPORT are both present in the ssa.ini file and have different values, the value in the [SERVER] section takes precedence.

Parameter	Description
PRIORITY	An integer value (**0**-**10**) that specifies the default priority of the local Data Vault Agent processes (where the higher the value, the greater the priority). The load balancer will always use the highest priority, non-busy Agent when delegating tasks. For example, if Agent 1 on machine 1 has a priority level of **5**, Agent 2 on machine 2 has a priority level of **10**, and neither Agent is currently performing any tasks, the load balancer will assign the next client query to Agent 2 because it has the higher priority. Note that this priority value can be overridden by specifying the **-p** option when starting a Data Vault Agent process. Default is 10.
LOGDIR	The directory path to the log file for the Data Vault Agent. Default is**.** (current directory).
LOGAPPEND	This parameter specifies whether Data Vault Agent logs are appended to (the default) or overwritten each time an Agent process is started on the same port. Possible LOGAPPEND values: **1** or **Y** turns log appending on **0** or **N** turns log appending off Default is 1 or Y
AGENTLOG_FILESIZE	Represents the approximate maximum file size of a single agent log file, in MB. For example, if you set the parameter to 10, after the agent log file reaches 10MB in size, a new log file will be created and used by agent. Recommended value range: 1 to 2000 Default is 100 MB.

META Section

The META section of the ssa.ini file contains the startup parameters for the Data Vault Service load balancer.

The following table describes the parameters in the META section of the ssa.ini file:

Parameter	Description
CONNECTION	The connection name that was used to start the Data Vault repository database. There must be a corresponding [CONNECTION *connection-name*] section in the **nucleus.ini** file.
DATABASE	The name of the Data Vault repository database.
UID	The user name that will be used to connect to the Data Vault repository database. It must be a user with at least INSERT, UPDATE, and DELETE privileges on the tables in the SDS_REPOSITORY schema.
PWD	The password for the user specified by UID. If the user does not have a password, leave this parameter empty.

The following table describes the optional parameters of the META section:

Parameter	Description
AUTOSTARTUP	This parameter is used to enable the automatic startup of a Data Vault repository database when the load balancer is started. The same Data Vault repository database shuts down when the load balancer is shut down. Possible AUTOSTARTUP values: **1** enables autostartup **0** disables autostartup When AUTOSTARTUP=1, the STARTUPHOST and STARTUPCOMMAND parameters should be set as well. Default is 0.
STARTUPHOST	The name of the machine where the Data Vault repository database will be started when AUTOSTARTUP=1. If autostartup is turned on and this parameter is missing or empty, the local machine is assumed to be the Data Vault repository database host machine. If a remote host is specified, a Data Vault Agent process with an AID set to **0** must be started on that machine in order for the autostartup option to work. This Agent will be dedicated to starting/stopping the Data Vault repository database, and will not be able to process client requests. Thus, there must be at least one other Data Vault Agent process running on the same machine, or elsewhere on the network, to handle queries. Default is localhost.

STARTUPCOMMAND	The command line that will be used to start the Data Vault repository database, when AUTOSTARTUP=1. Use the following syntax:
	For Microsoft Windows:
	`STARTUPCOMMAND=fb_inet_server -a`
	For UNIX:
	`STARTUPCOMMAND=fb_inet_server -m`

LOGINTIMEOUT	Sets the amount of time (in seconds) before a connection attempt to the Data Vault repository database is cancelled. Specifying **0** means the timeout period defined for the Nucleus ODBC Driver will be used (by default, **120** seconds). Note that a negative value is not permitted for this parameter. Default is 0.
CASTING	Enables (1) or disables (0) data type conversion. When data type conversion is enabled, certain data types can differ between columns in a registered compressed data file and ones in the archived table to which it belongs. The server will attempt to convert the compressed data file column to make it compatible with the table column. When data type conversion is disabled, the data type of a column in a compressed data file must be exactly the same as that of the corresponding column in the table, although length, precision and/or scale may differ in some cases. Default is 0

Parameter	Description
TRUNCATION	Enables (**1**) or disables (**0**) data truncation. (This setting only works when data type conversion is turned on: CASTING=1.) When data truncation is enabled, more data types can be converted between a compressed data file and an archived table, but the trade-off is that some Data Vault data might be truncated by the conversion. (Note that the "truncation" here is only virtual; the data in compressed data files is read-only and cannot be changed.) When data truncation is disabled, the types of conversions that can result in truncated Data Vault data are disallowed. Default is 0.
ADMIN_TASKTIMEOUT	Sets the timeout value for administration tasks. Enter the same value as the Task_Timeout administration tool command. Value is in milliseconds. Default value is 60000 (60 seconds).

ADMIN Section

The ADMIN section of the ssa.ini file contains the parameters for the Data Vault Administration Tool.

The following table describes the parameters in the ADMIN section of the ssa.ini file:

Parameter	Description
EXEHOST	The name of the machine on which the Data Vault Service load balancer is running. This is identical to the HOST parameter in the [SERVER] section. If [SERVER] HOST and [ADMIN] EXEHOST parameters are both present in the ssa.ini file and have different values, the value in the [SERVER] section takes precedence.
EXEPORT	The port through which the Data Vault Service load balancer and Data Vault Administratition Tool processes communicate. This is identical to the EXEPORT parameter in the [SERVER] section. If [SERVER] EXEPORT and [ADMIN] EXEPORT are both present in the ssa.ini file and have different values, the value in the [SERVER] section takes precedence.

STARTER Section

The STARTER section of the ssa.ini file contains the parameters for the automation of the Data Vault Service startup and shutdown.

The following table describes the parameters in the STARTER section of the ssa.ini file:

Parameter	Description
SERVER_CMD	The command to start the Data Vault Service. Default is ssaserver.
AGENT_CMD	The command to start the Data Vault Agent. Default is ssaagent. Do not include the -u parameter to specify a numeric ID for the agent process. The startup service assigns an agent ID.
AGENT_COUNT	The number of Data Vault Agents processes to start. Default is 2.
LOGDIR	The directory where the log files will be written. On UNIX, the daemon log is named ssa_starter. On Windows the service activities are logged by the operating system, which can be viewed through the Event Viewer.

Parameter	Description
AGENT_CONTROL	Enables (1) or disables (0) automatic restart if the Data Vault Agent fails. Default is 1.
SERVER_CONTROL	Enables (1) or disables (0) automatic restart if the Data Vault Service fails. Default is 1.
USER	UNIX only. The user ID under which all processes will be started by the ssa_starter daemon. It is usually important to set this parameter, since the default is the root user, which can cause problems. For example, if a process running as root creates a file, only another root process can access that file. This parameter can also be set using the SSA_STARTER_USER environment variable. If both the ssa.ini file and the environment variable define the user ID, the environment variable takes precedence.
VERBOSE	The level of detail for the messages written to the log files. The following are the possible options: - 0 : Includes all error messages and the most important messages - 1 : Level 0 plus more detailed messages - 2 : Level 1 plus system trace information Default is 2.
EXE[0-9]	These 10 optional parameters (EXE0, EXE1, EXE2, ..., EXE9) refer to custom commands, scripts, or executables that will run in conjunction with Data Vault Service on startup. Note that the numbering of the parameters has no relationship to the order in which they are processed. These parameters are run asynchronously, so there is no guarantee about the execution order.

Common Configuration

There are some common parameters that can be set in multiple ssa.ini sections, allowing different components to use different parameter values. However, if the same parameter value is required for different components, there is an alternative to setting the same parameter multiple times: The COMMON section is where common parameters can be set once for all or some of the Data Vault Service components.

A common parameter is first sought in its proper section. If it is not found there, the COMMON section is checked. So in order for a common parameter to be used, the parameter should be set in the COMMON section and also omitted from the sections that will make use of this common parameter. A common parameter can be set in both the COMMON section and a specialized section, but the specialized section value will override the common one.

Note that the default value of a common parameter will be used only if there is no valid value in both the specialized section and the COMMON section.

The following are the parameters that can be set in the COMMON section, along with the other sections where they can appear:

- EXEHOST – [SERVER], [AGENT], [ADMIN]
- EXEPORT – [SERVER], [AGENT], [ADMIN]
- LOGDIR – [SERVER], [AGENT], [STARTER]
- SHAREDIR – [SERVER], [QUERY]
- TEMPDIR – [SERVER], [QUERY], [EXPORT], [MERGE]
- IDVCHARSET
- MAXNUMLEGALHOLD

Source Data Encoding

The **IDVCHARSET** parameter gives you the choice to represent source data as either extended ASCII or UTF-8 in Data Vault.

To designate ASCII or UTF-8 encoding for source data, add the following parameter to the `ssa.ini` file:

`IDVCHARSET=`
When you provide an "ASCII" value, the `IDVCHARSET` parameter enables ASCII mode and handles source data as sets of bytes in the range 0x00-0xFF. For example: `IDVCHARSET=ASCII`

If you provide an "UTF-8" value, Data Vault handles all object names and string data as UTF-8 compliant. For example: `IDVCHARSET=UTF-8`

Do not switch to ASCII mode on a Data Vault installation with archived or retired data. This results in a combination of ASCII and UTF-8 data, which may cause some Data Vault functions to fail.

Maximum Number of Legal Holds

Add the MAXNUMLEGALHOLD parameter to the COMMON section of the ssa.ini file to set the maximum number of legal holds allowed per table. By default, the maximum number of legal holds allowed per table is 256, but you can configure the parameter up to 16384.

Amazon S3 Configuration

To configure Amazon S3 cloud storage as Data Vault archive storage, add the required S3 parameters to the end of the `ssa.ini` file.

[AWS_CONNECTION]

Name of the archive folder for the AWS connection. For example, if the archive folder name is "test": `[AWS_CONNECTION test]`

AWS_KEY=

Use the following format: *Amazon Web Services Key*:*Amazon Web Services Secret Key*

For example: *AWS_KEY=AKIAI62XJDYYYXXXXXX*:*MFx94W+ZlGTdEXom+21BKBh4Y41y11x1x1x1x1*

The length of the key must be greater than 64 characters.

ARCHIVE_NAME=

Name of the AWS storage bucket. For example: `ARCHIVE_NAME=awsBucket`

URL=

URL where the storage bucket is created, including the region. For example: use the URL `https://s3.amazonaws.com` for the US East (North Virginia) region. To access different regions, the region name must be part of the URL. For example: `http://s3.eu-central-1.amazonaws.com/`

For a full list of regions, see Amazon Web Services documentation.

Testing the Connection

Use the `ssadrv` command to validate the connection details you provided in the `ssa.ini` file.

On the machine that hosts the Data Vault service, run the following command:

```
ssadrv -a aws://<connection name>/<URL>/<Amazon aws bucket>/
```

For example: `ssadrv -a aws://test/https://s3.amazonaws.com/awsBucket/`

AWS S3 Configuration for Keyless Access

You can configure AWS S3 cloud storage for Data Vault archive storage. If Data Archive and Data Vault are installed on a Windows 64-bit or Red Hat Enterprise Linux 7 environment, you can set up keyless access to the AWS S3 archive store.

To set up keyless access to the AWS S3 archive store, enter the following properties to the end of the `ssa.ini` file:

[AWS_SDK_CONNECTION]

Name of the archive folder. For example, if the name of the archive folder is `test`, then enter: `[AWS_SDK_CONNECTION test]`

PROFILE_NAME

The profile name of the S3 connection. Specify a custom name or set to `default`. For example: `PROFILE_NAME=default`

To connect a Data Archive instance that runs on Amazon EC2 to Amazon S3, add an Identity Access Management role. Set the profile name to `default`.

When Data Archive uses a service other than Amazon EC2, connect to Amazon S3 in one of the following ways:

- Set the AWS_ACCESS_KEY_ID and AWS_SECRET_ACCESS_KEY properties as environment variables. Set the profile name to `default`.
- Create a credentials file using AWS CLI. This file contains the user defined profile and the values for the AWS_ACCESS_KEY_ID and AWS_SECRET_ACCESS_KEY properties.

Testing the Connection

Use the `ssadrv` command to validate the connection details you provided in the `ssa.ini` file.

On the machine that hosts the Data Vault service, run the following command: `ssadrv -a aws_sdk://<connection name>/<Amazon bucket url>`

For example:

```
ssadrv -a aws_sdk://test/https://s3.amazonaws.com/awsBucket/
```

You can use AWS virtual-hosted references. For example:

```
aws_sdk://<ConnectionName>/https://<bucketname>.s3.amazonaws.com:443/
```

nucleus.ini

The nucleus.ini file can contain the following sections:

CONNECTION

A CONNECTION section (server-side and client-side) indicates the port number and host name used to connect to a database instance. A server-side CONNECTION section can also contain various other system parameters relating to operations for the connection.

CLIENT Section

Contains parameters that set options for client applications (ssadl, ssasql, or ODBC-enabled front-ends) connecting to database instances.

CONNECTION Section

The CONNECTION section contains the port number and host name used to connect to a Data Vault repository. It can also contain various other system parameters relating to operations for the connection (note that these are usually left as the Nucleus default values, and so generally do not appear in the nucleus.ini file).

The server-side nucleus.ini file should contain a CONNECTION section for each database instance that will be running concurrently on the same host machine (since each instance requires its own port number). Any number of non-concurrent database instances may use the same connection.

CONNECTION sections in client-side nucleus.ini files specify port numbers and host names for connection to database instances running on remote engines.

The format for the section header is [CONNECTION *connection-name*], where *connection-name* can be any string of 31 characters or less. The connection name will be specified when starting or connecting to a database instance on the specified host using the specified port, in the following manner:

```
ssasql connection-name instance-name user-name[/user-password ]
ssadl connection-name instance-name user-name[/user-password ] load-spec-file
```

CONNECTION Section Parameters

The following table describes the CONNECTION section parameters:

Parameter	Description
Host	The host name or IP address of the computer on which the database instance is running. Default is current host name.
Port	The operating system port to be used for connection to the database instance. Each concurrently-running database instance must have its own port number. The port number must be between **1024** and **65,535**. Default is 5001.

CLIENT Section

The CLIENT section contains parameters that set options for client applications connecting to database instances. This section appears only in client-side nucleus.ini files.

CLIENT Section Parameters

The following table describes the CLIENT section parameters:

Parameter	Description
DatePic	A picture string indicating how a value stored using the DATE datatype will appear in output from the client application, as well as the format in which date values will be presented to the Data Vault Loader. Additionally, date constants may be referenced in SQL statements according to the user-defined date picture, as well as using the standard date formats. default. *yyyy-mm-dd* (for July 1, 2003, Data Vault Service will display 2003-07-01). Entries can be various combinations of the letters **y** for year, **m** for Month, **d** for Day, and **w** for Weekday (lowercase only), along with any other characters used as separators. Some common separators are colons (**:**), hyphens (**-**), and slashes (**/**). The Day, Month, Year and Weekday components can be ordered in any way, for example: **dd/mm/yy**, **mm/dd/yy**, **yy/mm/dd**. If the date picture specifies only the last two digits of the year (*yy*), the Data Vault Service must assume a default century for date constants that use this format. This determination is based on the last two digits of the year: - Last two digits. 00 - 50; Base year. 2000 (21st century) - Last two digits. 51 - 99; Base year. 1900 (20th century) The full year is produced by adding the last two year digits to the base year. For example, if *yy* = **03**, the base year is **2000**, and the full year is 03 + 2000 = **2003**. If the user-defined date picture is invalid, date values will be displayed using the default format.
TimePic	A picture string indicating how a value stored using the TIME datatype will appear in output from the client application, as well as the format in which time values will be presented to the Data Vault Loader. default. *hh:mm:ss* (for ten minutes past 2 pm, Data Vault Service will display 14:10:00) Entries can be various combinations of the letters **h** for Hours, **m** for Minutes, and **s** for Seconds (lowercase only); any other characters can be used as separators. Commonly used separators are colon (:), hyphen (-), and period (.). The Hour, Minute, and Second components can be ordered in any way, for example: **hh:mm:ss**, **ss:mm:hh**, **hh:ss:mm**. For a 24-hour clock, place the number 24 at the end of the picture. For a 12-hour clock, place the number 12 at the end of the picture. If you do not specify the clock type, a 24-hour clock is assumed. For 12-hour clocks, you can include the characters “am” in your picture in the location you want “am” or “PM” to appear. Use the case you want used in the display. According to the 24-hour clock format, noon is represented as 12:00:00 and midnight is 00:00:00. If the user-defined time picture is invalid, time values will be displayed using the default format.
TempDrive	Sets the drives where the temporary loader files will be written. Multiple drives can be listed, separated by semicolons (**;**). The number of temporary files created by the loader is related to the **ssadl -j** flag value: if the **-j** value is *n*, then there will be at least *n* temporary files. Default. . (current directory) For best results, ensure that the temporary drive is **not** located on the same disk as the database or, especially, the flat file. In addition, make sure that the amount of free disk space in the temporary drive is at least three (**3**) times the size of the flat file.

Parameter	Description
TempCache	Allocates buffer space in memory (in megabytes) to hold Virtual File System (VFS) data during Parallel Loader operations. By default, VFS data is not cached. Enabling this parameter (by setting it to a positive value) can speed up load operations and compressed data file creation. - Default. 0 - Minimum. 0 - Maximum. none
TempPage	Sets the Virtual File System (VFS) page size (in kilobytes). Data is written to the VFS in clusters of this size. If multiple locations for the VFS are specified by the TempDrive parameter, one page of data at a time will be written to each location in "round robin" fashion. - Default. 32 - Minimum. 1 - Maximum. none

Date Picture Components

The following tables define the components of the Date Picture parameter.

Day

The following table describes the Day component of the Date Picture parameter:

Column	Description
d	Integer between 1 and 31, corresponding to day of month; leading zeros are not displayed.
dd	Integer between 01 and 31, corresponding to day of month; leading zeros are displayed.
ddd	Integer between 1 and 365 (366 in a leap year), corresponding to number of days since first of the year; leading zeros are not displayed.

Month

The following table describes the Month component of the Date Picture parameter:

Column	Description
m	Integer between 1 and 12, corresponding to a month; leading zeros are not displayed
mm	Integer between 01 and 12, corresponding to a month; leading zeros are displayed
mmm	A three-character abbreviation for the month corresponding to the stored month number (for example: JAN, FEB, MAR, and so on)
mmmm	Month name corresponding to the stored month number (for example: January, February, March, and so on)

Year

The following table describes the Year component of the Date Picture parameter:

Column	Description
yy	Last two digits of stored year
yyyy	Entirety of stored year

Weekday

The following table describes the Weekday component of the Date Picture parameter:

Column	Description
ww	An integer corresponding to the day of week (Monday=1, Tuesday=2, Wednesday=3, and so on)
www	Three-character abbreviation for the day of the week (for example, MON, TUE,WED, and so on)
wwww	The day of the week on which a date falls (for example, Monday, Tuesday, Wednesday, and so on)

The following examples show some date pictures using the date August 31, 2002:

Date Picture	Displayed Date
m/d/yy	8/31/02
mm/dd/yy	08/31/02
mm/dd/yyyy	08/31/2002
dd-mm-yy	31-08-02
yyyy-mm-dd	2002-08-31
mmm-dd-yyyy	AUG-31-2002
mmddyyyy	08312002
mmmm,yyyy	August,2002

Time Picture Components

The following tables define the components of the Time Picture parameter.

Hour

The following table describes the Hour component of the Time Picture parameter:

Column	Description
h	An integer between 0 and 23 (inclusive), corresponding to the hour of the day; does not display leading zeros
hh	An integer between 00 and 23 (inclusive), corresponding to the hour of the day; displays leading zeros

Minute

The following table describes the Minute component of the Time Picture parameter:

Column	Description
m	An integer between 0 and 59 (inclusive), corresponding to a number of minutes; does not display leading zeros
mm	An integer between 00 and 59 (inclusive), corresponding to a number of minutes; displays leading zeros

Second

The following table describes the Second component of the Time Picture parameter:

Column	Description
s	An integer between 0 and 59 (inclusive), corresponding to a number of seconds; does not display leading zeros
ss	An integer between 00 and 59 (inclusive), corresponding to a number of seconds; displays leading zeros

The following examples show time format masks and the resulting output, using the time 15:08:25.

Time Picture for 12-hour Clock	Displayed Date
hh:mm:ss am 12	03:08:25 pm
hh.mm 12	03.08
AM h-m-s 12	PM 3-8-25

Time Picture for 24-hour Clock	Displayed Date
hh:mm:ss 24	15:08:25
hh.mm 24	15.08
h-m-s 24	15-8-25

Environment Variables

The Data Vault uses environment variables to store information about the Data Vault environment.

You can use the following environment variables:

- SSA_INI_DIR. Stores the directory path to the ssa.ini file.
- SSA_STARTER_USER. UNIX only. Stores the user ID that runs all processes started by the ssa_starter daemon.
- NUCLEUS. Stores parameters found in the nucleus.ini.

SSA_INI_DIR Environment Variable

The SSA_INI_DIR environment variable contains the path to the ssa.ini configuration file. If the SSA_INI_DIR variable is not defined, the Data Vault Service load balancer and agent look for the configuration file in the current directory. An error message is returned if the ssa.ini file is not found.

SSA_STARTER_USER Environment Variable

On UNIX, the SSA_STARTER_USER environment variable sets the user ID under which all processes will be started by the ssa_starter daemon. It is usually important to set this value, since the default is the root user, which can cause problems. For example, if a process running as root creates a file, only another root process can access that file.

The user ID can also be set through the USER parameter in the STARTER section of the ssa.ini file. If both the ssa.ini file and the environment variable define the user ID, the environment variable takes precedence.

NUCLEUS Environment Variable

Any of the operating parameters that can be set in the nucleus.ini file can also be set for the current operating environment by including them in the NUCLEUS variable. Settings made in the NUCLEUS variable override those made in the nucleus.ini file. To set the NUCLEUS variable, refer to the set of instructions for the appropriate operating system/shell.

UNIX

Use the following syntax at the operating system command prompt (the quotation marks must be typed as shown):

- Bourne and Korn Shells

```
NUCLEUS='PARAMETER-NAME:value [PARAMETER-NAME:value...]'
export NUCLEUS
```

- C Shell

```
setenv NUCLEUS 'PARAMETER-NAME:value [PARAMETER-NAME:value...]'
```

Any number of parameters can be set in this way; each parameter-name/value pair must be separated by a space. If more than one parameter is set in the NUCLEUS variable, they must be enclosed by single or double quotation marks. If a hyphen (**-**) is specified for *value*, the Nucleus default value is used.

Windows

Perform the following actions to set or edit the NUCLEUS environment variable in Windows 2000. Normally, the NUCLEUS variable with the SUPPORT parameter is automatically set during the initial Data Vault installation. Note that quotation marks must not surround the parameter values in Windows 2000.

1. Open the Windows Control Panel (*Start Menu* **>***Settings* **>***Control Panel*).
2. On the Control Panel, double-click System. The System Properties dialog box opens.
3. On the Advanced pane, click ENVIRONMENT VARIABLES (Alt+E). The Environment Variables dialog box appears.
4. On the Environment Variables dialog box, click NEW (Alt+N) under the system variables section to create the NUCLEUS variable, or select NUCLEUS and click EDIT (Alt+I) if the variable has already been defined. The New System Variable or Edit System Variable dialog box opens.
5. On the New/Edit System Variable dialog box, enter the required information:

   Variable Name (Alt+N): NUCLEUS

   Variable Value (Alt+V): *PARAMETER-NAME:value* [*PARAMETER-NAME:value*...]

   Any number of parameters can be set in this way; each parameter-name/value pair must be separated by a space. If a dash (**-**) is specified for value, the Nucleus default value is used. Note that the SUPPORT parameter value must be a *short* path name.
6. Click **OK** to make the settings and close the New/Edit System Variable dialog box.
7. Click **OK** to close the Environment Variables dialog box.
8. Click **OK** to close the System Properties dialog box.

The following text gives an example for UNIX.

```
NUCLEUS='SUPPORT:/usr/sand PRIMARYDB:db1'
export NUCLEUS
```

Any database instance started subsequently will run with db1 as the underlying primary ("real") database.

Client Connections

Clients must connect to the Data Vault to query archived tables registered with the Data Vault repository. Clients can connect to Data Vault using an ODBC-enabled front-end tool. To make a connection possible, a new [CONNECTION *connection-name*] section must be added to the client-side `nucleus.ini` file (where *connection-name* can be any desired name for the connection). The Host and Port values in this section must match those used by the Data Vault Service load balancer, as specified in the `ssa.ini` file. The *connection-name* is subsequently used by the client to connect to the load balancer.

The following table gives an example of the client connections:

ssa.ini (excerpt)	**client-side nucleus.ini (excerpt)**

`[SERVER]`
`Host=ALPHA1`
`Port=9975`
`...`
	`[CONNECTION dvs]`
	`Host=ALPHA1`
	`Port=9975`
	`...`

In the above example, *dvs* is the connection name. To connect to a Data Vault repository database called *adm*, the user authorization JOE uses the following command line invocation:

```
ssasql dvs adm JOE
```

ODBC client connections also require a new Data Source Name (DSN) that references the information described above.

CHAPTER 4

Data Vault SSL Setup

This chapter includes the following topics:

- Data Vault SSL Setup Overview, 46
- Security Certificates, 47
- Generate Certificate Authority and Certificate Authority-Signed Certificates, 47
- Generate Self-Signed Certificates, 51
- SSL Configuration in Data Vault, 52
- Alternate Names, 55
- OpenSSL Configuration File for UNIX OS, 56
- OpenSSL Configuration File for Microsoft Windows OS, 61
- SSL Setup in Data Archive, 65
- Limitations, 66
- Troubleshooting, 66

Data Vault SSL Setup Overview

Data Vault uses TLS1.2 technology to secure communication between the Data Vault service and clients such as ODBC, JDBC, SSASQL, Data Archive, and more. To implement SSL, you can use either Certificate Authority (CA) certificates or self-signed certificates.

Secure Sockets Layer, or SSL, is a security technology that establishes an encrypted link between a server and client. The encrypted link ensures that data passed between the web server and browser remains private and secure.

Data Vault uses a successor of SSL called Transport Layer Security, or TLS, which is also commonly called SSL. When secured by TLS, connections between a client, for example a web browser, and a server, for example wikipedia.org, have one or more of the following properties:

- A secure, private connection. TLS protocol uses symmetric cryptography to encrypt the transmitted data. The keys for the symmetric encryption are generated uniquely for each connection and are based on a shared secret negotiated at the start of the session. The server and client negotiate the details of which encryption algorithm, or cypher, and cryptographic keys to use before any data is transmitted. The negotiation of a shared secret is secure, because the negotiated secret is unavailable to eavesdroppers and cannot be obtained, even by an attacker who places themselves in the middle of the connection. Attackers cannot modify the communications during the negotiation without being detected.

- Identification of the communicating parties. The identity of the communicating parties can be authenticated using public-key cryptography. This authentication can be optional, but it is generally required for at least one of the parties, typically the server.
- Integrity during data transmission. Each message transmitted includes a message integrity check that uses a message authentication code to prevent undetected loss or alteration of the data during transmission.
- A cipher suite. A cipher suite is a named combination of authentication, encryption, message authentication code (MAC), and key exchange algorithms. It is used to negotiate the security settings for a network connection using the TLS/SSL network protocol.

In previous versions of Data Vault, you enabled SSL by adding "SSL=1" to the [SERVER] section of the ssa.ini file. This implementation only handled data encryption. There was no server certificate authentication and no handshake. The SSL=1 flag is obsolete in version 6.4.3 HotFix 1 and later. Even if you set the SSL=1 parameter in ssa.ini, your data will not be encrypted.

The current SSL implementation uses TLS1.2 and server certificate validation to make a secured connection. To use encryption in Data Vault, you must have the required server certificate and use one of the following implementations:

1. Using a certificate signed by a Certificate Authority:
   - CA root certificate at the client side and CA-signed server certificate at the server side.
2. Using a self-signed certificate:
   - Self-signed certificate at both client side and as the server certificate.

Security Certificates

To be able to create an SSL connection, a web server requires an SSL certificate. You can create a self-signed certificate or get a certificate from a certificate authority, for example Verisign.

Both certificates work in the same way. They enable you to create sites that are inaccessible to third parties. Data transferred through an SSL or HTTPS connection is encrypted to provide a high level of security.

A certificate from a CA implies that your website is secure, because it is certified by a trusted source. CA's like Verisign verify the ownership of the domain and check the trustworthiness of the business before issuing an SSL security certificate.

A self-signed certificate can be useful in testing environments or when the transmitted data is not particularly sensitive.

Generate Certificate Authority and Certificate Authority-Signed Certificates

A Certificate Authority (CA) is an entity that issues digital certificates.

A digital certificate certifies the ownership of a public key by the named subject of the certificate. This allows other parties to rely upon signatures or on assertions made about the private key that corresponds to the certified public key. A CA acts as a third party trusted both by the owner of the certificate and by the party that relies upon the certificate. The format of these certificates is specified by the X.509 standard.

You will need both a CA root certificate and a server certificate. Place the server certificate only at the server location. Place the CA certificate only at the client side. During a connection, the client sends a hello that contains a list of its ciphers to the server. The server then sends a list of supported ciphers and its server certificate as a response. The client authenticates the server certificate with the CA certificate and generates a symmetric certificate. This symmetric certificate is passed to the server. The server generates its own symmetric certificate. This symmetric certificate is used to create a symmetric cryptography bridge between the client and server, and start an encrypted session.

You can create your own CA certificate and use it to generate a server key and certificate.

Prerequisites

Before you generate CA or CA-signed certificates, complete the following prerequisites:

1. Download and install the OpenSSL toolkit. Add the directory to the PATH environment variable.
2. Create a directory where you want to generate the keys and certificates. Change directories to that location.
3. Create a configuration file called `openssl.cnf` and depending on the OS, configure it as shown in the topics "OpenSSL Configuration File for UNIX" or "OpenSSL Configuration File for Microsoft Windows." Set the variable OPENSSL_CONF with the following command:
   On UNIX:

   ```
   export OPENSSL_CONF=<full path of openssl.cnf_file>
   ```

   On Microsoft Windows:

   ```
   Set OPENSSL_CONF=< full path of openssl.cnf_file >
   ```
4. Create a directory structure as shown in the following example:
   INFA_sampleSigningCA1, INFA_sampleSigningCA1/ca.db.certs.

   Use the chmod command to give full permissions to these files.
5. Inside the INFA_sampleSigningCA1 directory, create the files below:
   `ca.db.serial`, `ca.db.index`, `ca.db.rand`

   Reference the "OpenSSL Configuration File for UNIX" or "OpenSSL Configuration File for Microsoft Windows" topics to configure the contents of these files.

   In the `ca.db.serial` file, input value 01, save, and close the file.

   Use the chmod command to give full permissions to these files.

Step 1. Create the Certificate Authority Root Key and Certificate

Create the keypair and use the root key to sign the CA certificate.

1. Create the keypair:

   ```
   openssl genrsa -des3 -out <your CA key name> <key length>
   ```

   For example: `openssl genrsa -des3 -out -root-ca.key 2048`

   ```
   Generating RSA private key, 2048 bit long modulus
       ..............++++++
       ..........++++++
       e is 65537 (0x10001)
       Enter pass phrase for root-ca.key:
       Verifying - Enter pass phrase for root-ca.key:
   ```

   You are asked for a password that will be the CA password. Then the system asks for the password again. The output of this command, the file `root-ca.key`, contains an RSA keypair that is encrytped using the password you supply. For someone to use this key to create new certificates, either host or client, they will need both this file and the password.

2. Use the CA root key to sign the CA certificate:

```
openssl req -new -x509 -days <days of validity> -key <your CA key name> -out <root
CA certificate name> -config <config file name>
```

For example:

```
openssl req -new -x509 -days 3650 -key root-ca.key -out root-ca.crt -config
openssl.cnf
```

```
Create a new, self-signed X.509 certificate valid for ten years, for the keypair in
the file root-ca.key, and place the output in the file root-ca.crt.
```

You are prompted to give identifying information for the certificate. Do not to use single quotes in the responses, due to a quirk in the Globus implementation. For example, don't use a common name like "Alice's CA". If you have customized the configuration file as suggested above, the defaults you specified there will make this step easier. The openssl req command recognizes that the request is for a self-signed certificate, and automatically applies suitable options, such as setting "CA:TRUE."

The default values as shown above in square brackets are from the configuration file. You can input any value or use the default. Provide a common name. Do not use an email address.

The following text is a sample output screen:

```
Enter pass phrase for root-ca.key:
    You are about to be asked to enter information that will be incorporated
    into your certificate request.
    What you are about to enter is what is called a Distinguished Name or a DN.
    There are quite a few fields but you can leave some blank
    For some fields there will be a default value,
    If you enter '.', the field will be left blank.
    -----
    Country Name (2 letter code) [US]:
    State or Province Name (full name) [SampleProvince]:
    Locality Name (eg, city) [Madison]:
    Organization Name (eg, company) [SampleOrg]:
    Second Organization Name (eg, company) [Computer Sciences Department]:
    Organizational Unit Name (eg, section) [INFA_sample Project]:
    Common Name (eg, YOUR name) []:<Any Name eg: MyRootCA>
    Email Address []:
```

3. Use the commands below to move the `root.ca.crt` and `root-ca.key` files:

On UNIX:

```
cp root-ca.crt INFA_sampleSigningCA1/signing-ca-1.crt
cp root-ca.key INFA_sampleSigningCA1/signing-ca-1.key
```

On Microsoft Windows:

```
copy root-ca.crt INFA_sampleSigningCA1\signing-ca-1.crt
copy  root-ca.key INFA_sampleSigningCA1\signing-ca-1.key
```

You must copy the files above because you have set the location and name of the CA certificate this way in the `openssl.conf` file.

Step 2. Generate a Host Certificate for the Server

Generate a host certificate for the server and provide information for the certificate request.

- Generate the host certificate:

```
openssl req -newkey rsa:2048 -keyout <server key>.key -nodes -config <conf file
name> -out <server cert>.req
```

For example:

```
openssl req -newkey rsa:2048 -keyout myserver.key -nodes -config openssl.cnf -out
myserver.req
```

```
Generating a 2048 bit RSA private key
    ..............++++++
```

```
    .++++++
    writing new private key to 'host_omega.key'
    -----
    You are about to be asked to enter information that will be incorporated
    into your certificate request.
    What you are about to enter is what is called a Distinguished Name or a DN.
    There are quite a few fields but you can leave some blank
    For some fields there will be a default value,
    If you enter '.', the field will be left blank.
    -----
    Country Name (2 letter code) [US]:
    State or Province Name (full name) [SampleProvince]:
    Locality Name (eg, city) [Madison]:
    Organization Name (eg, company) [SampleOrg]:
    Second Organization Name (eg, company) [Computer Sciences Department]:
    Organizational Unit Name (eg, section) [INFA_sample Project]:
    Common Name (eg, YOUR name) []: server.domain.com  ( PROVIDE SERVER HOST NAME
WITH FULLY QUALIFIED NAME)
    Email Address []:someone@something.com
```

The default values as shown above in square brackets are from the configuration file. You can input any value or use the default. For common name, provide the fully-qualified domain name of the server for which the certificate is being generated. For example, if the hostname is abc and the domain name is dom.com, the common name is abc.dom.com.

Step 3. Sign the Certificate

Use the openssl command to sign the certificate.

- Sign the certificate:

```
openssl ca -config <conf file name> -out <server cert>.crt -infiles <server cert>.req
```

For example:

```
openssl ca -config openssl.cnf -out myserver.crt -infiles myserver.req
```

```
Using configuration from openssl.cnf
    Enter pass phrase for ./INFA_sampleSigningCA1/signing-ca-1.key:
    DEBUG[load_index]: unique_subject = "yes"
    Check that the request matches the signature
    Signature ok
    Certificate Details:
            Serial Number: 2 (0x2)
            Validity
                Not Before: Apr 29 15:18:20 2008 GMT
                Not After : Apr 29 15:18:20 2009 GMT
            Subject:
                countryName               = US
                stateOrProvinceName       = SampleProvince
                localityName              = Madison
                organizationName          = SampleOrg
                organizationName          = Computer Sciences Department
                organizationalUnitName    = INFA_sample Project
                commonName                = <FULLY QUALIFIED DOMAIN NAME>
                emailAddress              = something@something.com
            X509v3 extensions:
                X509v3 Basic Constraints:
                    CA:FALSE
                Netscape Comment:
                    OpenSSL Generated Certificate
                X509v3 Subject Key Identifier:
                    56:B9:56:A2:B1:BB:7B:61:0E:21:71:A1:BC:3E:CD:E2:79:DD:F1:75
                X509v3 Authority Key Identifier:
                    keyid:95:AE:11:9A:6C:3A:07:F5:6C:4A:CB:A8:5A:77:15:C5:02:30:08:37
                    DirName:/C=US/ST=SampleProvince/L=Madison/O=SampleOrg/O=Computer
Sciences Department/OU=INFA_sample Project/CN=MyRootCA
                    serial:ED:11:AB:0C:05:2F:6B:84
```

```
    Certificate is to be certified until Apr 29 15:18:20 2009 GMT (365 days)
    Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
    Write out database with 1 new entries
    Data Base Updated
```

The OpenSSL toolkit generates the CA root and server certificates. Proceed to the topic "SSL Configuration in Data Vault."

Generate Self-Signed Certificates

When appropriate, you can use self-signed certificates for the SSL implementation.

You must pay for security certificates from a certificate authority. To reduce costs, you can use a self-signed certificate whenever possible. For example, web pages that do not require credit card information or sensitive data can be handled with a self-signed certificate. When developers are working on a secure website, they can test the site using a self-signed SSL security certificate.

To generate a self-signed certificate, first generate an RSA Private Key with the OpenSSL toolkit. Then create a certificate signing request and generate the certificate.

Prerequisites

Before you generate self-signed certificates, complete the following tasks:

1. Download and install the OpenSSL toolkit. Add the directory to the PATH environment variable.
2. Create a directory where you want to generate the keys and certificates. Change directories to that location.
3. Create a configuration file called `openssl.cnf` and depending on the OS, configure it as shown in the topics "OpenSSL Configuration File for UNIX" or "OpenSSL Configuration File for Microsoft Windows." Set the variable OPENSSL_CONF with the following command:
   On UNIX:
   ```
   export OPENSSL_CONF=<full path of openssl.cnf_file>
   ```
   On Microsoft Windows:
   ```
    Set OPENSSL_CONF=< full path of openssl.cnf_file >
   ```
4. Set the RANDFILE parameter with the following command:
   On UNIX:
   ```
   export RANDFILE=<accessible directory>/.rnd
   ```
   On Microsoft Windows:
   ```
   set RANDFILE = <accessible directory>/.rnd
   ```

Step 1. Generate a Private Key

Use the OpenSSL toolkit to generate an RSA Private Key.

The RSA Private Key is a 2048 bit RSA key. This will be used to generate signing requests.

Use the following command to generate the RSA Private Key:

```
openssl genrsa -out <name>.key 2048
```

Step 2. Generate a Certificate Signing Request

After you generate the private key, generate a certificate signing request (CSR). The CSR can be used in one of two ways. You can send the CSR to a certificate authority, such as Thawte or Verisign, who will verify the identity of the requestor and issue a signed certificate. You can also use the following command to self-sign:

```
openssl req -new -key <name>.key -out <name>.csr
```

```
Country Name (2 letter code) [US]:
    State or Province Name (full name) [SampleProvince]:
    Locality Name (eg, city) [Madison]:
    Organization Name (eg, company) [SampleOrg]:
    Second Organization Name (eg, company) [Computer Sciences Department]:
    Organizational Unit Name (eg, section) [INFA_sample Project]:
    Common Name (eg, YOUR name) []: <Fully Qualified Domain Name ie hostname with domain
name>
    Email Address []:
```

The default values as shown above in square brackets are from the configuration file. You can input any value or use the default. For common name, provide the fully-qualified domain name of the server for which the certificate is being generated. For example, if the hostname is abc and the domain name is dom.com, the common name is abc.dom.com.

Step 3. Generate a Self-Signed Certificate

Use the following command to generate the self-signed certificate:

```
openssl x509 -req -days 365 -in <name>.csr -signkey <name>.key -out <name>.crt
```

The command generates `server.crt` and `server.key`. Use these files in the Data Vault server and client locations as described later in the document.

Alternate command to create a self-signed certificate

Once you have fulfilled the prerequisites for a self-signed certificate, you can generate a self-sign certificate using a single command:

```
openssl req -x509 -nodes -newkey rsa:2048 -keyout <key.pem> -out <cert.pem> -days 365
```

Because the certificate is self-signed, a single certificate file functions as both the CA and server certificate.

After you generate the self-signed certificate, proceed to the topic "SSL Configuration in Data Vault." You need the certificate and the key that you generated to proceed.

SSL Configuration in Data Vault

Edit the ssa.ini and nucleus.ini files to configure SSL communication in Data Vault. You must also configure Data Vault to use SSL communication for JDBC and ODBC connections, and configure the LD_LIBRARY_PATH environment variable for Perl tools.

Configuring the Ssa.ini and Nucleus.ini Files

Edit the ssa.ini and nucleus.ini files to configure SSL communication in Data Vault.

1. For the Data Vault server component installation, add the following entry in both the [AGENT] and [ADMIN] sections:

```
CertFile = <full CA certificate file path>
```

The [ADMIN] section is not available by default. You must add this section manually.

For perl utilities, for example ssadbcopy, ssamigrate, ssacleanup, and ssaencrypt, it is important to add the certificate file path entry in the [AGENT] section. These utilities use the location internally.

2. In the [SERVER] section of ssa.ini, add the following entries:

```
CertFile = <full certificate file path>
PrivateKey = <full private key file path>
```

3. If you are not using alternate names as explained in the topic "Alternate Names," then you must add the fully-qualified domain name, for example "myhostname.mydomain.com," for the HOST variable in the [SERVER] section of the ssa.ini file. If you are using alternate names, for example "myhostname," then you can provide the same name for the HOST variable in the [SERVER] section of ssa.ini. Data Vault uses this entry internally when running Perl tools, so it is important for users who run utilities such as ssadbcopy, ssacleanup, ssamigrate, and others.

4. In the [CONNECTION] section of the nucleus.ini file, add the following parameters:

```
CertFile = <full path of the certificate file>
SSL = 1
```

For example:

```
[CONNECTIon fas]
Port=8507
Host=inkr58dsg519.informatica.com
CertFile=/data1/idauser/root.ca.crt
SSL=1
```

If you are not using alternate names as described in the "Alternate Names" topic, then you must add the fully-qualified domain name for the host variable in the [CONNECTION] section. Otherwise, if you are using subject alternate names, for example "myhostname," then you can provide the same name for HOST in this [CONNECTION] section.

Making JDBC Connections

In order to make JDBC connections, you must create a trust store. The trust store is used to store certificates from trusted certificate authorities.

1. Verify that the JDBC URL has SSL=1 appended to the URL: `jdbc:infafas://<server hostname with fully qualified domain name>:<Data Vault port number>?SSL=1`

   For example: `jdbc:infafas://invrlx62ilm29.informatica.com:7507?SSL=1`

2. Use the following OpenSSL command to create the export .der file:

```
openssl x509 -outform der -in <certificate>.pem -out <export file>.der
```

3. Use the following command to create a key store:

```
keytool -import -alias your-alias -keystore <keystore name>.jks -file <export
file>.der
```

4. For 3rd party JDBC tools, for example Squirrel, add the trust store in the batch file to start the tool, as shown in the following example:

   -Djavax.net.ssl.trustStore=E:/QA/SSL_Tests/TestCaInfa2/TestCaInfa/TestCA/demo10.jks -Djavax.net.ssl.keyStorePassword='password'

5. For a simple java program, add .JKS file as a connection property in the Java code.

6. Alternatively, you can pass the trust store details within the JDBC URL, using the following format:

   jdbc:infafas://<server hostname with fully qualified domain name>:<Data Vault server port number>?SSL=1&TrustStore=<jks file with absolute file name>&TrustStorePassword=<trustorepassword>&TrustStoreType=jks

   For example: jdbc:infafas://irs11ilm02:8564/adm?SSL=1&TrustStore=D:/filename.jks&TrustStorePassword=password&TrustStoreType=jks

   These parameters are optional. The .jks file path must contain a forward slash ("/) , regardless of the operating system.

Making ODBC Connections

Complete the following steps to make ODBC connections:

Before making an ODBC connection, ensure that the NUCLEUS variable is set in the SYSTEM variables as shown below:

```
NUCLEUS=SUPPORT:<Installation location which contains nucleus.ini file>
```

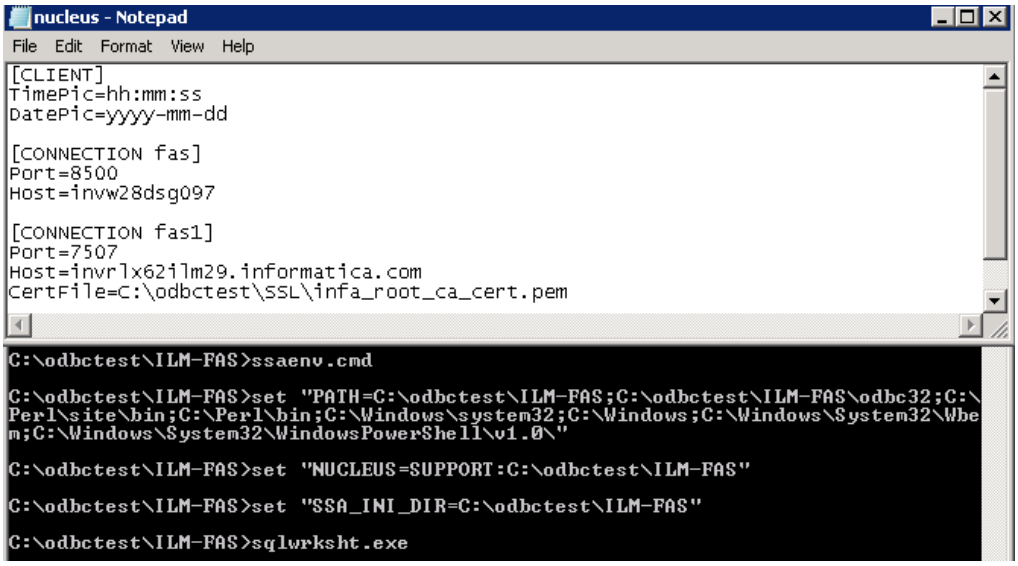
1. Create the DSN with the host name as a fully-qualified domain name. If the host name is part of the subject alternate names, you can use just the host name. For example:

   If the host name is "myhost" and the domain is "mydomain.com," if "myhost" is not added in the subject alternate names, input the host name as:
   myhost.mydomain.com

   Otherwise, if "myhost" is added in the subject alternate names, you can input only "myhost" in the host field when you create the DSN.

   Since you set the NUCLEUS variable as a pre-requisite, the nucleus.ini file in the location is updated with the new DSN name with the connection name as provided when you created the DSN.
2. Under the corresponding [CONNECTION] name, verify that the host name is exactly as given during DSN creation.
3. Add the following entries under [CONNECTION]:

   ```
   CertFile = <full path and name of the CA certificate file or self-signed certificate file>
   SSL=1
   ```



Using Perl Tools

Perl tools require odbc32 libraries. Before you use these tools, pre-pend the LD_LIBRARY_PATH with the odbc32 folder inside the Data Vault installation folder.

For example:

```
export LD_LIBRARY_PATH=/u01/Official/Jenkins_slave/workspace/SSL_TEST/ILM-FAS/
odbc32:$LD_LIBRARY_PATH
```

Unless you have defined the host name without the domain name as a subject alternate name, ensure that the HOST parameter in the SERVER section of ssa.ini is a fully qualified-domain name. The perl tools internally connect using the host name as provided in the SERVER section of ssa.ini.

For example:

HOST in the SERVER section of ssa.ini is "myhost" and the domain name is "mydomain.com."

If you have not used "myhost" as a subject alternate name while creating certificates, then ensure that the HOST parameter value in the SERVER section of ssa.ini is "myhost.mydomain.com." If you have used "myhost" as a subject alternate name, you can provide "myhost" as the HOST parameter value.

Alternate Names

You can provide the HOST parameter within nucleus.ini and ssa.ini as any of one the following: IP address, hostname, and hostname with fully-qualified domain names.

For these values, you must add the section "[alt_names]" inside the `openssl.cnf` file that you configured to generate the server certificates. The example `openssl.cnf` files provided in this chapter contain the required entries. You can edit this section of the configuration file to refer to an IP address or DNS. Data Vault does not support wild card characters in alternate names. For example, " *.example.com" is not a valid alternate name, while "sample.example.com" and "sample" are both valid.

The following text is an example of the [alt_names] section of the `openssl.cnf` file:

```
[alt_names]
DNS.1 = kb.example.com
DNS.2 = helpdesk.example.org
DNS.3 = systems.example.net
IP.1 = 192.168.1.1
IP.2 = 10.74.112.73
IP.3 = 2001:0db8:85a3:0000:0000:8a2e:0370:7334
```

OpenSSL Configuration File for UNIX OS

```
#
# OpenSSL Configuration file for INFA_sample Multi-level CA
#

# This definition stops the following lines choking if HOME isn't
# defined.
HOME            = .
RANDFILE        = $ENV::HOME/.rnd

####################################################################
[ ca ]
default_ca  = CA_default        # The default ca section

####################################################################
[ CA_default ]

dir     = ./INFA_sampleSigningCA1    # Where everything is kept
certs       = $dir/ca.db.certs  # Where the issued certs are kept
#crl_dir        = $dir/crl  # Where the issued crl are kept
database    = $dir/ca.db.index  # database index file.
new_certs_dir   = $dir/ca.db.certs      # default place for new certs.

certificate = $dir/signing-ca-1.crt # The CA certificate
serial      = $dir/ca.db.serial # The current serial number
#crl        = $dir/crl.pem      # The current CRL
private_key = $dir/signing-ca-1.key # The private key
RANDFILE    = $dir/.rand        # private random number file

x509_extensions = usr_cert      # The extentions to add to the cert
#x509_extensions = x509v3_extensions # The extentions to add to the cert
default_days = 730 # how long to certify for
default_crl_days= 30 # how long before next CRL
default_md = sha256 # which md to use.
preserve = no # keep passed DN ordering
# Comment out the following two lines for the "traditional"
# (and highly broken) format.
name_opt    = ca_default        # Subject Name options
cert_opt    = ca_default        # Certificate field options

# Extension copying option: use with caution.
# copy_extensions = copy

# Extensions to add to a CRL. Note: Netscape communicator chokes on V2 CRLs
# so this is commented out by default to leave a V1 CRL.
# crl_extensions    = crl_ext

default_days    = 365           # how long to certify for
default_crl_days= 30            # how long before next CRL
default_md = sha256           # which md to use.
preserve    = no            # keep passed DN ordering

# A few difference way of specifying how similar the request should look
# For type CA, the listed attributes must be the same, and the optional
# and supplied fields are just that :-)
```

```
policy      = policy_match

# A new one - quells a complaint from openssl ca
unique_subject = yes

####################################################################
[ CA_marginal ]

dir     = ./INFA_sampleSigningCA3    # Where everything is kept
certs       = $dir/ca.db.certs  # Where the issued certs are kept
#crl_dir        = $dir/crl  # Where the issued crl are kept
database    = $dir/ca.db.index  # database index file.
new_certs_dir   = $dir/ca.db.certs      # default place for new certs.

certificate = $dir/signing-ca-3.crt # The CA certificate
serial      = $dir/ca.db.serial # The current serial number
#crl        = $dir/crl.pem      # The current CRL
private_key = $dir/signing-ca-3.key # The private key
RANDFILE    = $dir/.rand        # private random number file

x509_extensions = usr_cert      # The extentions to add to the cert

# Comment out the following two lines for the "traditional"
# (and highly broken) format.
name_opt    = ca_default        # Subject Name options
cert_opt    = ca_default        # Certificate field options

# Extension copying option: use with caution.
# copy_extensions = copy

# Extensions to add to a CRL. Note: Netscape communicator chokes on V2 CRLs
# so this is commented out by default to leave a V1 CRL.
# crl_extensions    = crl_ext

default_days    = 365           # how long to certify for
default_crl_days= 30            # how long before next CRL
default_md = sha256           # which md to use.
preserve    = no            # keep passed DN ordering

# A few difference way of specifying how similar the request should look
# For type CA, the listed attributes must be the same, and the optional
# and supplied fields are just that :-)
policy      = policy_match

####################################################################
[ CA_root ]

dir             = ./INFA_sampleRootCA        # Where everything is kept
certs           = $dir/ca.db.certs      # Where the issued certs are kept
#crl_dir                = $dir/crl      # Where the issued crl are kept
database        = $dir/ca.db.index      # database index file.
new_certs_dir   = $dir/ca.db.certs         # default place for new certs.

certificate     = $dir/root-ca.crt  # The CA certificate
serial          = $dir/ca.db.serial     # The current serial number
#crl            = $dir/crl.pem          # The current CRL
private_key     = $dir/root-ca.key      # The private key
RANDFILE        = $dir/.rand            # private random number file

x509_extensions = v3_ca              # The extentions to add to the cert

# Comment out the following two lines for the "traditional"
# (and highly broken) format.
name_opt        = ca_default            # Subject Name options
cert_opt        = ca_default            # Certificate field options

# Extension copying option: use with caution.
# copy_extensions = copy

# Extensions to add to a CRL. Note: Netscape communicator chokes on V2 CRLs
# so this is commented out by default to leave a V1 CRL.
```

```
# crl_extensions        = crl_ext

default_days    = 1095                   # how long to certify for
default_crl_days= 30                    # how long before next CRL
default_md = sha256                   # which md to use.
preserve        = no                    # keep passed DN ordering

# A few difference way of specifying how similar the request should look
# For type CA, the listed attributes must be the same, and the optional
# and supplied fields are just that :-)
policy          = policy_match
unique_subject = yes

# For the CA policy
[ policy_match ]
countryName     = optional
stateOrProvinceName = optional
localityName            = optional
organizationName    = optional
organizationalUnitName  = optional
commonName      = supplied
emailAddress        = optional

# For the 'anything' policy
# At this point in time, you must list all acceptable 'object'
# types.
[ policy_anything ]
countryName     = optional
stateOrProvinceName = optional
localityName        = optional
organizationName    = optional
organizationalUnitName  = optional
commonName      = supplied
emailAddress        = optional

####################################################################
[ req ]
default_bits        = 2048
default_keyfile     = privkey.pem
distinguished_name  = req_distinguished_name
attributes      = req_attributes
x509_extensions = v3_ca # The extentions to add to the self signed cert

# Passwords for private keys if not present they will be prompted for
# input_password = secret
# output_password = secret

# This sets a mask for permitted string types. There are several options.
# default: PrintableString, T61String, BMPString.
# pkix   : PrintableString, BMPString.
# utf8only: only UTF8Strings.
# nombstr : PrintableString, T61String (no BMPStrings or UTF8Strings).
# MASK:XXXX a literal mask value.
# WARNING: current versions of Netscape crash on BMPStrings or UTF8Strings
# so use this option with caution!
string_mask = nombstr

# req_extensions = v3_req # The extensions to add to a certificate request

[ req_distinguished_name ]
countryName         = Country Name (2 letter code)
countryName_default     = US
countryName_min         = 2
countryName_max         = 2

stateOrProvinceName     = State or Province Name (full name)
stateOrProvinceName_default = SampleProvince

localityName            = Locality Name (eg, city)
localityName_default        = Madison
```

```
0.organizationName      = Organization Name (eg, company)
0.organizationName_default  = SampleOrg

1.organizationName      = Second Organization Name (eg, company)
1.organizationName_default  = Computer Sciences Department

organizationalUnitName      = Organizational Unit Name (eg, section)
organizationalUnitName_default  = INFA_sample Project

commonName          = Common Name (eg, YOUR name)
commonName_max          = 64

emailAddress            = Email Address
emailAddress_max        = 64

# SET-ex3           = SET extension number 3

[ req_attributes ]
# challengePassword     = A challenge password
# challengePassword_min     = 4
# challengePassword_max     = 20

# unstructuredName      = An optional company name

[ usr_cert ]

# These extensions are added when 'ca' signs a request.
basicConstraints = CA:FALSE
keyUsage = digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth
nsComment = "Issued by DSG Data Vault Product for Testing"
subjectAltName = @alt_names
# This goes against PKIX guidelines but some CAs do it and some software
# requires this to avoid interpreting an end user certificate as a CA.

basicConstraints=CA:FALSE

# Here are some examples of the usage of nsCertType. If it is omitted
# the certificate can be used for anything *except* object signing.

# This is OK for an SSL server.
# nsCertType            = server

# For an object signing certificate this would be used.
# nsCertType = objsign

# For normal client use this is typical
# nsCertType = client, email

# and for everything including object signing:
# nsCertType = client, email, objsign

# This is typical in keyUsage for a client certificate.
# keyUsage = nonRepudiation, digitalSignature, keyEncipherment

# This will be displayed in Netscape's comment listbox.
nsComment           = "OpenSSL Generated Certificate"

# PKIX recommendations harmless if included in all certificates.
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer:always

# This stuff is for subjectAltName and issuerAltname.
# Import the email address.
# subjectAltName=email:copy
# An alternative to produce certificates that aren't
# deprecated according to PKIX.
# subjectAltName=email:move

# Copy subject details
# issuerAltName=issuer:copy
```

```
#nsCaRevocationUrl      = http://www.domain.dom/ca-crl.pem
#nsBaseUrl
#nsRevocationUrl
#nsRenewalUrl
#nsCaPolicyUrl
#nsSslServerName


[alt_names]
DNS.1 = kb.example.com
DNS.2 = helpdesk.example.org
DNS.3 = systems.example.net
IP.1 = 192.168.1.1
IP.2 = 10.74.112.73
IP.3 = 2001:0db8:85a3:0000:0000:8a2e:0370:7334
[ v3_req ]

# Extensions to add to a certificate request

basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment

[ v3_ca ]


# Extensions for a typical CA


# PKIX recommendation.

subjectKeyIdentifier=hash

authorityKeyIdentifier=keyid:always,issuer:always

# This is what PKIX recommends but some broken software chokes on critical
# extensions.
#basicConstraints = critical,CA:true
# So we do this instead.
basicConstraints = CA:true

# Key usage: this is typical for a CA certificate. However since it will
# prevent it being used as an test self-signed certificate it is best
# left out by default.
# keyUsage = cRLSign, keyCertSign

# Some might want this also
# nsCertType = sslCA, emailCA

# Include email address in subject alt name: another PKIX recommendation
# subjectAltName=email:copy
# Copy issuer details
# issuerAltName=issuer:copy

# DER hex encoding of an extension: beware experts only!
# obj=DER:02:03
# Where 'obj' is a standard or added object
# You can even override a supported extension:
# basicConstraints= critical, DER:30:03:01:01:FF

[ crl_ext ]

# CRL extensions.
# Only issuerAltName and authorityKeyIdentifier make any sense in a CRL.

# issuerAltName=issuer:copy
authorityKeyIdentifier=keyid:always,issuer:always
```

OpenSSL Configuration File for Microsoft Windows OS

```
#
# OpenSSL Configuration file for INFA_sample Multi-level CA
#

# This definition stops the following lines choking if HOME isn't
# defined.
HOME            = .
RANDFILE        = $ENV::HOME/.rnd

####################################################################
[ ca ]
default_ca  = CA_default        # The default ca section

####################################################################
[ CA_default ]

dir     = INFA_sampleSigningCA1    # Where everything is kept
certs       = $dir\\ca.db.certs  # Where the issued certs are kept
#crl_dir        = $dir/crl  # Where the issued crl are kept
database    = $dir/ca.db.index  # database index file.
new_certs_dir   = $dir/ca.db.certs      # default place for new certs.

certificate = $dir/signing-ca-1.crt # The CA certificate
serial      = $dir/ca.db.serial # The current serial number
#crl        = $dir/crl.pem      # The current CRL
private_key = $dir/signing-ca-1.key # The private key
RANDFILE    = $dir/.rand        # private random number file

x509_extensions = usr_cert      # The extentions to add to the cert
#x509_extensions = x509v3_extensions # The extentions to add to the cert
default_days = 730 # how long to certify for
default_crl_days= 30 # how long before next CRL
default_md = sha256 # which md to use.
preserve = no # keep passed DN ordering
# Comment out the following two lines for the "traditional"
# (and highly broken) format.
name_opt    = ca_default        # Subject Name options
cert_opt    = ca_default        # Certificate field options

# Extension copying option: use with caution.
# copy_extensions = copy

# Extensions to add to a CRL. Note: Netscape communicator chokes on V2 CRLs
# so this is commented out by default to leave a V1 CRL.
# crl_extensions    = crl_ext

default_days    = 365           # how long to certify for
default_crl_days= 30            # how long before next CRL
default_md = sha256           # which md to use.
preserve    = no            # keep passed DN ordering

# A few difference way of specifying how similar the request should look
# For type CA, the listed attributes must be the same, and the optional
# and supplied fields are just that :-)
policy      = policy_match

# A new one - quells a complaint from openssl ca
unique_subject = yes

####################################################################
[ CA_marginal ]

dir     = INFA_sampleSigningCA3    # Where everything is kept
certs       = $dir/ca.db.certs  # Where the issued certs are kept
#crl_dir        = $dir/crl  # Where the issued crl are kept
```

```
database    = $dir/ca.db.index  # database index file.
new_certs_dir   = $dir/ca.db.certs      # default place for new certs.

certificate = $dir/signing-ca-3.crt # The CA certificate
serial      = $dir/ca.db.serial # The current serial number
#crl        = $dir/crl.pem      # The current CRL
private_key = $dir/signing-ca-3.key # The private key
RANDFILE    = $dir/.rand        # private random number file

x509_extensions = usr_cert      # The extentions to add to the cert

# Comment out the following two lines for the "traditional"
# (and highly broken) format.
name_opt    = ca_default        # Subject Name options
cert_opt    = ca_default        # Certificate field options

# Extension copying option: use with caution.
# copy_extensions = copy

# Extensions to add to a CRL. Note: Netscape communicator chokes on V2 CRLs
# so this is commented out by default to leave a V1 CRL.
# crl_extensions    = crl_ext

default_days    = 365           # how long to certify for
default_crl_days= 30            # how long before next CRL
default_md = sha256           # which md to use.
preserve    = no            # keep passed DN ordering

# A few difference way of specifying how similar the request should look
# For type CA, the listed attributes must be the same, and the optional
# and supplied fields are just that :-)
policy      = policy_

####################################################################
[ CA_root ]

dir             = INFA_sampleRootCA        # Where everything is kept
certs           = $dir/ca.db.certs      # Where the issued certs are kept
#crl_dir                = $dir/crl      # Where the issued crl are kept
database        = $dir/ca.db.index      # database index file.
new_certs_dir   = $dir/ca.db.certs         # default place for new certs.

certificate     = $dir/root-ca.crt  # The CA certificate
serial          = $dir/ca.db.serial     # The current serial number
#crl            = $dir/crl.pem          # The current CRL
private_key     = $dir/root-ca.key      # The private key
RANDFILE        = $dir/.rand            # private random number file

x509_extensions = v3_ca              # The extentions to add to the cert

# Comment out the following two lines for the "traditional"
# (and highly broken) format.
name_opt        = ca_default            # Subject Name options
cert_opt        = ca_default            # Certificate field options

# Extension copying option: use with caution.
# copy_extensions = copy

# Extensions to add to a CRL. Note: Netscape communicator chokes on V2 CRLs
# so this is commented out by default to leave a V1 CRL.
# crl_extensions        = crl_ext

default_days    = 1095                   # how long to certify for
default_crl_days= 30                    # how long before next CRL
default_md = sha256                   # which md to use.
preserve        = no                    # keep passed DN ordering

# A few difference way of specifying how similar the request should look
# For type CA, the listed attributes must be the same, and the optional
# and supplied fields are just that :-)
policy          = policy_match
```

```
unique_subject = yes

# For the CA policy
[ policy_match ]
countryName     = optional
stateOrProvinceName = optional
localityName            = optional
organizationName    = optional
organizationalUnitName  = optional
commonName      = supplied
emailAddress        = optional

# For the 'anything' policy
# At this point in time, you must list all acceptable 'object'
# types.
[ policy_anything ]
countryName     = optional
stateOrProvinceName = optional
localityName        = optional
organizationName    = optional
organizationalUnitName  = optional
commonName      = supplied
emailAddress        = optional

####################################################################
[ req ]
default_bits        = 2048
default_keyfile     = privkey.pem
distinguished_name  = req_distinguished_name
attributes      = req_attributes
x509_extensions = v3_ca # The extentions to add to the self signed cert

# Passwords for private keys if not present they will be prompted for
# input_password = secret
# output_password = secret

# This sets a mask for permitted string types. There are several options.
# default: PrintableString, T61String, BMPString.
# pkix   : PrintableString, BMPString.
# utf8only: only UTF8Strings.
# nombstr : PrintableString, T61String (no BMPStrings or UTF8Strings).
# MASK:XXXX a literal mask value.
# WARNING: current versions of Netscape crash on BMPStrings or UTF8Strings
# so use this option with caution!
string_mask = nombstr

# req_extensions = v3_req # The extensions to add to a certificate request

[ req_distinguished_name ]
countryName         = Country Name (2 letter code)
countryName_default     = US
countryName_min         = 2
countryName_max         = 2

stateOrProvinceName     = State or Province Name (full name)
stateOrProvinceName_default = SampleProvince

localityName            = Locality Name (eg, city)
localityName_default        = Madison

0.organizationName      = Organization Name (eg, company)
0.organizationName_default  = SampleOrg

1.organizationName      = Second Organization Name (eg, company)
1.organizationName_default  = Computer Sciences Department

organizationalUnitName      = Organizational Unit Name (eg, section)
organizationalUnitName_default  = INFA_sample Project

commonName          = Common Name (eg, YOUR name)
commonName_max          = 64
```

```
emailAddress            = Email Address
emailAddress_max        = 64

# SET-ex3           = SET extension number 3

[ req_attributes ]
# challengePassword     = A challenge password
# challengePassword_min     = 4
# challengePassword_max     = 20

# unstructuredName      = An optional company name

[ usr_cert ]

# These extensions are added when 'ca' signs a request.
basicConstraints = CA:FALSE
keyUsage = digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth
nsComment = "Issued by DSG Data Vault Product for Testing"
subjectAltName = @alt_names
# This goes against PKIX guidelines but some CAs do it and some software
# requires this to avoid interpreting an end user certificate as a CA.

basicConstraints=CA:FALSE

# Here are some examples of the usage of nsCertType. If it is omitted
# the certificate can be used for anything *except* object signing.

# This is OK for an SSL server.
# nsCertType            = server

# For an object signing certificate this would be used.
# nsCertType = objsign

# For normal client use this is typical
# nsCertType = client, email

# and for everything including object signing:
# nsCertType = client, email, objsign

# This is typical in keyUsage for a client certificate.
# keyUsage = nonRepudiation, digitalSignature, keyEncipherment

# This will be displayed in Netscape's comment listbox.
nsComment           = "OpenSSL Generated Certificate"

# PKIX recommendations harmless if included in all certificates.
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer:always

# This stuff is for subjectAltName and issuerAltname.
# Import the email address.
# subjectAltName=email:copy
# An alternative to produce certificates that aren't
# deprecated according to PKIX.
# subjectAltName=email:move

# Copy subject details
# issuerAltName=issuer:copy

#nsCaRevocationUrl      = http://www.domain.dom/ca-crl.pem
#nsBaseUrl
#nsRevocationUrl
#nsRenewalUrl
#nsCaPolicyUrl
#nsSslServerName


[alt_names]
DNS.1 = kb.example.com
```

```
DNS.2 = helpdesk.example.org
DNS.3 = systems.example.net
IP.1 = 192.168.1.1
IP.2 = 10.74.112.73
IP.3 = 2001:0db8:85a3:0000:0000:8a2e:0370:7334
[ v3_req ]

# Extensions to add to a certificate request

basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment

[ v3_ca ]


# Extensions for a typical CA


# PKIX recommendation.

subjectKeyIdentifier=hash

authorityKeyIdentifier=keyid:always,issuer:always

# This is what PKIX recommends but some broken software chokes on critical
# extensions.
#basicConstraints = critical,CA:true
# So we do this instead.
basicConstraints = CA:true

# Key usage: this is typical for a CA certificate. However since it will
# prevent it being used as an test self-signed certificate it is best
# left out by default.
# keyUsage = cRLSign, keyCertSign

# Some might want this also
# nsCertType = sslCA, emailCA

# Include email address in subject alt name: another PKIX recommendation
# subjectAltName=email:copy
# Copy issuer details
# issuerAltName=issuer:copy

# DER hex encoding of an extension: beware experts only!
# obj=DER:02:03
# Where 'obj' is a standard or added object
# You can even override a supported extension:
# basicConstraints= critical, DER:30:03:01:01:FF

[ crl_ext ]

# CRL extensions.
# Only issuerAltName and authorityKeyIdentifier make any sense in a CRL.

# issuerAltName=issuer:copy
authorityKeyIdentifier=keyid:always,issuer:always
```

SSL Setup in Data Archive

To setup up SSL communication with Data Vault, you must complete configuration tasks from the Data Archive side. These tasks include configuring the `conf.properties` file, editing the `startApplimation.bat` or `startApplimation.sh` files, configuring the source connection to Data Vault, and more.

For more information about configuring SSL for Data Vault from the Data Archive side, see the *Data Archive Administrator Guide*.

Limitations

SSL communication in Data Vault has the following limitations:

1. The md5 algorithm is not supported. This is because Data Vault is certified on Java 1.8. In Java 1.8, the cryptographic hash algorithm MD5 is not considered secure and is no longer be used. Accordingly, this has been deactivated by default in the Oracle JSSE implementation by adding it to the "jdk.tls.disabledAlgorithms" security property. Now, both TLS handshake messages and X.509 certificates signed with this algorithm are no longer acceptable by default. If required, this algorithm can be reactivated by removing "MD5withRSA" and "MD5" from the "jdk.tls.disabledAlgorithms" and "jdk.certpath.disabledAlgorithms" security properties respectively.
2. Java 1.8.0.92 and greater are supported.
3. Subject alternate names do not support wild card characters like "*." For example, "*.informatica.com" is not a valid alternate name.

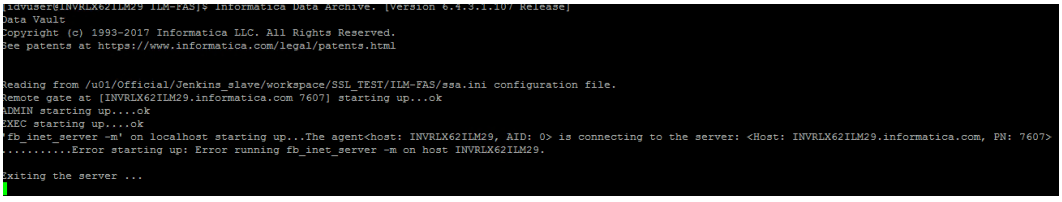
Troubleshooting

The following section describes common troubleshooting scenarios.

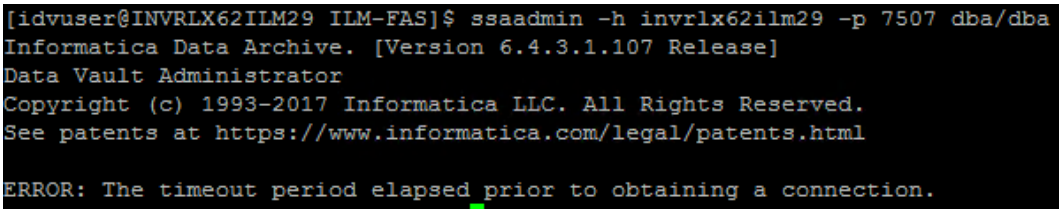
If the server certificate or private key is not correct, the server will not start and returns the following message:

- Certificate file does not exist or file is not valid.
- Private key file does not exist or file is not valid.

If you do not provide the certificate information in the [AGENT] section of the ssa.ini, but provide it in the [SERVER] section, and then try to start the server, the server does not start. You receive the following message:



If you do not provide the certificate information in the [ADMIN] section of the ssa.ini file and try to connect ssaadmin to the host using the hostname and port number, the connection fails with a timeout.



If you do not provide the certificate information, SSL = 1 parameter, or host that is either the common name or an alternate name, in the nucleus.ini file, and try to connect with a client like ssasql, you receive an error. The SQL Worksheet and other ODBC tools will timeout after one minute in this scenario.

```
[idvuser@INVRLX62ILM29 ILM-FAS]$ ssasql fas adm dba/dba
Informatica Data Archive. [Version 6.4.3.1.107 Release]
Data Vault Interactive SQL
Copyright (c) 1993-2017 Informatica LLC. All Rights Reserved.
See patents at https://www.informatica.com/legal/patents.html

Error state 08004.  Server refused connection
```

CHAPTER 5

Data Vault ODBC Setup

This chapter includes the following topics:

- Data Vault ODBC Setup Overview, 68
- Install the Data Vault ODBC Driver Files, 68
- Verify the Target Connection in the Data Archive Workbench, 69
- Create the ODBC Data Source for the Data Vault, 70
- ODBC Connection String Parameters, 74

Data Vault ODBC Setup Overview

You must use the Data Vault ODBC driver to connect to the Data Vault. Install the Data Vault ODBC driver on the machine that hosts a client application that you want to use to connect to the Data Vault.

You can use any ODBC-compliant client application to access the data stored in the Data Vault. For example, you can use PowerCenter to connect to the Data Vault as a source or target. On Windows, you can use the Data Vault SQL Tool to access data in the Data Vault.

To use the Data Vault ODBC driver to connect to the Data Vault, complete the following steps:

1. Install the Data Vault ODBC files on the machine that hosts the client application.
2. Verify the connections to the Data Vault that are available in the Data Archive Workbench.
3. Create the ODBC data source.

Install the Data Vault ODBC Driver Files

To connect to the Data Vault with ODBC, install the Data Vault ODBC files on the machine that hosts the ODBC client application.

Use the Data Vault installer to install the ODBC driver. Install one of the following Data Vault components to get the ODBC driver files:

Data Vault ODBC driver

You can install the ODBC driver on Windows or UNIX.

Data Vault Administration Tool and Data Vault SQL Tool

You can install the Data Vault SQL Tool on a Windows client machine and use the Data Vault SQL Tool to connect to the Data Vault.

For more information about installing the Data Vault ODBC driver, see the *Data Archive Installation and Upgrade Guide*.

On Windows, the Data Vault installer copies the ODBC driver files to the following directories:

- 32-bit Windows: `<Component Directory>/odbc32`
- 64-bit Windows: `<Component Directory>/odbc`

On UNIX, the Data Vault installer copies the ODBC driver files to the following directory:

`<Component Directory>/odbc`

Verify the Target Connection in the Data Archive Workbench

Before you create the ODBC data source, verify that Data Archive contains a target connection to the Data Vault that you can use for the ODBC data source.

You must use a Data Archive target connection of type Data Vault in the ODBC data source. You can use the same target connection for multiple ODBC data sources.

The connection, host name, and port number parameters in an ODBC data source must match the connection, host name, and port number properties of the Data Archive target connection. In the Data Archive Workbench, view the Data Vault connection you want to use and verify the connection properties.

The following table describes the target connection properties that you need to verify for the ODBC data source:

Property	Description
Connection Name	Name of the connection to the Data Vault. Use the connection name in the ODBC connection parameter.
Data Vault Archive Folder Name	Name of the folder for the archived data in the Data Vault. Use the Data Vault archive folder name in the ODBC database parameter. Folder name cannot exceed 128 characters.
Data Vault Host	Host name or IP address of the machine that hosts the Data Vault. Use the Data Vault host in the ODBC host parameter.
Data Vault Port	Port number to connect to the Data Vault Service. Use the Data Vault port in the ODBC port parameter.
Data Vault User	User account to connect to the Data Vault. On Windows, use the Data Vault user in the ODBC user parameter. On UNIX, use the Data Vault user in the ODBC UID parameter.

Create the ODBC Data Source for the Data Vault

When you create the ODBC data source, use the name of a Data Vault archive folder as the name of the database in the ODBC data source.

The Data Vault Service stores archived data in a Data Vault archive folder in the Data Vault. The Data Vault archive folder in the Data Vault is equivalent to the database in the ODBC data source. When you create the ODBC data source for the Data Vault, set the database property to the name of Data Vault archive folder you want to access.

Creating an ODBC Data Source on Windows

You can use the Windows ODBC Data Source Administrator or the SQL Worksheet to create the ODBC data source. The Windows Data Source Administrator and the SQL Worksheet ODBC manager require the same parameters.

When you create the data source, the Windows ODBC Data Source Administrator and the SQL Worksheet ODBC manager store the ODBC driver parameters in the Windows registry.

Using the Windows ODBC Data Source Administrator to Create a Data Source

Windows provides an administrative tool to create and manage ODBC data sources.

1. From the Start Menu, select **Administrative Tools** > **Data Sources (ODBC)**.
2. In the ODBC Data Source Administrator, add a DSN.
3. Select the following driver: **Nucleus ODBC Driver**
4. Set the parameters for the Data Vault ODBC driver.

Using the SQL Worksheet to Create a Data Source

The SQL Worksheet uses the Data Vault ODBC driver to connect to the Data Vault.

1. Go to the Start Menu and select **Programs** > **Informatica Data Vault Service** > **SQL Worksheet**.
2. In the Connect ODBC window, click Configure.
3. In the Configuration window, add a DSN.
4. Set the parameters for the Data Vault ODBC driver.

   By default, the SQL Worksheet ODBC manager uses the Data Vault ODBC driver for the data source. You do not have to select an ODBC driver.

Data Vault ODBC Data Source Parameters

When you create an ODBC data source that uses the Data Vault ODBC driver, you must configure the parameters required for the Data Vault ODBC driver.

The following table describes the parameters that you set for a data source that uses the Data Vault ODBC driver:

ODBC Parameter	Description
Data Source Name	Name of the ODBC data source.
Description	Additional information to describe the data source.
Database name	Name of the folder in the Data Vault to connect to.
Database user	User account to log in to the Data Vault. The user account must be defined in Data Archive.
Connection name	Name of a Data Archive connection to the Data Vault. The connection name must match a target connection defined in the Data Archive Workbench.
Host	Host name of the machine that hosts the Data Vault Service.
Port	Port number to connect to the Data Vault.

The following table describes the advanced parameters that you can set for a data source that uses the Data Vault ODBC driver:

ODBC Parameter	Description
Code page	The code page to use for code page translations between the client and the Data Vault. Select from one of the following options: - 1252 (ANSI - Latin I) - 65001 (UTF-8)
Date, Time, Number formats	Format to use for date, time, and numeric values. Select from one of the following options: - ISO - Regional settings
Autorefresh	Configures the Data Vault ODBC driver to commit or roll back all transactions before each SELECT statement so that the client accesses the most recent version of the Data Vault archive folder. Select the Autorefresh parameter and then set the parameter to one of the following options: - Commit. The ODBC driver commits all transactions before the next query. - Rollback. The ODBC driver rolls back all transactions before the next query. By default, the Autorefresh parameter is not enabled and the ODBC driver does not commit or roll back transactions before a SELECT statement.
Autocommit	Configures the Data Vault ODBC Driver to commit transactions after each SQL command that modifies data such as an UPDATE or INSERT. Select the parameter to enable the ODBC driver to commit transactions after each SQL command that modifies data. By default, the Autocommit parameter is not enabled and ODBC Driver does not commit transactions after each SQL command that modifies data.

Creating an ODBC Data Source on UNIX

On UNIX, you can use any ODBC driver manager such as iODBC or unixODBC to create an ODBC data source. The ODBC driver manager adds an entry for the ODBC data source definition to the odbc.ini file.

You can also manually add the ODBC data source definition to the odbc.ini file. On UNIX, all ODBC data source definitions are stored in the odbc.ini file. You can add or modify an ODBC data source definition in the odbc.ini file without using the ODBC driver manager interface.

Typically, the odbc.ini file is located in the user /etc directory. For example, for unixODBC, the odbc.ini file is in the following directory:

```
/usr/local/etc
```

To verify the location of the odbc.ini file, check the ODBCINI environment variable.

Data Vault ODBC Driver

Use the correct Data Vault ODBC driver for the 32-bit or 64-bit operating system. The 32-bit and 64 bit ODBC driver files have the different names and are installed in the same directory.

The following table describes the Data Vault ODBC driver files on UNIX:

Operating System	ODBC Driver
32-bit operating system	<Data Vault ODBC Driver Directory>/odbc32/libsando.so
64-bit operating system	<Data Vault ODBC Driver Directory>/odbc/libsando64.so

Data Vault ODBC Data Source Parameters

When you create an ODBC data source that uses the Data Vault ODBC driver, you must configure the parameters required for the Data Vault ODBC driver.

The following table describes the parameters that you set for the Data Vault ODBC driver in the odbc.ini file:

ODBC Parameter	Description
Data Source Name	Name of the ODBC data source. Use as heading for the data source definition in the following format: `[Data Source Name]`
DESCRIPTION	Additional information to describe the data source.
DRIVER	Path and file name of the Data Vault ODBC Driver file.
DATABASE	Name of the folder in the Data Vault to connect to.
UID	User account to log in to the optimized file archive. The user account must be defined in Data Archive.
CONNECTION	Name of a Data Archive connection to the optimized file archive. The connection name must match a target connection defined in the Data Archive Workbench.
HOST	Host name of the machine that hosts the Data Vault.

ODBC Parameter	Description
PORT	Port number to connect to the Data Vault.
UTF8Data	The code page to use for code page translations between the client and the optimized file archive. Set the parameter to one of the following options: - True. Sets the code page to 65001 (UTF-8). - False. Sets the code page to 1252 (ANSI - Latin I).
UseRegionalSettings	Format to use for date, time, and numeric values. Set the parameter to one of the following options: - True. Sets the date, time, and numeric settings to regional setting. - False. Sets the date, time, and numeric settings to ISO.
AUTOREFRESH	Configures the Data Vault ODBC driver to commit or roll back all transactions before each SELECT statement so that the ODBC client accesses the most recent version of the Data Vault archive folder. Set the parameter to one of the following options: - COMMIT. The ODBC driver commits all transactions before the next query. You can also set the - ROLLBACK. The ODBC driver rolls back all transactions before the next query. - 0. The Autorefresh parameter is not enabled and the ODBC driver does not commit or roll back transactions before a SELECT statement. Default is 0. The Autorefresh parameter is not enabled and the ODBC driver does not commit or roll back transactions before a SELECT statement.
AUTOCOMMIT	Configures the Data Vault ODBC Driver to commit transactions after each SQL command that modifies data such as an UPDATE or INSERT. Set the Autocommit parameter to one of the following options: - True. The ODBC driver commits all transactions after an SQL command that modifies data. - False. The ODBC driver does not commit transactions after an SQL command that modifies data. Default is 0. The Autocommit parameter is not enabled and ODBC Driver does not commit transactions after each SQL command that modifies data.

odbc.ini File

On UNIX, the Data Vault ODBC driver uses the information in the odbc.ini file to determine how to connect from an ODBC client to the Data Vault.

The odbc.ini file must include a section for each ODBC data source definition. The section starts with a header that specifies the name of the DSN. The section includes the ODBC parameters defined for the data source.

The following ODBC entry for the *OptFileArchive* data source shows an example of a DSN that uses the Data Vault ODBC driver:

```
[OptFileArchive]
DRIVER=/usr/FAS/libsando.so
DATABASE=FASfolder100
CONNECTION=server1_dbtest
HOST=myhost
PORT=8600
UTF8Data=1
UseRegionalSettings=1
AUTOREFRESH=1
AUTOCOMMIT=0
```

ODBCINI Environment Variable

The ODBCINI environment variable stores the location of the ODBC.ini file.

You can view or set the value of the ODBCINI variable in the initialization file for the UNIX shell that you use. For example, on the Korn or Bourne shell, you can set the ODBCINI variable in the .profile file. On the C shell, you can set the variable in the .cshrc file.

To set the ODBCINI variable, add the following command to the shell initialization file:

.profile

```
ODBCINI=/<Path and file name of the odbc.ini file>
export ODBCINI
```

.cshrc

```
setenv ODBCINI /<Path and file name of the odbc.ini file>
```

ODBC Connection String Parameters

You can include the ODBC parameters in the connection string that a client application uses to connect to the Data Vault. Use semicolons (;) to separate the connection parameters in the connection string.

The following table describes the parameters that you can include in the connection string:

Parameter	Description
DRIVER	Path and file name of the Data Vault ODBC driver. Alternatively, you can set the driver parameter to the ODBC driver name in the following format: DRIVER={NUCLEUS ODBC Driver}
DATABASE	Name of the folder in the Data Vault to connect to.
UID	User account to log in to the Data Vault. The user account must be defined in Data Archive.
CONNECTION	Name of a Data Archive connection to the Data Vault. The connection name must match a target connection defined in the Data Archive Workbench.
HOST	Host name of the machine that hosts the Data Vault.
PORT	Port number to connect to the Data Vault.
UTF8DATA	The code page to use for code page translations between the client and the Data Vault. Set the parameter to one of the following options: - True. Sets the code page to 65001 (UTF-8). - False. Sets the code page to 1252 (ANSI - Latin I).
USEREGIONALSETTINGS	Format to use for date, time, and numeric values. Set the parameter to one of the following options: - True. Sets the date, time, and numeric settings to regional setting. - False. Sets the date, time, and numeric settings to ISO.

Parameter	Description
AUTOREFRESH	Configures the Data Vault ODBC driver to commit or roll back all transactions before each SELECT statement so that the ODBC client accesses the most recent version of the Data Vault archive folder. Set the parameter to one of the following options: - COMMIT. The ODBC driver commits all transactions before the next query. You can also set the - ROLLBACK. The ODBC driver rolls back all transactions before the next query. - 0. The Autorefresh parameter is not enabled and the ODBC driver does not commit or roll back transactions before a SELECT statement.
AUTOCOMMIT	Configures the Data Vault ODBC Driver to commit transactions after each SQL command that modifies data such as an UPDATE or INSERT. Set the Autocommit parameter to one of the following options: - True. The ODBC driver commits all transactions after an SQL command that modifies data. - False. The ODBC driver does not commit transactions after an SQL command that modifies data.
PWD	Password for the user account specified by the UID parameter.

Alternatively, you can include the data source name (DSN) in the connection string.

To use the DSN in the connection string, add one of the following parameters to the connection string.

- DSN. Indicates a user or system DSN. Use the following format:

  ```
  DSN=Data Source Name
  ```

- FILEDSN. Indicates a file DSN. Use the following format:

  ```
  FILEDSN=Data Source Name
  ```

Parameters set in the connection string take precedence over parameters set in the data source. If the value of the parameter in the connection string is different from the value of the same parameter in the data source, the client application uses the value in the connection string. For example, if the value of UID in the data source is *DBA* but the connection string contains *UID=BOB*, the client application uses the value of the connection string UID parameter, *BOB*.

The following text shows an example of the Data Vault ODBC parameters in a connection string:

```
DRIVER=C:\WINNT
\System32\sando.dll;CONNECTION=serv1_db01;DATABASE=db01;AUTOCOMMIT=True;UID=BOB
```

CHAPTER 6

Data Vault Administration

This chapter includes the following topics:

- Data Vault Administration Overview, 76
- System Setup, 77
- System Operations, 81
- Metadata Administration, 86
- Data File Operations, 95
- Data Indexing, 95
- Data Vault Performance Tuning, 100
- Data Vault Cleanup, 101
- Column Definitions, 102
- Materialized Views, 103

Data Vault Administration Overview

You can use the Data Vault administration command line program to manage the Data Vault.

The following table describes the administration tasks you can perform with the Data Vault administration command line program:

Task	Description
System setup	- Set or view Data Vault parameters. - Display information about Data Vault tables. - Manage the log files.
System operations	- Shut down the Data Vault load balancer. - Terminate the Data Vault Agent processes. - Change the Data Vault Agent priority levels. - Set or remove the Data Vault Agent host preferences. - View information about the load balancer, hosts, host preferences, Data Vault Agent processes, and the Data Vault repository. - Display or reset load balancer statistics. - Shut down the Data Vault components.

Task	Description
Data Vault operations	- Create, alter, renew, or drop indexes on tables in the Data Vault. - Repartition tables in the Data Vault.
Session administration	- Cancel active client queries. - Close client connections. - Display a list of Data Vault data files accessed by a query. - View information about clients, active queries, and active tasks.
Metadata administration	- Update metadata in the Data Vault repository. - Check metadata and Data Vault data files for errors. - Remove unreferenced objects from the metadata. - Discard changes to metadata. - View information about Data Vault table columns, domains, and minimum and maximum values.
Data file operations	- Check Data Vault data file integrity. - Configure batch mode command execution

System Setup

This section provides information on setting up the system.

Displaying a List of Archived Tables

Use the Tables command to display a list of all archived tables within the Data Vault repository.

```
Tables
```

Each table appears in the form <dbname>.<schema>.<table>, along with a table ID.

Displaying Data Files in the Data Vault

The Files command displays a list of Data Vault data files registered with the Data Vault repository.

```
Files [ <table-ID> ]
```

The returned information includes the internal Table ID and File ID, table type, the full path of the Data Vault data file, the number of rows in the table, the size of the file, and whether the file is encrypted. All registered objects are displayed by default. To view only the files registered for a specific archived table, include the <table-ID> parameter. Table ID values can be determined using the Tables command.

Displaying Information About Files in the Data Vault

The FileReport command displays information about the specified file in the Data Vault.

```
FileReport <file-id>
```

The following table describes the Data Vault information:

Parameter	Description
ID	The internal file ID for the file in the Data Vault.
Columns	The number of columns in the table specified by the file ID.
Domains	The number of domains used by columns in the table specified by the file ID.
Row Length	The maximum length of a record in the file specified by the file ID (in bytes).
File Size	The total size of the file (in bytes) specified by the file ID.
Total Rows	The total number of records stored in the file specified by the file ID.
Flat file size	The estimated uncompressed size of the data in the file specified by the file ID (in bytes).
Compression ratio	The estimated degree of data compression in the file specified by the file ID.

Displaying Information About Archived Tables

The TableReport command displays information about the specified archived tables.

```
TableReport [ <table-id> | <schema>.<table> ]
```

The following table provides information that appears when you run the TableReport command for a specific archived table:

Parameter	Description
ID	The internal archived table ID.
Name	The name of the archived table, qualified by schema.
Columns	The number of columns in the archived table.
Row length	The maximum length of a record in the archived table (in bytes).
Files	The total number of files registered for this archived table.
Total size	The total size of all files registered for this archived table (in bytes).
Total rows	The total number of rows in the archived table (the combined number of rows in all files registered for this table).
Flat file size	The estimated uncompressed size of the data in the archived table (in bytes).
Compression ratio	The estimated degree of data compression in the archived table.

The following table provides information that is displayed by the TableReport command for all registered archived tables:

Parameter	Description
Total tables	The total number of archived tables.
Files	The total number of files registered with archived tables.
Total size	The total size of all files registered with archived tables (in bytes).
Total rows	The total number of rows across all registered archived tables (the combined number of rows in all files).
Flat file size	The estimated uncompressed size of the data in all registered archived tables (in bytes).
Compression ratio	The estimated degree of data compression in all registered archived tables.

Setting Administration Parameters

The Set command is used to configure Data Vault administration parameters.

```
Set <parameter> <value>
```

The administration parameters and their possible values are the following:

- Detailed_Log { ON | OFF } – enable or disable the writing of extra information to the load balancer log (OFF by default)
- Encrypt_Key *hex-value* – set the encryption key in hexadecimal form. At run time, this value overrides the CRYPTOKEY parameter in the [COMMON] section of the `ssa.ini` file.
- File_Pref { ON | OFF } – enable or disable host preferences for the Data Vault (ON by default)
- Task_Timeout *n* – set the timeout value for administration tasks (in milliseconds) (60000 by default)

Displaying Administration Parameter Values

The Get command displays the current value of the specified administration parameter.

```
Get <parameter>
```

The administration parameters are the following:

- Detailed_Log – whether the writing of extra information to the load balancer log is enabled (ON) or disabled (OFF)
- File_Pref – whether Data Vault host preferences is enabled (ON) or disabled (OFF)
- Task_Timeout – the timeout value (in milliseconds) for administration tasks

Setting the Length of the Column Display

The MaxLength (or Maxlen) command sets the maximum character length for columns displayed in the administration command output.

```
MaxLength <length>
-or-
Maxlen <length>
```

The specified length can either be **0** (which is a shorthand for the maximum possible length), or an integer value from **5** to **256** inclusive. The default is **40** characters. Administration commands that display columnar information will show, at most, the MaxLength number of characters for each column. If information exceeds this maximum length, the display of this information will be truncated.

Note: Any changes to the MaxLength setting applies only to the current administration command session.

Configuring the Server Log

The Log command enables or disables the load balancer log, and sets the level of logging detail. The information is written to a text file (**ssa-**<timestamp>**.log**) in the directory specified by the LOGDIR parameter in the [SERVER] section of the ssa.ini file.

```
Log <value>
```

The possible log values are the following:

- 0. disables the load balancer log.
- 1. enables logging, with the detail level set to LOW (errors and important messages only)
- 2. enables logging, with the detail level set to HIGH (level **1** + more detailed messages)
- 3. enables logging, with the detail level set to TRACE (level **2** + tracing messages)

  Note: Level **3** produces the same information as the trace log (see the TraceLog command below).

Enabling or Disabling the Trace Log

The TraceLog command enables or disables the system trace log. The trace log (**ssa_trace-**<timestamp>**.log**) is created in the directory specified by the LOGDIR parameter in the [SERVER] section of the ssa.ini file.

```
TraceLog <state>
```

The state of the trace log is either enabled (**1**) or disabled (**0**).

The information written to the trace log includes the current section of code being executed, the current system layer, parameter values passed to functions, and other similar information.

Displaying the Version Number of a Component

The version command allows you to view the version of a Data Vault component.

```
<Data Vault component name> --version
```

You can view the version of each component separately, or you can view the versions of all components.

The following table describes how to verify the version for each Data Vault component:

Data Vault Component	How to Verify the Version
Data Vault server	From the root installation directory of the component, run the following command: `ssaserver --version`
Data Vault Agent	From the root installation directory of the component, run the following command: `ssaagent --version`

Data Vault Component	How to Verify the Version
Data Vault ODBC driver	Locate the version in the .dll file properties. - 64-bit drivers on Windows: View the `install_dir/ODBC/sando.dll` and `install_dir/ODBC/sandos.dll` files. - 32-bit drivers on Windows: View the `install_dir/ODBC32/sando.dll` file.

Data Vault Plug-in	The Data Vault Plug-in is a package of multiple Data Vault components. To determine the version of each component, run the following commands:
	`ssaadmin --version`
	`ssau --version`
	`ssasql --version`
	`ssadrv --version`
	To determine the ODBC version, refer to the instructions in this table for the Data Vault ODBC driver.

Data Vault Administrator Tool and SQL Tool	From the root installation directory of the component, run the following command: `ssaadmin --version`

To view the versions of all Data Vault components, perform the following steps:

- From the root Data Vault installation directory, run the following command to connect to the Data Vault server:

  ```
  ssaadmin dba
  ```

- Run the following ssaadmin command to view list of components and corresponding versions:

  ```
  version
  ```

The ssaadmin command utility communicates with the server. The server gets the version for each component, even if the components are on separate machines.

System Operations

This section describes the system operations.

Shutting Down the Load Balancer

Use the Shutdown command to shut down load balancer.

```
Shutdown
```

If the Data Vault repository database was set to start with load balancer, it will be shut down as well. Note that Data Vault Agent processes have to be shut down manually, which can be done using the Stop command.

Disconnecting from the Data Vault

Use the Exit or Quit command to close the Data Vault administration command connection to the Data Vault and shut down the Data Vault administration command program. Optionally, the Commit keyword can be included to save the latest changes to the Data Vault repository.

```
Exit [ Commit ]
-or-
Quit [ Commit ]
```

If the Commit keyword is omitted and there are unsaved changes, the user will be prompted to save or discard the changes when exiting.

Stopping an Agent Process

The Stop command stops the specified Data Vault Agent process. If the process is in the middle of executing a task when the Stop command is issued, the process will be allowed to complete the task before it is ended.

```
Stop <host-name> <agent-id>
```

The <host-name> parameter is the host machine where the Data Vault Agent process is running, and the <agent-ID> parameter refers to the numeric value that identifies the agent process. Agent IDs can be determined by executing the Current command and viewing the AID field in the Agents list.

Note: An agent process with an ID of 0 cannot be terminated with the Stop command, as processes with this ID are responsible for Data Vault repository database startup/shutdown when autostartup is enabled.

Killing an Agent Process

The Kill command stops the specified Data Vault Agent process immediately, even if it is currently executing a task.

```
Kill <host-name> <agent-id>
```

The <host-name> parameter is the host machine where the Data Vault Agent process is running, and the <agent-ID> parameter refers to the numeric value that identifies the agent process. Agent IDs can be determined by executing the Current command and viewing the AID field in the Agents list.

Note: An agent process with an ID of 0 cannot be terminated with the Kill command, as processes with this ID are responsible for Data Vault repository database startup/shutdown when autostartup is enabled.

Changing the Priority Level for an Agent

The Priority command changes the priority level for a Data Vault Agent process.

```
Priority <host-name> <agent-ID> <priority-level>
```

The <host-name> parameter specifies the machine where the process is running. The <agent-ID> parameter refers to the Agent process ID, which can be viewed by executing the Current command; the Agent ID is indicated by the AID field in the Agents list. The <priority-level> parameter is the new priority level for the Agent process.

You can set the priority level to a number between zero and and two million. A higher number indicates a higher priority. If you set the agent priority level to 0 or 1, the Data Vault Service assigns a specific status to the agent.

Priority levels 0 and 1 indicate the following statuses:

- Priority level 0. Indicates that the agent is not available to process any task. The Data Vault Service does not assign a task to an agent with a priority level of 0. To make the agent available to process tasks, change the priority level to a value higher than 0.
- Priority level 1. Indicates that the agent primarily processes high-priority tasks such as registering data files or other administration tasks. The Data Vault Service can assign query tasks to an agent with priority level 1 if the agent is not busy and the Data Vault Service receives a high volume of queries.

If the agent is performing a task when you change the priority level from 0 or 1 to a higher priority, the agent does not change status until it completes the task.

Displaying the Current State

The Current (or Cur) command displays the current state of the Data Vault system in terms of host machines, agent processes/tasks, and Data Vault Service administrators.

```
Cur[rent]
```

The information displayed is identical to what is returned by the Hosts, Agents, Tasks, and Admins commands.

Displaying Host State

The Hosts command displays information about each computer in the Data Vault system with one or more running Data Vault Agent processes:

```
Hosts
```

The following table describes information that is displayed for each host:

Information	Description
Name	The name of the host machine.
Agents	The number of Data Vault Agent processes that are running on the host machine.
Running Tasks	The number of tasks currently being executed by Data Vault Agent processes on the host machine.
Idle Time	The total amount of time the host machine has been inactive (no Data Vault Agent processes are executing tasks).
Running Time	The total amount of time the host machine has been active (one or more Data Vault Agent processes are executing tasks).

Displaying Agent State

The Agents command displays information about each Data Vault Agent process started in the Data Vault system:

```
Agents
```

The following table describes information that is displayed for each Data Vault Agent process:

Information	Description
Host	The machine where the Data Vault Agent process is running.
AID	The Agent ID, a numeric ID that identifies the Data Vault Agent process. This is either a user-defined value, the operating system process ID (PID) for the agent process, or **0**.
Priority	The priority level for the Agent. This value can be changed with the Priority command.
State	The state of the Data Vault Agent process. It is either READY (waiting to execute) or the internal ID of the task being executed.

Displaying Task State

The Tasks command displays information about the tasks that are queued for execution or are currently being executed by Data Vault Agent processes:

```
Tasks
```

The following table describes information that is displayed for each task:

Information	Description
Task	The internal task ID.
Query	The query ID for the task. This query ID can be used in the Cancel command to terminate the query.
File	The unique file ID for the file currently being acted upon by the task, if applicable. This file ID can be specified with the Files command to view information about the file.
Command	The command line used to execute each Data Vault Agent task. Note that all passwords are concealed.
State	The state of the task. It is either Wait (queued for execution) or the host name and AID of the Data Vault Agent process executing the task.
Waiting Time	The total amount of time the task has been queued for execution, if it is not running.
Running Time	The total amount of time that the task has been running, if it is not queued for execution.
Fetched	The total number of rows fetched by a query so far.
Pref Hosts	The preferred host for task execution. This means that Agent processes on this machine will be preferred over other Agent processes for executing the task. The task will be executed by an Agent process on another machine only if all the Agents on the preferred host are busy. Host preference is set with the SetPref command.

Displaying Administrator State

The Admins command displays information about each Data Vault Administration Tool session started in the Data Vault system:

```
Admins
```

The following table describes information that is displayed for each Data Vault Administration Tool session:

Information	Description
Host	The name of the host machine where the Data Vault Administration Tool session was started.
AID	The operating system process ID for the Data Vault Administration Tool session.
Mode	Whether the Data Vault Administration Tool session was started in DBA Mode or Public Mode.

Displaying Load Balancer Statistics

The Stat command displays load balancer statistics.

```
Stat
```

The following table describes information that is displayed for each machine in the Data Vault system:

Information	Description
Name	A host machine where at least one registered Data Vault Agent process.
MaxAgents	The highest number of Data Vault Agent processes that have run concurrently on the specified host.
Agents	The number of Data Vault Agent processes running on the specified host.
Tasks	The total number of tasks completed by Data Vault Agent processes on the current machine.
ExeTime	The total amount of time spent executing tasks.
IdleTime	The total amount of idle time.
Volume	The total amount of data fetched for queries (in bytes).
AvResponseTime	The average length of time it takes for a task to be executed after it is queued for execution.
AvExeTime	The average execution time for each task.

Note: The load balancer statistics can be reset using the ResetStat command.

Resetting Load Balancer Statistics

The Rest Stat command resets the load balancer statistics that can be viewed with the Stat command.

```
ResetStat
```

When ResetStat is executed, the *Tasks*, *ExeTime*, *IdleTime*, *Volume*, *AvResponseTime*, and *AvExeTime* fields are all reset to zero for each machine.

Setting Host Preference and Priority Level

The SetPref command establishes a host preference and priority level for a particular Data Vault data file. This means that the load balancer will attempt to use Data Vault Agent processes running on the specified

machine before any other to carry out tasks involving the Data Vault data file. Other Agent processes are only used if all the Agents on the preferred host are unavailable.

```
SetPref < file-ID> <host-name> <priority-level>
```

The priority level, which can be any positive integer, is used when multiple machines are set as preferred hosts for a particular Archive: the machines are checked for available Agent processes in the order indicated by the respective priority levels (where a larger value means higher priority). Host preferences can be removed with the DropPref command.

Removing Host Preferences

The DropPref command removes the host preferences for a particular data file in the Data Vault.

```
DropPref <file-ID> [ <host-name> ]
```

By default, all host preferences for an object are removed. To remove a specific host preference, include the host name as a parameter. Host preferences are set with the SetPref command.

Displaying a List of Preferred Hosts with Priority Levels

The Prefs command displays a list of preferred hosts and their priority levels.

```
Prefs [ <file-ID> ]
```

Using the SetPref command, a machine can be specified as a preferred host for a data file, meaning load balancer will attempt to use the Data Vault Agent processes running on that machine before any other to carry out tasks involving the file. A priority level is also set for each preferred host, so if multiple machines are designated as preferred hosts for a particular object, the machines are checked for available Agent processes in the order indicated by the respective priority levels (where a higher priority value means greater priority). If Agent processes on a particular machine have variable priority levels, the highest priority available Agent is used.

By default, the Prefs command displays all preferred hosts. To view only the preferred hosts for a specific compressed data file, include the <file-ID> parameter. File ID values can be determined using the Files command.

Displaying Information About Metadata Database Connection

The MDCon command displays information about the Data Vault repository database, including database name, the connection used to start the database, the authorization name used to log in to the database, and the current DATE, TIME, and TIMESTAMP pictures.

```
MDCon
```

Metadata Administration

This section describes the Metadata Administration tasks.

Checking Metadata Integrity

The Check command checks the metadata for internal inconsistencies. This applies to the metadata information in the Data Vault and the Data Vault repository.

```
Check [ FULL ] [ TABLES <table ID list> | FILES <file ID list> | ALL ]
```

Specify TABLES to check the validity of selected archived tables in the Data Vault repository, FILES to check the validity of selected data files registered with the Data Vault repository, or ALL to check all metadata information. If there is a problem with the metadata, a diagnostic message describing the nature of the problem will be returned.

When TABLES or FILES is specified, the validity check is limited to the list of, respectively, table IDs (archived tables) or file IDs (data files), separated by whitespace. Note that table IDs can be determined using the Tables command, while file IDs can be determined with the Files command.

If the FULL keyword is included, there is an additional comparison of the metadata in data files against the Data Vault repository for discrepancies. As this comparison can require a significant amount of time, it is not the default behavior.

If Check is executed without any parameters, only a basic validity check of metadata in the Data Vault and the Data Vault repository is performed.

To verify the integrity of data files only, use the CheckSCT command.

If unreferenced objects (for example, data files associated with a non-existent archived tables) are found in the metadata, they can be removed using the Cleanup command.

Remove Unreferenced Objects from the Data Vault Repository

The Cleanup command removes unreferenced objects from the Data Vault repository.

```
Cleanup
```

Since more than one administrator can update the Data Vault repository at the same time, there is a possibly that unintended discrepancies might be introduced: some objects, such as archived tables or data files, could remain in the metadata after they have been unregistered. The Cleanup command deletes the unreferenced objects and restores the consistency of the metadata.

Note: The Cleanup command does not prompt the user for confirmation: unreferenced metadata objects are removed. To view a list of the unreferenced metadata objects without automatic removal, execute the Check ALL command.

Discarding Changes to the Data Vault Repository

The Rollback command discards all changes made to the Data Vault repository since the last Commit command, or since Data Vault Administration Tool session was started if the Commit command was never executed.

```
Rollback
```

Saving Changes to the Data Vault Repository

The Commit command saves the latest changes to the Data Vault repository.

```
Commit
```

Displaying Column Metadata

The Columns command displays information about the columns contained in data files registered with the Data Vault repository. This information includes the File ID of the data file, column name, datatype, precision, scale, and nullability. By default, information about all columns are displayed.

```
Columns [ <file-ID> ]
```

To view only the columns for a specific object, include the <file-ID> parameter. File ID values can be determined using the Files command.

Displaying Metadata Columns

If you need to display or query metadata columns through the SQL worksheet, run the following command:

```
alter session set hidemetafields=0
```

Query results appear for the current user session only.

To hide the metadata columns, run the following command:

```
alter session set hidemetafields=1
```

The default value for the parameter is 1.

Displaying Domain Metadata

The Domains command displays a list of the domains used by data files registered with the Data Vault repository. By default, all domains are displayed.

```
Domains [ <file-ID> ]
```

To view only the domain information for a specific file, include the <file-ID> parameter. File ID values can be determined using the Files command.

Displaying Minimum and Maximum Column Values

Displays the minimum/maximum values for columns in a data file.

```
FileLimits <file-ID> [ <columns-list> ]
```

By default, this information is returned for each column in the object. To limit the information to selected columns, include a list of the column names separated by whitespace. Note that null values will appear as the minimum and maximum only if all values in a column are null.

Displaying the Minimum and Maximum Values for Columns in a Archived Table

The TableLimits command displays the minimum/maximum values for columns in an archived table.

```
TableLimits { <table-id> | <schema>.<table> } [ <column-list> ]
```

The table can be specified by including either the schema-qualified table name or the table ID (which can be viewed using the Tables command). By default, the min/max information is returned for each column in the table. To limit the information to selected columns, include a list of the column names separated by whitespace. Note that null values will appear as the minimum and maximum only if all values in a column are null.

Accessing Catalog Tables

The metadata repository contains catalog tables with information about Data Vault users, tables, privileges, and more. You can query these tables to build reports.

The following list describes the catalog tables:

Table Name

CAT_USERS

Table Description

Contains a list of Data Vault users.

Table Columns

- NAME: VARCHAR (128), nullable
- PASSWORD: VARCHAR (128), nullable
- PASSWORDCHANGE: TIMESTAMP, nullable
- UID: INTEGER, not null
- ROLE: CHAR (1), nullable
- DEFAULTDB: INTEGER, nullable
- DEFAULTSCHEMA: INTEGER, nullable
- CREATOR: INTEGER, nullable
- CREATED: TIMESTAMP, nullable
- ALTERED: TIMESTAMP, nullable
- CONSTRAINT CAT_USERS_PK: Primary key (UID)

Table Name

CAT_DATABASES

Table Description

Contains a list of databases in Data Vault.

Table Columns

- NAME: VARCHAR (256), nullable
- DBID: INTEGER, not null
- CREATOR: INTEGER, nullable
- CREATED: TIMESTAMP, nullable
- ALTERED: TIMESTAMP, nullable
- CONSTRAINT CAT_DATABASES_FK1: Foreign key (CREATOR), references CAT_USERS (UID)
- CONSTRAINT CAT_DATABASES_PK: Primary key (DBID)

Table Name

CAT_SCHEMAS

Table Description

Contains a list of schemas in Data Vault.

Table Columns

- NAME: VARCHAR (256), nullable
- SCHID: INTEGER, not null
- DBID: INTEGER, nullable
- CREATOR: INTEGER, nullable
- CREATED: TIMESTAMP, nullable
- ALTERED: TIMESTAMP, nullable
- CONSTRAINT CAT_SCHEMAS_FK1: Foreign key (CREATOR), references CAT_USERS (UID)
- CONSTRAINT CAT_SCHEMAS_PK: Primary key (SCHID)

Table Name

CAT_TABLES

Table Description

Contains a list of tables in Data Vault.

Table Columns

- NAME: VARCHAR (256), nullable
- TID: INTEGER, not null
- TABLETYPE: CHAR (1), nullable
- ARCHIVETYPE: CHAR (1), nullable
- COMMENTS: VARCHAR (1024), nullable
- DBID: INTEGER, nullable
- SCHID: INTEGER, nullable
- CREATOR: INTEGER, nullable
- CREATED: TIMESTAMP, nullable
- ALTERED: TIMESTAMP, nullable
- CONSTRAINT CAT_TABLES_FK1: Foreign key (CREATOR), references CAT_USERS (UID)
- CONSTRAINT CAT_TABLES_PK: Primary key (TID)

Table Name

CAT_DOMAINS

Table Description

Contains a list of domains in Data Vault.

Table Columns

- NAME: VARCHAR (256), nullable
- DOMID: INTEGER, not null
- DBID: INTEGER, nullable
- SCHID: INTEGER, nullable
- CREATOR: INTEGER, nullable
- DATATYPE: INTEGER, nullable
- PREC: INTEGER, nullable
- SCALE: INTEGER, nullable
- CREATED: TIMESTAMP, nullable
- ALTERED: TIMESTAMP, nullable
- CONSTRAINT CAT_DOMAINS_FK1: Foreign key (CREATOR), references CAT_USERS (UID)
- CONSTRAINT CAT_DOMAINS_FK2: Foreign key (DBID), references CAT_DATABASES (DBID)
- CONSTRAINT CAT_DOMAINS_FK3: Foreign key (SCHID), references CAT_SCHEMAS (SCHID)
- CONSTRAINT CAT_DOMAINS_PK: Primary key (DOMID)

Table Name

CAT_COLUMNS

Table Description

Contains a list of columns in Data Vault.

Table Columns

- NAME: VARCHAR (256), nullable
- COLNO: INTEGER, not null
- FLAG: CHAR (1), nullable
- NULLFLAG: CHAR (1), nullable
- TID: INTEGER, not null
- DOMID: INTEGER, nullable
- CREATED: TIMESTAMP, nullable
- ALTERED: TIMESTAMP, nullable
- CONSTRAINT CAT_COLUMNS_FK1: Foreign key (TID), references CAT_TABLES (TID)
- CONSTRAINT CAT_COLUMNS_FK2: Foreign key (DOMID), references CAT_DOMAINS (DOMID)
- CONSTRAINT CAT_COLUMNS_PK: Primary key (TID, COLNO)

Table Name

CAT_DATATYPES

Table Description

Contains a list of datatypes in Data Vault.

Table Columns

- DATATYPE: INTEGER, nullable
- NAME: VARCHAR (9), nullable
- MAX_LENGTH: DECIMAL (18, 0), nullable
- MAX_SCALE: INTEGER, nullable

Table Name

CAT_VIEWS

Table Description

Contains a list of views in Data Vault.

Table Columns

NAME: VARCHAR (256), nullable

VWID: INTEGER, not null

DBID: INTEGER, nullable

SCHID: INTEGER, nullable

CREATOR: INTEGER, nullable

TYPE: CHAR (1), nullable

QUERYTYPE: CHAR (1), nullable

STATE: CHAR (1), nullable

CREATED: TIMESTAMP, nullable

ALTERED: TIMESTAMP, nullable

CONSTRAINT CAT_VIEWS_FK1: Foreign key (DBID), references CAT_DATABASES (DBID)

CONSTRAINT CAT_VIEWS_FK2: Foreign key (SCHID), references CAT_SCHEMAS (SCHID)

CONSTRAINT CAT_VIEWS_FK3: Foreign key (VWID), references CAT_TABLES (TID)

CONSTRAINT CAT_VIEWS_FK4: Foreign key (CREATOR), references CAT_USERS (UID)

CONSTRAINT CAT_VIEWS_PK: Primary key (VWID)

Table Name

CAT_DATABASE_PRIVS

Table Description

Contains a list of privileges at the database level in Data Vault.

Table Columns

GRANTOR: INTEGER, nullable

GRANTEE: INTEGER, nullable

DBID: INTEGER, nullable

PRIVILEGE: CHAR (1), nullable

GRANTABLE: CHAR (1), nullable

CREATED: TIMESTAMP, nullable

CONSTRAINT CAT_DATABASE_PRIVS_FK1: Foreign key (DBID), references CAT_DATABASES (DBID)

CONSTRAINT CAT_DATABASE_PRIVS_FK2: Foreign key (GRANTOR), references CAT_USERS (UID)

CONSTRAINT CAT_DATABASE_PRIVS_FK3: Foreign key (GRANTEE), references CAT_USERS (UID)

Table Name

CAT_SCHEMA_PRIVS

Table Description

Contains a list of privileges at the schema level in Data Vault.

Table Columns

- GRANTOR: INTEGER, nullable
- GRANTEE: INTEGER, nullable
- SCHID: INTEGER, nullable
- PRIVILEGE: CHAR (1), nullable
- GRANTABLE: CHAR (1), nullable
- CREATED: TIMESTAMP, nullable
- CONSTRAINT CAT_SCHEMA_PRIVS_FK1: Foreign key (SCHID), references CAT_SCHEMAS (SCHID)
- CONSTRAINT CAT_SCHEMA_PRIVS_FK2: Foreign key (GRANTOR), references CAT_USERS (UID)
- CONSTRAINT CAT_SCHEMA_PRIVS_FK3: Foreign key (GRANTEE), references CAT_USERS (UID)

Table Name

CAT_TABLE_PRIVS

Table Description

Contains a list of privileges at the table level in Data Vault.

Table Columns

- GRANTOR: INTEGER, nullable
- GRANTEE: INTEGER, nullable

- TID: INTEGER, nullable
- PRIVILEGE: CHAR (1), nullable
- GRANTABLE: CHAR (1), nullable
- CREATED: TIMESTAMP, nullable
- CONSTRAINT CAT_TABLE_PRIVS_FK1: Foreign key (TID), references CAT_TABLES (TID)
- CONSTRAINT CAT_TABLE_PRIVS_FK2: Foreign key (GRANTOR), references CAT_USERS (UID)
- CONSTRAINT CAT_TABLE_PRIVS_FK3: Foreign key (GRANTEE) , references CAT_USERS (UID)

Table Name

CAT_COLUMN_PRIVS

Table Description

Contains a list of privileges at the column level in Data Vault.

Table Columns

- GRANTOR: INTEGER, nullable
- GRANTEE: INTEGER, nullable
- TID: INTEGER, nullable
- COLNO: INTEGER, nullable
- PRIVILEGE: CHAR (1), nullable
- GRANTABLE: CHAR (1), nullable
- CREATED: TIMESTAMP, nullable
- CONSTRAINT CAT_COLUMN_PRIVS_FK1: Foreign key (TID, COLNO), references CAT_COLUMNS (TID, COLNO)
- CONSTRAINT CAT_COLUMN_PRIVS_FK2: Foreign key (GRANTOR), references CAT_USERS (UID)
- CONSTRAINT CAT_COLUMN_PRIVS_FK3: Foreign key (GRANTEE), references CAT_USERS (UID)

Backup and Restore the Metadata Repository

Use the ssabackup and ssarestore utilities to backup and restore the metadata repository database.

Ssabackup

The ssabackup command line utility creates an online snapshot of the metadata repository while the Data Vault system is running. Ssabackup creates the backup by starting a transaction that spans the backup period.

When the backup is complete, the transaction ends. Any transactions you start after the backup process begins will not have any changed data written to the backup file. The backup represents a copy of the entire database at the moment the backup began.

The backup creates a tar-ball file in a specified backup location, which can then be backed up to tape, NFS storage, or any other file system.

Optionally, ssabackup can also backup the SCT files at the same time. To backup the SCT files, you must specify a flag with the SCT files backup location (such as NFS, NAS, or SAN storage systems), because they are closed at the operating system level once they have been created.

The ssabackup utility uses the following syntax:

```
ssabackup [ <flags> ]
[-h (displays the help screen)]
[-b<dir> (metadata repository backup folder)]
[-r<dir> (temporary directory for log, report, and runtime files)]
[-s<dir> (SCT files backup folder)]
[-u UID[/PWD] (Data Vault user name and password)]
```

For example, the following command generates an online metadata repository backup:

```
./ssabackup -b /u01/idvmeta/backup -r ./temp -u dba/<PWD>
```

The following command generates an online metadata repository backup and a backup of SCT files:

```
./ssabackup -b /u01/idvmeta/backup -s /nfs1/storage1/sct -r ./temp -u dba/<PWD>
```

The ssabackup utility does not backup any SCT files that are located in external storage.

Ssarestore

The ssarestore command line utility restores the metadata repository from the backup repository created by the ssabackup tool. Use ssarestore if the original metadata repository database is unable to operate because of physical or logical damage.

The restore job runs for a specific date. The data filter is used to select the tar-ball backup file with a timestamp in the file name.

You must stop Data Vault before restoring a metadata backup. Any transactions created after the backup was taken will be lost after you restore a backup.

The ssarestore utility uses the following syntax:

```
ssarestore [ <flags> ]
[-h (displays the help screen)]
[-d<yyyy-mm-dd> (specific date for which to restore the archive)]
[-l<location> (archive directory)]
[-n<backup file name> (alternative backup file name: <file name>_<timestamp>.tar.gz)]
[-r <location> (location of the run-time temporary directory)]
[-u UID[/PWD] (Data Vault user name and password)]
```

To restore the metadata repository from a backup created by the ssabackup utility, complete the following tasks:

1. Stop the Data Vault server and agent.
2. If ssabackup ran with the `-s<dir>` flag, copy the SCT files manually from the SCT backup folder to the original file locations.
3. Trigger the ssarestore utility:

   ```
   ssarestore –l <backup folder with the tar-ball file created by ssabackup> -r
   <temp_directory>  -d <date of the tar file to be restored in yyyy-mm-dd format> -u
   <Data Vault user/password>
   ```

   If multiple archive files exist for the same date, use the `-n<backup file name>` flag to give the exact file name that you want to restore.

Run the ssabackup and ssarestore utilities on the same instance.

For example:

```
[idvuser@INVRLX62ILM29 b]$ ssarestore -l /u01/Official/Jenkins_slave/workspace/
IDV642_Lin64/temp/b -n idv_metadata_backup_2016_04_12_10_47_40.tar.gz -r . -u dba/dba -d
2016-04-12
```

Data File Operations

This section describes the Data Vault data file operations.

Checking Data File Integrity

The CheckSCT command checks one or more registered data files for integrity errors.

```
CheckSCT { ALL | TABLES <table ID list> | FILES <file ID list> }
[ LEVEL:{1|2|3} ]
```

The CheckSCT command can be applied to all data files registered in the Data Vault repository (ALL); or can be applied to all of the data files registered with one or more specific tables (TABLES); or can be applied to a list of one or more specific compressed data files (FILES). The TABLES option must specify a list of table IDs, separated by whitespace. Similarly, the FILES option must specify a list of file IDs, separated by whitespace.

Optionally, LEVEL:*n* can be included to set the level of integrity checking. There are three possible levels:

1. Check the metadata only.
2. Check the encoded data only.
3. Perform all checks on the compressed data file.

Generally, the higher the level, the more time is required to complete the CheckSCT command. Depending on the size and number of data files being checked, this command can take quite a long time to finish executing, especially at LEVEL:3.

Note: The CheckSCT command is equivalent to executing the **ssau -ch** command.

Batch Mode

If the user name and password are included in the command line invocation of Data Vault Administration Tool, a list of administration commands in a text file can be redirected to the program for batch execution. The commands are executed in the order in which they are arranged in the file. This can be useful for unattended, script-based Data Vault administration.

The batch mode syntax is the following:

```
ssaadmin user-name[/password] < batch-file
```

The Data Vault Administration Tool will start, execute all of the batch file commands in sequence, and then exit.

Data Indexing

You can create and manage indexes on tables in the Data Vault to increase query performance. The indexes eliminate data files that do not meet the query WHERE clause filters. The indexes also help to increase the performance of row selection for columns in the WHERE clause filter.

For example, you archived purchase order data by the purchase order ID. You frequently run queries to find purchase orders for a specific date. When you run the queries, the query performance is slow. You create an index for the purchase order date to improve the query performance.

You can create an index for one or more columns in a table. When you create an index, the Data Vault creates an index file for each data file for which the table has a registration. The Data Vault stores the index files in a directory that you define during the installation or you configure in the `ssa.ini` file. When you run queries, the

Data Vault checks if an index exists for the table columns in the query WHERE clause. If an index exists, the Data Vault uses the index to select the data files and to filter the rows.

You can use the standard log or the query statistics log to analyze if the index improves the query performance. The standard log displays the number of data files that the index eliminates as compared to the total amount of data files. For example, the index eliminates 10 data files out of 100 data files. The query statistics log displays the number of data files that the query processes. Use the log files to compare the number of data files that the query processes before and after you create the index.

When you run index commands on a table, you do not affect query access to the table. The Data Vault simultaneously manages index requests and query requests for the same table. Queries only use the updated indexes after the Data Vault completes the index commands.

After you create indexes on tables, you can renew, alter, and drop the indexes. The *ssasql* command line program includes commands to create, renew, alter, and drop indexes.

Rules and Guidelines for Data Indexing

Use standard relational database indexing principles to determine your indexing strategy.

Consider the following rules and guidelines for data indexing:

- Index table columns that you frequently use in queries. The Data Vault uses the indexes to increase the performance of query WHERE clauses. The optimizer does not consider indexes for multiple table joins.
- Consider the partition key of the data files that you want to query. Indexes work best on columns that you did not use as the original partition key.
- Consider the query operation. By default, the system uses a range index when you run queries. Index columns if the queries use operations other than range operations. For example, exact comparison operations such as equal to, not equal to, in, or not in. The indexes that you create supplement the default range index.
- Consider the cardinality of the table column against the total number of rows in the table. Index table columns that have a large amount of unique values such as transaction IDs, customer IDs, and phone numbers. Indexes on table columns with low cardinality, such as gender, age, and country, do not significantly increase the query performance.
- Consider the expected size of the query result set. Index columns if you expect a small result set. If the query result set selects a large amount of rows in a table, then the indexes do not significantly increase the query performance.
- Consider query performance versus database space requirements. When you create an index, the query performance might be drastically reduced. However, the overall database size increases. It might take while to create the index. The time it takes to create an index decreases with the number of Data Vault agents that you run. Balance the needs of query performance with the database space requirements and the time it takes to create the index.

Data Indexing Example

You want to run a query to find a specific customer ID. When you originally archived the customer table, you did not use the customer ID as the partition key. When you run the query, the query performance is slow.

The following table shows a simplified example of data files for a table that includes customer IDs:

Data File	Customer ID	Range of Values
1	1000, 1005, 1010, 1015, 1020, 1025	1000-1025
2	1001, 1006, 1011, 1016, 1021, 1026	1001-1026
3	1002, 1007, 1012, 1017, 1022, 1027	1002-1027

Data File	Customer ID	Range of Values
4	1003, 1008, 1013, 1018, 1023, 1028	1003-1028
5	1004, 1009, 1014, 1019, 1024, 1029	1004-1029

You run a query to search for customer ID 1011. Without the index, the query processes all data files because 1011 is in the value range of all the data files. You create an index on the customer ID column and run the query again. The query performance improves because the query only processes one data file. Data file 2 is the only data file that includes the value 1011.

Create Index

The Create Index command creates an index on an archived table. The Data Vault creates one index file for each data file for which the table has a registration.

You can create an index for a table any time after you archive data to the Data Vault. For example, you archive data to the Data Vault. You identify frequently used queries that have significant performance delays. You create indexes for the tables.

The *ssasql* Create Index command uses the following syntax:

```
create index on <schema name>.<table name> (column name);
```

The following table describes the *ssasql* Create Index arguments:

Argument	Description
schema name	Required. Name of the schema that includes the table columns you want to index.
table name	Required. Name of the table that includes the columns you want to index.
(column name)	Required. Name of the columns on which you want to create an index. You can create the index on one or more columns in the table. The maximum number of columns is the maximum amount of columns in the table. Use a comma to separate multiple columns.

To create indexes on multiple tables, run the command separately for each table. If you create an index for a table that already has an index, the command fails.

After you create an index, you must renew the index if you archive more data to the table. You must drop and create an index if you repartition the table.

Create Index Example

The following syntax shows an example of the *ssasql* Create Index command:

```
create index on loyalty.customer (last_name)
```

When you run the command, the Data Vault creates an index for column *last_name* in table *customer* of schema *loyalty*.

Renew Index

The Renew Index command renews table indexes in the Data Vault. Renew an index when you archive new data to an indexed table.

When you create an index, the index includes information for data files that currently exists in the table. The index is not automatically updated when you add or remove data from the indexed table. If you continue to archive data to an indexed table, you must renew the index to include the newly archived data.

For example, you archive customer data to the Data Vault on a quarterly basis. You created an index for the customer table to increase query performance. When you created the index, the customer table had 100 data files. The next quarter, you archive more data to the customer table. The customer table now includes 500 data files. If you do not renew the indexes for the customer table, the query uses the index only for the original 100 data files. Renew the index to include the new data files.

When you renew an index table, the Data Vault updates all indexes that exist for the table. The Data Vault updates the indexes for new data files that you added for the table. If you removed data files, the Data Vault marks the data files inactive in the index.

The *ssasql* Renew Index command uses the following syntax:

```
renew index on <schema name>.<table name>;
```

The following table describes the *ssasql* Renew Index arguments:

Argument	Description
schema name	Required. Name of the schema that includes the table index to renew.
table name	Required. Name of the table that includes the index to renew.

To renew indexes on multiple tables, run the command separately for each table.

Renew Index Example

The following syntax shows an example of the *ssasql* Renew Index command:

```
renew index on loyalty.customer;
```

When you run the command, the Data Vault renews the index for all indexed columns in table *customer* of schema *loyalty*.

Alter Index

The Alter Index command drops and adds indexes on columns in the same table.

The *ssasql* Alter Index command uses the following syntax:

```
alter index on <schema name>.<table name>
[drop (column name)]
[add (column name)];
```

The following table describes the *ssasql* Alter Index arguments:

Argument	Description
schema name	Required. Name of the schema that includes the table index that you want to alter.
table name	Required. Name of the table that includes the column you want to alter.
(column name)	Required. Name of the column on which you want to drop or add an index. Use a comma to separate multiple columns.

To alter indexes on multiple tables, run the command separately for each table.

Alter Index Example

The following syntax shows an example of the *ssasql* Alter Index command:

```
alter index on loyalty.customer
[drop (last_name)]
[add (phone_number]
```

When you run the command, the Data Vault drops the index for column *last_name* in table *customer* of schema *loyalty*. The Data Vault creates an index for column *phone_number*.

Drop Index

The Drop Index command drops an index on an archived table. When you drop an index, the Data Vault deletes all of the index files for all columns in the table. You must drop and create an index if you repartition an indexed table.

If you archive data to a table that already has an index, you do not have to drop and create the indexes. Use the Renew Index command instead.

The *ssasql* Drop Index command uses the following syntax:

```
drop index on <schema name>.<table name>;
```

The following table describes the *ssasql* Drop Index arguments:

Argument	Description
schema name	Required. Name of the schema that includes the table index you want to drop.
table name	Required. Name of the table from which to drop the index.

To drop indexes on multiple tables, run the command separately for each table.

Drop Index Example

The following syntax shows an example of the *ssasql* Drop Index command:

```
drop index on loyalty.customer;
```

When you run the command, the Data Vault drops all indexes in table *customer* of schema *loyalty*.

Data Vault Performance Tuning

Use the ssatune utility to tune the Data Vault system and update configuration files based on evaluated hardware resources.

The ssatune utility performs the following tasks:

- Analyzes available system resources and calculates Data Vault configuration parameters based on CPU and RAM.
- Analyzes meta query response time and sets the optimum query execution mode for the metadata layer.
- Tunes the metadata server configuration.
- Analyzes the Data Vault temporary storage requirement based on current archive volume and returns a recommendation.
- Refreshes metadata catalog indexes to optimize internal query response time.
- Performs metadata database garbage collection to improve overall metadata layer performance.
- On UNIX systems, operates in high availability mode, which minimizes the Data Vault system down time in the event of an abnormal termination or unexpected shutdown.
- Evaluates I/O response time.
- Updates Data Vault configuration files and performs a backup of the existing configuration.
- Generates a report about current system resources and recommended configuration changes.

The ssatune utility operates in one of the following two modes:

- Evaluation and report only
- Evaluation and apply changes

The ssatune utility uses the following syntax:

```
ssatune [ <flags> ]
[-h (displays the help screen)]
[-m{0,1} (report only vs. update configuration)]
[-t <temporary file directory>]
```

You can specify the flags in any order. The following table describes the flags:

Option	Description
-h	Displays the help screen.
-m	Enter 0 to generate a report about system resources and recommended configuration changes. Enter 1 to update the configuration based on the recommendations. Default is 0.
-t	Location of the temporary directory for log, report, and run-time files. When you run the utility, it generates a log and a report in the temporary directory. The log file contains information about the functions that were successfully called and exited. The report contains information on the parameters that were updated and by how much.

When the ssatune utility is finished, it generates a shell script named `fasd.sh` in the Data Vault root folder. The utility also configures the `ssa.ini` file settings in the STARTER and META sections to enable Data Vault process controls through the generated script. After ssatune generates the script, you can use the script to manage all Data Vault processes.

To start the Data Vault system, run the following command:

```
../fasd.sh start
```

The following message appears: "Starting-> IDV components"

To stop the Data Vault system, run the following command:

```
../fasd.sh stop
```

The following message appears: "Stopping IDV components"

Data Vault Cleanup

Use the ssacleanup utility to remove specified objects from the Data Vault, for example a database, schema, or table. You can also use ssacleanup to remove orphaned SCT files and SCT files assigned to a database that you also remove.

The ssacleanup utility uses the following syntax:

```
ssacleanup [ <flags> ]
-h (displays the help screen)
-a (removes orphan SCT files from disk only)
-e (reports orphan SCT files from disk only)
-b <n> (time in hours to remove orphan files older than <n> hour. Default is 0.5)
-c (removes SCT files assigned for database for -d flag)
-d <name> (removes database)
-l<dir>[,<dir>] (SCT files path(s) for -a mode or -e mode only. Multiple paths are
separated by commas)
-o (deletes offline SCT files for a database, schema, or table from disk and metadata)
-o -e (reports offline SCT files for database/schema/table)
-r <dir> (temporary directory for log, report, and runtime files
-s <name> (removes schema)
-t <name> (removes table)
-u UID[/PWD] (Data Vault user name and password)
```

You can specify the flags in any order. The following table describes the flags:

Option	Description
-h	Displays the help screen.
-e	Generates a report of the orphan SCT files. The report is generated in the current working directory from where ssacleanup is called. If you want to generate and review a report of orphaned SCT files before you remove them, use the `-e` flag.
-a	Removes orphaned files from disk only. You must specify the location of the files with the `-l` flag.
-b	Specifies the time in hours for which the command removes orphaned files older than the number of hours (`<n>`). By default, this value is .5 hours and all orphaned files created more than 30 minutes after running the utility will be deleted. Files that have been created within 30 minutes of running the utility are not deleted by default. This protects SCT files that were recently-created by a running archive loader job, but are not yet registered in the metadata catalog. You can change the default value to retain SCT files that were created within a certain time period.
-c	Removes SCT files assigned to a database object that you will remove by specifying a value for the `-d` flag. By default, all SCT files registered for removed database objects remain on disk. Once cleanup is finished, the files become orphan. Use the `-c` flag to remove the SCT files when you remove the object. Or, you can use the `-a` flag to remove orphaned files at a later time.

Option	Description
-d	Removes the specified database object. This parameter is mandatory for cleaning up Data Vault objects.
-l	Specifies the location of orphaned SCT files that you want to remove. Files that are present in this location but not referenced in the metadata catalog are deleted when you specify the `-a` flag. Both the `-a` flag and `-l` flag are mandatory to remove orphaned files. You can specify multiple locations. Separate the file paths with a comma.
-o	Unregisters and deletes the offline SCT files that belong to the particular database, schema, or table. -o deletes only the SCT files from the local disk and does not delete SCT files from any external storage. It unregisters the SCT files regardless of the file location. For example: `ssacleanup -o -d <DBNAME> -u dba/` will unregister and delete all of the offline SCT files stored on the local disk for the specified database.
-o -e	Generates a report with a list of the offline SCT files that belong to the particular database, schema, or table. For example: `ssacleanup -o -e -d <DBNAME> -u dba/` will report all of the offline SCT files for the specified database.
-r	Location of the temporary directory for log, report, and run-time files. When you run the utility, it generates a log and a report in the temporary directory. The log file contains information about the functions that were successfully called and exited. Create this directory before you run the command. If you encounter an error, check the log files.
-s	Removes the specified schema.
-t	Removes the specified table. When you specify a value for `-t<name>`, you must also specify a value for the flags `-d<name>` and `-s<name>`.
-u	Required. Data Vault user name and password.

Note: The Data Vault DBA user or owner of the database with granted privileges must run the ssacleanup utility. If a non-DBA user calls ssacleanup to remove objects that they do not own, the process fails with an authorization error.

Column Definitions

Use the ssadesc utility to list the column definitions for a specified table, schema, or database. By default, the utility prints the definitions in the screen output. Optionally, ssadesc can generate a Microsoft Excel file with the column definitions listed.

The following columns are included in the definition list:

- TABLE_CAT
- TABLE_SCHEM
- TABLE_NAME
- COLUMN_NAME
- TYPE_NAME

- COLUMN_SIZE
- DECIMAL_DIGITS
- POSITION
- IS_NULLABLE

The ssadesc utility uses the following syntax:

```
ssadesc [ <flags> ]
[-h (displays the help screen)]
[-e (creates Excel report file)]
[-d<name> (database name)]
[-r<dir> (temporary directory for log, report, and runtime files)]
[-s<name> (schema name)]
[-t<name> (table name)]
[-u UID[/PWD] (Data Vault user name and password)]
```

You can specify the flags in any order.

Materialized Views

You can create, refresh, and drop materialized views in the Data Vault. Materialized views improve the performance of queries that have multiple joins or aggregations.

A materialized view is a database object in the Data Vault that contains the results of an SQL query. When you create a materialized view, the Data Vault creates and registers the materialized view as a table. The Data Vault also registers the tables referenced in the SQL query of the materialized view. After you create the materialized view, you can include the materialized view in a SELECT statement.

You cannot apply legal hold, retention, data browse, or data purge functionality to materialized views. Compliance features are supported only on the retired tables themselves, and not the overlying materialized views. When you purge data from a table, you must manually refresh the materialized views in addition to the indexes to sync the materialized view with the table.

You cannot delete the tables that are referenced in a materialized view. If you update the tables that are referenced in the materialized view, you must refresh the materialized view to access the updated data.

When you create a materialized view, you cannot have a trailing space at the end of the alias column names in the query used to create the materialized view.

When you delete a materialized view, the Data Vault unregisters the data files that are associated with the materialized view. The Data Vault also deletes the materialized view table from the system catalog.

Materialized views in the Data Vault differ from the virtual views supported by Data Archive.

For information about the materialized views used by the Application Retirement for Healthcare Accelerator, see the *Informatica Data Archive Application Retirement for Healthcare Accelerator Reference*.

Location Parameter

To create a materialized view, use the argument MVIEWDIR as the location where the materialized view is created.

If you do not specify the MVIEWDIR location as shown in the examples below, Data Vault creates the materialized view in the SHAREDIR location as provided in the `ssa.ini` file. Data Vault housekeeping activities can clear the SHAREDIR location at any time. If Data Vault clears the SHAREDIR location, you can lose any materialized views you created previously. As a best practice, use the MVIEWDIR location in either the `ssa.ini` file or in the SQL command itself.

Specify the MVIEWDIR location in one of the following ways:

1. As an parameter in the `ssa.ini` file.
   Add the MVIEWDIR parameter in the [QUERY] and [SERVER] sections of the `ssa.ini` file. For example:

   [QUERY]

   ……

   MVIEWDIR=/data/mviewdir

   [SERVER]

   …….

   MVIEWDIR=/data/mviewdir

   After you add the entries to the `ssa.ini` file, restart Data Vault.
2. Alternatively, you can specify MVIEWDIR as part of the SQL to create the materialized view itself. With this method you do not need to restart Data Vault. For example:

   ```
   create materialized view <view name> as SELECT <Col1 or expression 1>, <Col2 or
   expression 2> …. < Coln or expression n> FROM <table name> WHERE <condition>
   {GROUP BY} {ORDER BY} {Limit}
   MVIEWDIR '/data/mviewdir'
   ```

   In the example above, `/data/mviewdir` is just an example. You can provide any location other than the SHAREDIR or TEMP folders.

Commands

- CREATE MATERIALIZED VIEW <view-name> AS <SELECT statement> MVIEWDIR <Directory Name>
- REFRESH MATERIALIZED VIEW <view-name>
- DROP MATERIALIZED VIEW <view-name>;

CHAPTER 7

Data Repartitioning

This chapter includes the following topics:

- Data Repartitioning Overview, 105
- Data Repartitioning Use Cases, 105
- Data Repartitioning Stages, 106
- Data File Calculation, 106
- Data File Creation, 107
- Data File Registration, 109
- Data Repartitioning Log Files , 110
- Data Repartitioning Example, 110
- ssapart, 111

Data Repartitioning Overview

You can repartition archived data to improve query performance and to free resources. You can also repartition data to change the number of rows in data files. You partition data directly in the Data Vault. You do not need to restore data back to the source.

You can repartition data any time after you archive the data. When you repartition data, you use one table column as the partition key. The partition key column can include any datatype except for LOB datatypes. The Data Vault uses the range partitioning method to repartition data.

Users can still query the tables that you repartition. The Data Vault simultaneously manages repartition requests and query requests for the same table. The repartition process occurs in one database commit. Queries only use the repartitioned data files after entire repartition process completes.

When you repartition data, Data Vault performance is not affected. You can use multiple threads to increase the data repartitioning performance.

Data Repartitioning Use Cases

Repartition archived data to improve query performance and to free resources. You can also repartition data to change the number of rows in data files.

Consider data repartitioning for the following use cases:

To improve query performance

Consider data repartitioning if query performance is poor. Query performance might be poor if the query filter does not match the partition key that you used to archive the data. The optimizer cannot effectively limit the data file selection if the query filter exists in a majority of the data files. In addition, the query might consume more resources due to the number of data files that the query processes. Repartition data to reduce the number of data files involved in query processing. When you reduce the number of data files in query processing, query performance improves.

You might want to repartition data if reporting requirements changed after you archived the data. For example, you used a timestamp characteristic as the original partition key when you archived the data. Queries that filter by the timestamp characteristic perform well. One year after you archived the data, you want to use the customer ID to filter data. Queries that filter by the customer ID perform poorly. Repartition the data based on the customer ID to improve the query performance.

To change the number of records in data files

Consider data repartitioning to change the number of records in data files. When you repartition data, you can increase or decrease the number of rows. You might want to increase the number of rows to reduce the number of data files on disk. You might want to decrease the number of rows to improve the performance of complex joins.

Data Repartitioning Stages

When you repartition data, the Data Vault calculates the total number of data files, creates the data files, and registers the data files to Data Vault repository.

Data repartitioning includes the following high-level stages:

Data file calculation

You specify the number of rows that each repartitioned data file includes. The Data Vault uses the row count to calculate the total number of repartitioned data files and the minimum and maximum value range for each repartitioned data file.

Data file creation

The Data Vault creates temporary files to copy data from the original data files to the corresponding value range of the repartitioned data files. Then the Data Vault merges the temporary files of the same value ranges and creates the repartitioned data files.

Data file registration

The Data Vault registers the repartitioned data files in the Data Vault repository. The Data Vault can register the repartitioned data files for the original archived table or for a new table in the Data Vault repository.

Data File Calculation

The Data Vault calculates the total number of repartitioned data files for a table based on the row count. Then, the Data Vault determines the minimum and maximum values for each repartitioned data file. You specify the row count when you repartition data.

You can configure the following row count options when you repartition data files:

Keep the same row count

Configure the same row count as the original data files to create the same number of repartitioned data files. You can view the row count for each data file in the file size report.

Decrease the row count

Configure a smaller row count to increase the number of repartitioned data files. You might want to increase the number of repartitioned data files to help improve the performance of complex queries.

For example, the system has memory issues when you run a query on data files that include a large number of rows. The query includes a complex join statement on a table that has 10 million rows in each data file. The query is slow because there are too many rows to join for each data file. You repartition the table to include 5 million rows in each repartitioned data file.

Increase the row count

Configure a higher row count to decrease the number of data files. You might want to decrease the number of data files to have fewer files on disk. For example, you want fewer files on disk to simplify your backup strategy. You have a table that users do not run queries on. The table includes a high number of data files. You repartition the table to include 75% less data files.

After the Data Vault determines the total number of repartitioned data files, the Data Vault calculates the minimum and maximum value range for each repartitioned data file. The Data Vault uses the minimum and maximum value ranges to copy data from the original data files to the repartitioned data files.

Data File Creation

The Data Vault determines the total number of repartitioned data files and the minimum and maximum value range for each repartitioned data file. The Data Vault creates temporary files to copy data from the original data files to the corresponding value range of the repartitioned data files. Then, the Data Vault merges the temporary files of the same value ranges and creates the final repartitioned data files.

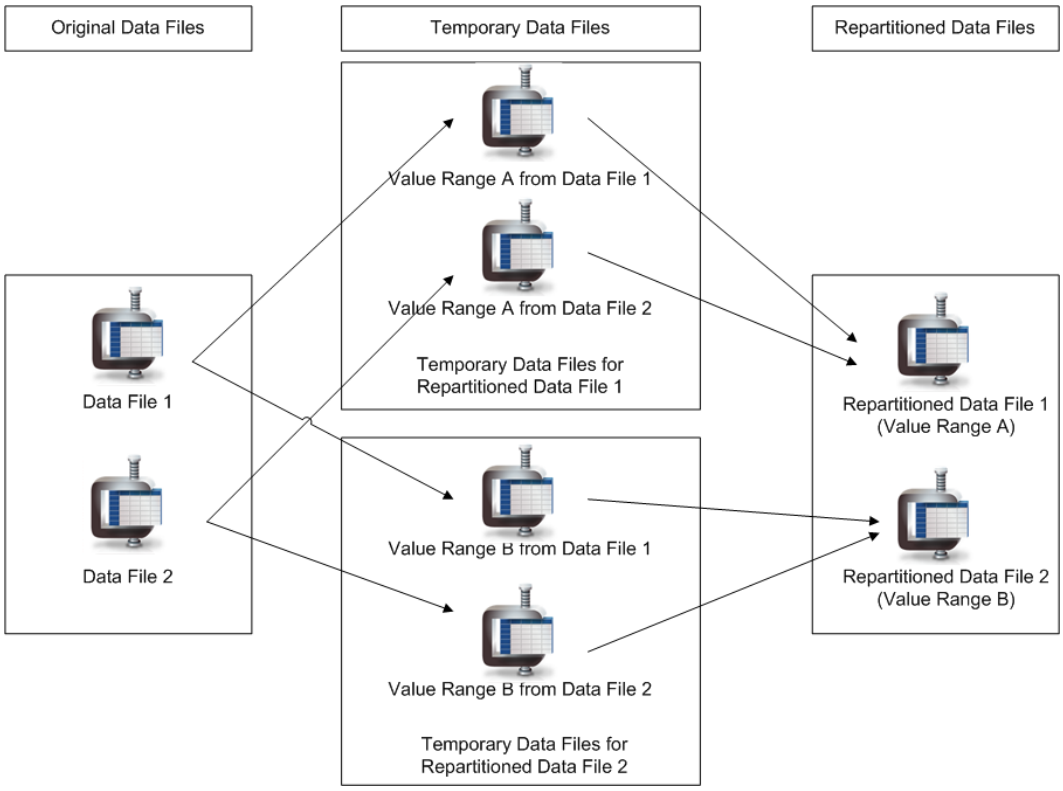
The Data Vault creates temporary files to split the values of the original data files to match the value range of the repartitioned data files. The Data Vault copies data from the original data file to the temporary file that corresponds to the minimum and maximum value range of the repartitioned data file. The number of temporary files the Data Vault creates for each original data file is the same as the total number of repartitioned data files. For example, if the final number of repartitioned data files is two, then the Data Vault creates two temporary files for each original data file. You specify the location of the temporary file storage when you repartition the data.

After the Data Vault splits the values of each data file to the value range of the repartitioned data file, the Data Vault creates the repartitioned data files. To create the repartitioned data files, the Data Vault merges all of the temporary files of the same value range. The Data Vault stores the repartitioned data files in the data file directory configured in the ssa.ini file. Then, the Data Vault registers the repartitioned data files to the Data Vault repository.

Data File Creation Example

You want to repartition a table that includes two data files. You configure the same row count as the original data files. The final number of repartitioned data files for the table is two.

The following image shows a simplified example of how the Data Vault creates repartitioned data files when the row counts are the same:



To repartition the data, the Data Vault completes the following high-level steps:

1. The Data Vault calculates the minimum and maximum value range for each repartitioned data file.

   Repartitioned data file 1 includes value range A. Repartitioned data file 2 includes value range B.

2. The Data Vault creates four temporary files to split data from the original data files into the value range of the repartitioned files.

   The first temporary file stores data from data file 1 that falls within value range A for repartitioned data file 1. The second temporary file stores data from data file 2 that falls within value range A for repartitioned data file 1. The third temporary file stores data from data file 1 that falls within the value range B for repartitioned data file 2. The fourth temporary file stores data from data file 2 that falls within the value range B for repartitioned data file 2.

3. The Data Vault merges the two temporary files that contain data in value range A to create repartition data file 1. The Data Vault merges the other two temporary files that contain data in value range B to create repartition data file 2.

Data File Registration

After the Data Vault creates the repartitioned data files, the Data Vault registers the repartitioned files in the Data Vault repository. The Data Vault can register the repartitioned data files for the original archived table or for a new table in the Data Vault repository.

Data File Registration for Archived Tables

When you repartition data, the Data Vault can register the repartitioned data files for the original archived table. The Data Vault registers the repartitioned data files in the Data Vault repository for the original archived table and unregisters the original data files. The original data files become orphaned files.

You can keep or delete the orphaned files. You might want to keep the orphaned files to revert back to the original data files. For example, if you erroneously repartition the wrong table or you use an incorrect partition key. You might want to delete the orphan files after you verify that the repartitioning was successful to ensure that the Data Vault repository is consistent.

Data File Registration for New Tables

When you repartition data, the Data Vault can register the repartitioned data files for a new table.

When you repartition data files, you can create a table for the repartitioned data files. Create a table if you do not want to register the repartitioned data files to the original archived table. You specify the table name when you repartition the data files. The Data Vault uses the structure of the original archived table to create the table.

The original archived table and data files remain. The Data Vault repository still contains the registration of the original data files to the original archived table. You can configure queries to read data from the original archived table or the new table. If you want existing queries to read data from the new table, you must update the table reference in the query.

Use Cases for New Table Registration

When you repartition data files, you can create a table for the repartitioned data files.

Consider the following use cases in which you might want to register repartitioned data files to a new table:

- To evaluate the performance benefits of the data repartitioning. For example, you can repartition the data and register the repartitioned data files to a new table. You can run a query on the original archived table and then run the same query on the new table. Use the query statistics logs to compare the statistics of both queries. If the repartitioned data files increase query performance, you can drop the original archived table. Or, you can repartition the data files and register the data files to the original archived table.
- To perform temporary analysis of the archived data. For example, you temporarily run a series of queries on the archived data. Users do not regularly run the queries. You repartition the data to improve query access and then drop the table and repartitioned data files when you finish the analysis.
- To preserve the immutability of the original archived data. For example, you do not want to change the original archived data.

Data Repartitioning Log Files

When you repartition data, the Data Vault creates a log file. The log file summarizes all steps that the Data Vault performs and provides any relevant error or warning messages. Analyze the log file to troubleshoot data repartitioning.

The Data Vault creates the log file in the temporary location that you specify when you repartition data.

You can find the log file in the following location:

```
<temporary storage directory>/sctpartition_<timestamp>.log
```

Data Repartitioning Example

You archive purchase orders on a quarterly basis. For example, you archive purchase orders with transaction dates between September, 1 2010, and December 31, 2010. You regularly run queries to find purchase orders on a specific date, such as December 24, 2010. When you run queries, the query performance is slow. The query statistics log shows that the query processes a majority of the data files. In addition, the query consumes a large amount of resources.

Before Data Repartitioning

When you originally archived the data, the data files were grouped by the purchase order ID, not by the purchase order date.

The following table shows a sample of the data files before you repartition the data files:

Data File	Purchase Order Date	Range of Values
1	9/01/10, 11/24/10, 12/24/10, 10/15/10	9/01/10 - 12/24/10
2	12/31/10, 9/02/10, 10/31/10, 11/02/10	9/02/10 - 12/31/10
3	11/28/10, 10/05/10, 12/24/10, 9/15/10	9/15/10 - 12/24/10
4	9/30/10, 11/24/10, 10/01/10, 12/28/10	9/30/10 - 12/28/10

When you run a query to select purchase orders for 12/24/10, the query processes all data files because 12/24/10 is found in the range of all data files. The query performance is slow because the query selects a large number of data files.

After Data Repartitioning

You repartition the data files by the purchase order date. After you repartition the data files, the data files are grouped by the purchase order date.

The following table shows a sample of the data files after you repartition the data files:

Data File	Purchase Order Date	Range of Values
1	9/01/10, 9/02/10, 9/15/10, 9/30/10	9/01/10 - 9/30/10
2	10/15/10, 10/31/10, 10/05/10, 10/01/10	10/01/10 - 10/31/10
3	11/24/10, 11/02/10, 11/28/10, 11/24/10	11/02/10 - 11/28/10
4	12/24/10, 12/31/10, 12/24/10, 12/28/10	12/24/10 - 12/31/10

When you run a query to select the 12/24/10 date, the query only selects data file 4. The query performance increases because the query selects a smaller amount of data files.

ssapart

Repartitions data files in the Data Vault.

The ssapart command uses the following syntax:

```
ssapart
-t <schema_name.table_name>
-c <column_name>
[-n <number_of_rows>]
[-m <number_of_parallel_processes>]
[-r <temporary folder location>]
[-a <schema_name.new_table_name>]
[-d (treat integer as date)]
[-h (displays help screen)]
<connection> <user>/<password>
```

You can specify the flags in any order.

Note: If you repartition data files in an indexed table, you must renew the table index.

The following table describes the ssapart command arguments:

Option	Argument	Description
-t	schema_name.table_name	Required. Name of the schema and table that you want to repartition.
-c	column_name	Required. Name of the column that you want to use as the partition key to repartition the data files. You can use one column as the partition key. If you enter multiple columns, the command fails.
-n	number_of_rows	Optional. Number of rows in each repartitioned data file. Configure the same row count as the original data files to create repartitioned data files with the same number of rows. You can use the file report to view the row count of the original data files. Configure a smaller row count to increase the number of repartitioned data files. Configure a larger row count to decrease the number of repartitioned data files. Default is 10,000.
-m	number_of_parallel_processes	Optional. Number of threads that run in parallel to repartition the data files. Increase the number of threads to increase the data repartitioning performance. Default is 1.
-r	temporary_folder_location	Optional. Directory where the Data Vault creates temporary files and log files. The directory must include space up to twice the size of the archived table. Use the table report to view the archived table size. The Data Vault deletes the temporary files when the Data Vault completes the data repartitioning. The log files remain in the directory. Default is the current working directory.
-a	schema_name.new_table_name	Required to create a table in which to register the repartitioned data files. Name of the table that the Data Vault creates. If you do not use the -a option, the Data Vault registers the repartitioned data files to the original archived table.

Option	Argument	Description
-d	-	Required if the partition key column you specify for the -c option stores dates in an integer datatype. Processes the partition key column value as a date datatype. The Data Vault only converts the value from numeric to date when the Data Vault creates the temporary files. The Data Vault does not convert the value to the date datatype in the database. For example, the partition key column includes value 11302006. If you specify the -c option, the Data Vault temporarily converts the value to 11/30/2006. Default is no conversion. The Data Vault uses the datatype of the partition key column.
-h	-	Optional. Displays onscreen help. The help shows the syntax of the command and the possible flags.
-	connection	Required. Data Vault connection name.
-	database	Required. Name of the database in the Data Vault. The database corresponds to the Data Archive archive folder.
-	user/password	Required. Name and password of the user account that connects to the Data Vault.

ssapart Example

The following syntax shows an example of the ssapart command:

```
ssapart -t loyalty.ticket_coupon -c ID_NUM -n 2500 -m 4 -r c:\temp fas loyalty dba/dba
```

When you run the command, the Data Vault uses the connection *fas* to connect to database *loyalty*. The Data Vault uses column *ID_NUM* in table *ticket_coupon* in schema *loyalty* to repartition the data. The Data Vault uses four parallel threads to repartition the data. Each data file includes 2,500 rows of data. The Data Vault stores the temporary files and log files in directory *c:temp*.

CHAPTER 8

Partial Data Vault Copy

This chapter includes the following topics:

- Partial Data Vault Copy Overview, 113
- Data Export, 114
- Data Export Files, 114
- Automatic File Transfer, 116
- Manual File Transfer, 117
- ssadbcopy for Data Export, 117
- Data Import, 119
- ssadbcopy for Data Import, 120
- Prerequisites for Partial Data Vault Copy, 121
- Known Limitations, 122
- Steps to Partially Copy Data Between Data Vault Systems, 122
- Partial Data Vault Copy Example, 123

Partial Data Vault Copy Overview

Perform a partial Data Vault copy to copy data between Data Vault systems. You can copy data from one database, one schema, or one table at a time.

Consider a partial Data Vault copy between Data Vault systems if you retire multiple applications over time. For example, you plan to retire one application every month for the next two years. You archive data from one application to a Data Vault test system. After you archive and validate the data in the Data Vault test system, you want to copy the data to a Data Vault production system.

When you copy data between Data Vault systems, the data files retain the original compression. The data remains immutable. The data copy is directly from the source to the target. You do not need an intermediate staging area.

With ssadbcopy, you can also perform a partial Data Vault copy to copy data from a Data Vault instance with a local file system to a Data Vault instance that stores database files in an external storage system, for example Centera.

When you perform a partial Data Vault copy, you run the ssadbcopy command line utility in the source and target systems. You run the utility in the following modes:

Data export

When you export data from the source Data Vault system, you run the utility in data export mode. You specify the data that you want to copy. The utility creates data export files that contain DDL and administrator scripts. You transfer the data export files to the target system manually or automatically with an FTP server.

Data import

When you import data in the target system, you run the utility in data import mode. The utility runs the DDL and administrator scripts from the data export files. You specify how to insert the data. You can append the data, or you can drop and create the data.

Data Export

When you export data from the source Data Vault system, you specify the data that you want to copy. You run the ssadbcopy command line utility to export the data. The utility creates data export files that contain DDL and administrator scripts. You transfer the data export files to the target Data Vault system manually or automatically with an FTP server.

When you run the ssadbcopy command for data export, the utility performs the following high-level steps:

1. Connects to the Data Vault server on the source system.
2. If you use automatic file transfer, the utility connects to the FTP server on the target system.
3. The utility extracts the metadata structure from the Data Vault repository for the database, schema, or table that you export.
4. The utility generates data export files.
   The files include DDL and Data Vault administrator scripts that re-create the identical structure on the target. The utility saves the files in a temporary file location on the source system.
5. If you use automatic file transfer, the utility copies the TAR archive file and the data files to the target system.

Data Export Files

When you run the ssadbcopy command for data export, the utility generates a TAR archive file, a data file list, and a log file on the source system. The utility stores the files in a temporary file location that you specify in the command. The TAR archive file and the data file list are required to import the data in the target system.

You can use an FTP server to copy the files to the target system, or you can manually copy the files to the target system. You specify which mode you want to use when you run the utility.

TAR Archive File

The TAR archive includes a set of SQL, DDL, and administrator scripts. The scripts contain information on the data and structure export. You use the scripts to import the data in the target system. The utility generates the scripts based on the command arguments that you specify.

The TAR archive file includes the following files:

idvmeta_sctlist_target.txt

Text file that includes a list of all required data files for the target system. When you import the data in the target, the ssadbcopy utility validates that all of the data files in the text file exist on the target.

idvmeta.sql

SQL script that creates the database on the target system if the database does not already exist.

idvmeta.ddl

DDL script that contains the DDL structure of the data that you want to extract. When you import the data, the script replicates the source structure on the target. The script creates the schemas, domains, and tables on the target system.

idvmeta_regtable.adm

Data Vault administrator script that registers tables in the Data Vault repository on the target system.

idvmeta_regsct.adm

Data Vault administrator script that registers data files in the Data Vault repository on the target system. To register the data files, the script uses the target data file location that you provide in the ssadbcopy command.

idvmeta_unregtable.adm

Data Vault administrator script that unregisters tables on the target system. The utility uses the script if you import data in drop and create mode.

idvmeta_cleanup.sql

DDL script that drops the schema, domains, and table structure on the target system. The utility uses the script if you import data in drop and create mode.

You can find the TAR archive file in the following location:

```
<temporary file directory>/idvmeta_backup_transfer_<timestamp>.tar
```

Data File Text File

The data file text file includes a list of all of data files that include the data that you want to copy. The data files need to exist in the target Data Vault system before you begin the data import phase.

If you transfer the data export files manually, use the list to identify the data files that you need to copy. You need to copy the data files from the source Data Vault system to the target system. When you run the ssadbcopy utility, you specify the location of the data files on the target system with the `-n` and `-l` options in the utility.

If you transfer the data export files automatically, you can use the data file list as an audit log. For example, you can verify that all of the required data files exist in the target system before you import the data.

You can find the data file list in the following location:

```
<temporary file directory>/idvmeta_sctlist.txt
```

Data Export Log File

The data export log file includes a summary of all the steps that the ssadbcopy utility completes for the data export. Use the log file for troubleshooting.

You can find the data export log file in the following location:

```
<temporary file directory>/ssadbcopy_<timestamp>.log
```

Automatic File Transfer

You can use an FTP server to automatically transfer the data export files that the ssadbcopy utility generates. You can use automatic file transfer mode if you transfer the files to target Data Vault systems on UNIX. To transfer files to a target Data Vault system on Microsoft Windows, use the manual file transfer mode.

If you specify the automatic file transfer mode in the ssadbcopy command, the utility transfers the following files from the source system to the target system:

Data Files

The utility transfers all of the data files identified in the data file text file to the compressed data file directory on the target system. You specify the target location when you run the ssadbcopy command.

TAR archive file

The utility transfers all of the files in the TAR achive to the target TAR directory on the target system. You specify the target directory in the FTP.ini file.

FTP.ini File Parameters

Configure the FTP.ini file on the source Data Vault system.

Configure the FTP.ini file to include the following required Data Vault parameters:

host

IP address of the target system.

user

Data Vault user that transfers the files.

password

Password for the Data Vault user.

mode

Mode for the file transfer. Use binary mode only.

tar_location

Location on the target system in which the ssadbcopy utility transfers the TAR archive file. Do not confuse with the data file location. You provide the data file target location in the ssadbcopy command.

FTP.ini File Example

The following example shows the FTP.ini file parameters for automatic transfer mode:

```
host=10.17.40.25
user=ilmadm
password=password
mode=binary
tar_location=/u01/ilmadm/dbcopy_archive
```

Manual File Transfer

You can manually transfer the data export files that the ssadbcopy utility generates. You must transfer the data export files manually if you want to copy data to a target Data Vault system on Microsoft Windows.

If you specify the manual file transfer mode in the ssadbcopy command, transfer the following files from the source Data Vault system to the target Data Vault system:

- Data files. Transfer all of the data files listed in the data file text file. To copy the data to an external storage system, transfer the data files to a temporary directory specified in the ssadbcopy options.
- TAR archive file.

You must transfer the files to the target system before you can import the data.

ssadbcopy for Data Export

Generates export files for a partial Data Vault copy. Run the ssadbcopy command on the source Data Vault system.

The ssadbcopy command uses the following syntax:

```
ssadbcopy
-d <database_name>
[-s <schema_name>]
[-t <table_name>]
-n <target_sct_file_directory>
[-r <source_temporary_file_directory>]
-u <Data_Vault_administrator_user>[/password]
[-f (enables automatic FTP file transfer)]
[-h (displays help screen)]
[-l <temporary directory for data files on the target external storage system>]
[-o (preserves the original data files locations)]
-b (admin task timeout in milliseconds)
```

The following table describes the ssadbcopy command arguments:

Option	Argument	Description
-d	database_name	Required. Database name from which you want to copy data. The database name is the same as the archive folder name in the Data Archive Data Vault target connection. You can copy data from one database at a time.
-s	schema_name	Required. Schema from which you want to copy data. You can copy data from one schema at a time. Use the full schema name. If you enter multiple schemas or use wildcards, the command fails. If you do not specify a schema, the utility copies data from all schemas in the database that you specify in the -d option.
-t	table_name	Optional. Table from which you want to copy data. You can copy data from one table at a time. If you enter multiple tables or use wildcards, the command fails. If you do not specify a table, the utility copies data from all tables from the schema that you specify in the -s option.

Option	Argument	Description
-n	target_data_file_directory	Required. Full path of the location in the target Data Vault system that stores the data files.
-r	source_temporary_file_directory	Location of the temporary directory on the source Data Vault system. The utility creates the ssadbcopy_temp folder in the specified directory and then creates the runtime files, log file, and export files inside of it. If you run two parallel ssadbcopy export jobs, do not use the same directory for -r. Default is the current working directory.
-l	-	Required when the -n directory is an external storage directory. Not required for a local file system. Full path of the location in the target external storage system that stores the data files that you want to import to the external storage system. Provide the -l option when you have manually transferred the data files to the target. When you run the utility, the data files are transferred to the `-n` directory.
-o	-	Optional. Keeps the original data file location. Use when data migration occurs between multiple Data Vault instances using a shared data file location within the same server.
-u	Data Vault_administrator_user>[/password]	Required. Name and password of the Data Vault administrator user account.
-f	-	Optional. Enables automatic FTP file transfer. If you specify the -f option, the utility uses the FTP server to transfer the export files to the target system. If you do not specify the -f option, you must manually transfer the export files to the target system. Default is manual transfer.
-h	-	Optional. Displays onscreen help. The help shows the syntax of the command and the possible flags.
-b		Task timeout, in milliseconds. This parameter is used for registration of SCT files during data import to the target. Default is 600000 ms.

ssadbcopy for Data Export Examples

You can run the ssadbopy command in data export mode with one or more options.

Copy data from one database and export files automatically with FTP server

You want copy data from one database and use an FTP server to automatically export files from the source Data Vault system to the target Data Vault system.

Run the ssadbcopy command with the following options:

```
ssadbcopy -f -n /u01/sct_stage -r /data/temp -d LOYALTY
```

The utility exports the data and structure for database name LOYALTY. The utility saves the data export files to the `/data/temp` directory on the source system. The utility uses an FTP server to transfer the data files and the data export files to the Data Vault target system. The utility transfers the data files to the `/u01/sct_stage` directory on the Data Vault target system. The utility transfers the data export files to the location that you configured in the FTP ini.file.

Copy data from one table and export the files manually

You want to copy data from one table and manually export files from the source Data Vault system to the target Data Vault system.

Run the ssadbcopy command with the following options:

```
ssadbcopy -n /u01/sct_stage -r /data/temp -d LOYALTY -s LOYALTY_AIR -t TICKET
```

The utility exports the data and structure for database LOYALTY, schema LOYALTY_AIR, table TICKET. The utility saves the data export files to the `/data/temp` directory on the source system. You transfer the SCT data files to the `/u01/sct_stage` directory on the Data Vault target system. You also transfer the data export files to a temporary directory on the Data Vault target system.

Copy data from one table and export the data files manually to a Data Vault instance with an external storage system

You want to copy data from one table and manually export the data files from the source Data Vault instance with a local file system to a target Data Vault instance with an external storage system.

Run the ssadbcopy command with the following options:

```
ssadbcopy -n cas://2_ssadbcopy_test/ -t LINEITEM -s TPC_H -d SDS -r /u01/
IDV642_Lin64/temp -l /u01/IDV642_Lin64/temp -u dba/dba
```

The utility exports the data and structure for database SDS, schema TPC_H, table LINEITEM. The utility saves the data export files to the `/u01/IDV642_Lin64/temp` directory on the source system. You must manually transfer the SCT data files to the `/u01/IDV642_Lin64/temp` directory on the external storage system. You also transfer the data export files to a temporary directory in the external storage file system.

Data Import

When you import the data in the target system, the ssadbcopy utility runs the DDL and administrator scripts on the target Data Vault system. You can append the data, or you can drop and create the data.

When you run the ssadbcopy command for data import, the utility performs the following high-level steps:

1. The utility extracts the files from the TAR archive file to a temporary directory.
   You provide the full path of the TAR archive file and specify the location of the temporary directory in the command.
2. The utility validates that all of the data files exist in the target system.
   The utility uses the data file list from the TAR archive to verify that all of the data files exist in the target data file directory.
3. The utility runs the SQL, DDL, and administrator scripts from the TAR archive file.
   The utility creates the data structures. The utility registers the tables and data files in the Data Vault repository. For example, if you insert data in drop and create mode, the utility drops the data structure and unregisters the tables from the Data Vault repository.
4. The utility validates the metadata of all new objects to verify that the partial copy is complete.

Data Import Log File

The data import log file includes a summary of all the steps that the ssadbcopy utility completes for the data import. The log file also contains warning and error messages. Use the log file for troubleshooting.

You can find the data export log file in the following location:

```
<temporary file directory>/ssadbcopy_<timestamp>.log
```

You specify the temporary file directory as an option when you run the ssadbcopy command.

ssadbcopy for Data Import

Imports data to a target system for a partial Data Vault copy. Run the ssadbcopy command on the target Data Vault system.

The ssadbcopy command uses the following syntax:

```
ssadbcopy
-i <TAR_archive_file_location>
[-a <insert_mode (0,1)>]
[-r <temporary_file_directory>]
[-h (displays help screen)
-u <Data_Vault_administrator_user>[/password]
-c (cleans up the SCT files from the staging location)
```

The following table describes the ssadbcopy command arguments:

Option	Argument	Description
-i	TAR_archive_file_location	Required. Full path of the directory on the target Data Vault system that contains the TAR archive file, including the TAR archive file name.
-a	insert_mode	Optional. Determines how the utility inserts data. Specify one of the following options: - 0. Appends data to existing tables. - 1. Drops and creates the data and structures. Use this mode if the table structure or data model on the source system is different than the structure on the target system. Default is 0.
-r	temporary_file_directory	Location of the temporary directory on the target Data Vault system. The utility creates the ssadbcopy_temp folder in the specified directory and then creates the runtime files, log file, and export files inside of it. If you run two parallel ssadbcopy export jobs, do not use the same directory for -r. Default is the current working directory.
-h	-	Optional. Displays onscreen help. The help shows the syntax of the command and the possible flags.
-u	Data Vault_administrator_user>[/password]	Required. Name and password of the Data Vault administrator user account.
-c		Optional. Removes the SCT files from the -l directory after successful migration.

ssadbcopy for Data Import Examples

You can run the ssadbopy command in data import mode with one or more options.

Data import in append mode

You want to insert data in the target system in append mode.

Run the ssadbcopy command with the following options:

```
ssadbcopy -i /u01/ilmadm/tar_cp/idvmeta_backup_transfer_2014_02_26_11_45_39.tar -
r /u01/ilmadm/temp
```

The utility opens the TAR archive file in directory `/u01/ilmadm/tar_cp/` and extracts the contents to directory `/u01/ilmadm/temp`. The utility appends the data to the existing data structure.

Data import in insert mode

You want to insert data in the target system in insert mode.

Run the ssadbcopy command with the following options:

```
ssadbcopy -a 1 -i /u01/ilmadm/tar_cp/idvmeta_backup_transfer_2014_02_26_11_45_39.tar
-r /u01/ilmadm/temp
```

The utility opens the TAR archive file in directory `/u01/ilmadm/tar_cp/` and extracts the contents to directory `/u01/ilmadm/temp`. The utility drops and creates the data structures.

Prerequisites for Partial Data Vault Copy

Before you begin, verify the prerequisites. Verify the prerequisites for the source and target system. To use an FTP server to transfer the files from the source system to the target system, verify the automatic file transfer prerequisites.

Prerequisites for Source and Target Data Vault Systems

Verify the prerequisites for the source and target Data Vault systems.

Verify the following prerequisites:

- The Data Vault is installed on both the source and target systems. The Data Vault version on the target system is the same or newer than the Data Vault version on the source system.
- The Data Vault Service is started on the source and target systems.
- The ssadbcopy command line utility is installed in the Data Vault root installation directory on the source and the target systems.

Prerequisites for Automatic File Transfer

Verify the following prerequisites to use an FTP server to automatically transfer files from the source Data Vault system to the target Data Vault system.

Verify the following prerequisites:

- The FTP server is started on the target Data Vault system.
- The Data Vault user account is added to the FTP user list.
- The FTP.ini file includes the required Data Vault parameters.
- The FTP.ini file is protected from unauthorized access. For example, use the `chmod 400 ftp.ini` command to protect the file.
- The FTP.ini file is included in the same directory as the ssadbcopy command line utility on the source Data Vault system.

Known Limitations

The ssadbcopy utility has the following known limitations:

You cannot use wildcards for the schema and table name.

You cannot migrate multiple databases (archive folders) at the same time. But you can migrate multiple databases at one time by running several ssadbccopy commands concurrently.

You cannot use FTP transfer mode from a Windows source to a Windows target. You can use the FTP transfer mode to a Unix target from a Unix or Windows source.

If the table structure in the source has changed, you must use the drop/create mode when you import the data in the target.

There are no run-time checkpoints. If the process fails, you must start the export and import from the beginning after you validate and fix the problem.

If the selected database, schema, or table has materialized views, the respective SCT files will be created in the local file system (mviewdir).

If the migration is performed at the schema or table level using the ssadbcopy utility, before migrating the database (from one Data Vault instance to the other), the SCT files of different tables should be present in only one storage type.

If the migration is performed at the table level using the ssadbcopy utility, all the SCT files of that particular table should be present in only one storage system. SCT files of other tables can be present in other storage systems.

Informatica supports the same version of the ssadbcopy utility for both export and import. For example, if you exported data with the ssadbcopy utility from version 6.4.3HF1, do not use the ssadbcopy utility from version 6.4.4 to import the same TAR file.

The ssadbcopy utility cannot be used on different operating systems. For example, you cannot export on one operating system and import on another operating sytem.

If you use the ssadbcopy utility for export on tables that have logically deleted rows, the ssadbcopy import fails with a check metadata error. You can ignore this error because the data is still migrated.

Steps to Partially Copy Data Between Data Vault Systems

Before you begin, verify the prerequisites for the source and target systems. To use an FTP server to transfer the export files, verify the automatic file transfer prerequisites.

1. Run the ssadbcopy command line utility in data export mode in the source Data Vault system.
2. If you did not use an FTP server to transfer the files to the target system, complete the following steps:
    a. Copy the TAR archive file to the target Data Vault system.

       You can find the TAR archive file in the following location:

       ```
       <temporary file directory>/idvmeta_backup_transfer_<timestamp>.tar
       ```

    b. Copy all of the data files to the target Data Vault system.

Use the data file list to view a list of all of the data files to transfer.

You can find the data file list in the following location:

```
<temporary file directory>/idvmeta_sctlist.txt
```

3. Run the ssadbopy command line utility in data import mode in the target Data Vault system.

Partial Data Vault Copy Example

You want to copy all of the data from one table to another Data Vault system. You want to use an FTP server to copy the data export files from the source to the target system. The source and target systems are both on Linux.

To copy data from one table, perform the following steps:

1. Configure the FTP.ini file on the source Data Vault system.
2. Run the ssadbcopy utility in the source Data Vault system to export the data.
3. Run the ssadbcopy utility in the target Data Vault system to import the data.

Source System Configuration

Before you begin, list the configuration that you want to use for the source Data Vault system.

The following table lists the configuration values that you want to use for the source Data Vault system:

Description	Value
Directory of the source Data Vault installation	/u01/vtruskov/ILM_FAS_6.1.1
Database from which you want to copy data	LOYALTY
Schema from which you want to copy data	LOYALTY_BLOB
Table from which you want to copy data	TICKET
Directory that stores the data export files	/u01/vtruskov/temp

Target System Configuration

Before you begin, list the configuration that you want to use for the target Data Vault system.

The following table lists the configuration values that you want to use for the target Data Vault system:

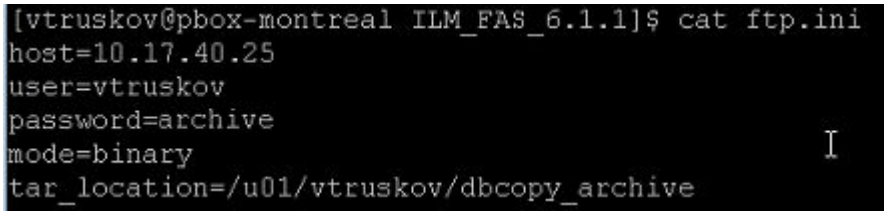
Description	Value
Directory of the target Data Vault installation	/u01/vtruskov/ILM-FAS
IP address that connects to the Data Vault	10.17.40.25
Data Vault user account that connects to the Data Vault	vtruskov

Description	Value
Directory that stores the data files	/u01/sct_stage
Directory that stores the TAR archive file	/u01/vtruskov/dbcopy_archive
Directory that stores extracted files from the TAR archive	/u01/vtruskov/temp

Step 1. Configure the FTP.ini File

When you run the ssadbcopy utility, you want the utility to automatically transfer the data export files to the target system. Configure the target system properties in the FTP.ini file. You save the FTP.ini file in the source system.

The following figure shows the FTP.ini file configuration:



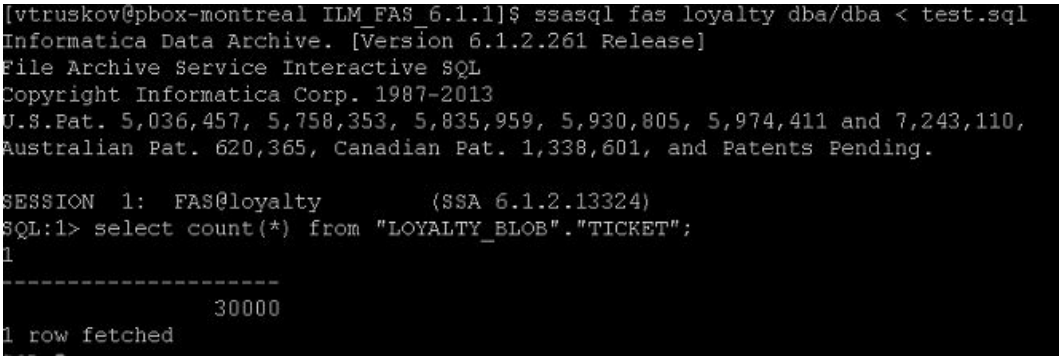
Next, export the data from the source system.

Step 2. Export the Data From the Source System

To export data from the source Data Vault system, run the ssadbcopy command in the source system.

Before you begin, you want to verify how many rows the source TICKET table includes. You run a query to count the rows in the TICKET table. The TICKET table includes 30,000 rows. After you copy the data to the target Data Vault system, you plan to run the same query on the target to confirm that the partial data copy is successful.

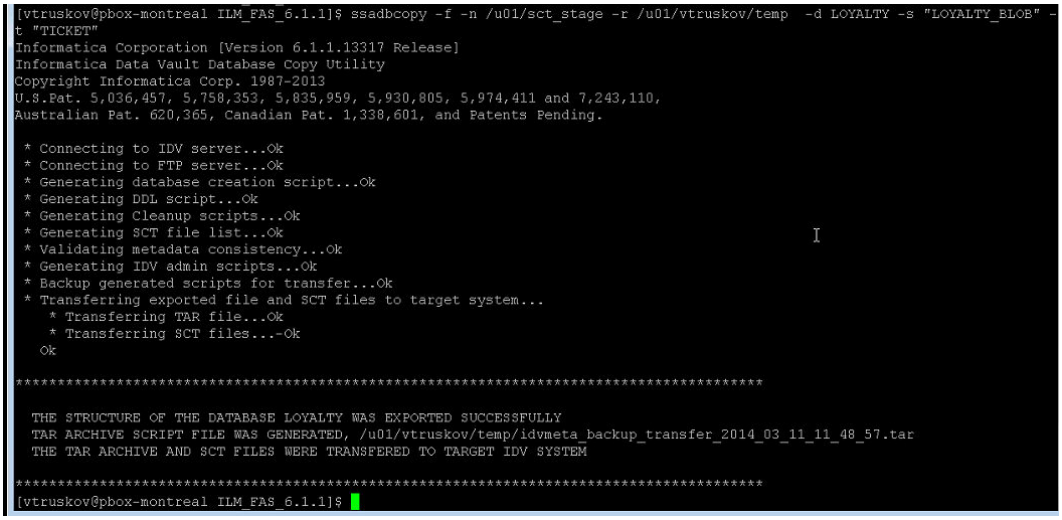
The following figures shows an example of the query and the query results:



To export the data, run the following ssadbcopy command in the source system:

```
ssadbcopy -f -n /u01/sct_stage -r /u01/vtruskov/temp -d LOYALTY -s "LOYALTY_BLOB" -t
"TICKET"
```

The following figure shows the command and the command output:

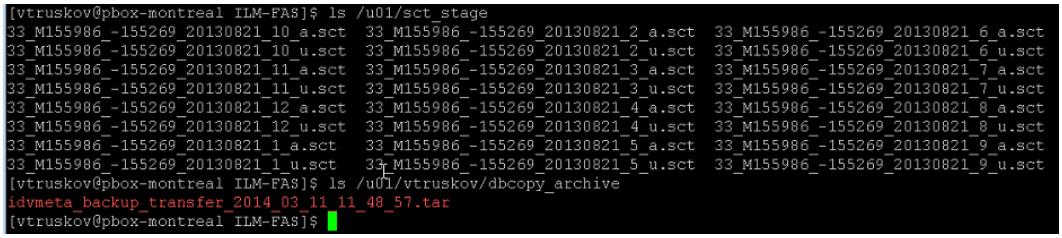


The utility generated the data export files in the source system directory `/u01/vtruskov/temp`.

The following figure shows the `/u01/vtruskov/temp` directory in the source system:

The utility used an FTP server to copy the data export files to the target system directory `/u01/vtruskov/dbcopy_archive`. The utility copied the data files to the `/u01/sct_stage` directory.

The following figure shows the directories for the data files and TAR archive file in the target system:



All the data export files exist in the target system. Next, import the data in the target system.

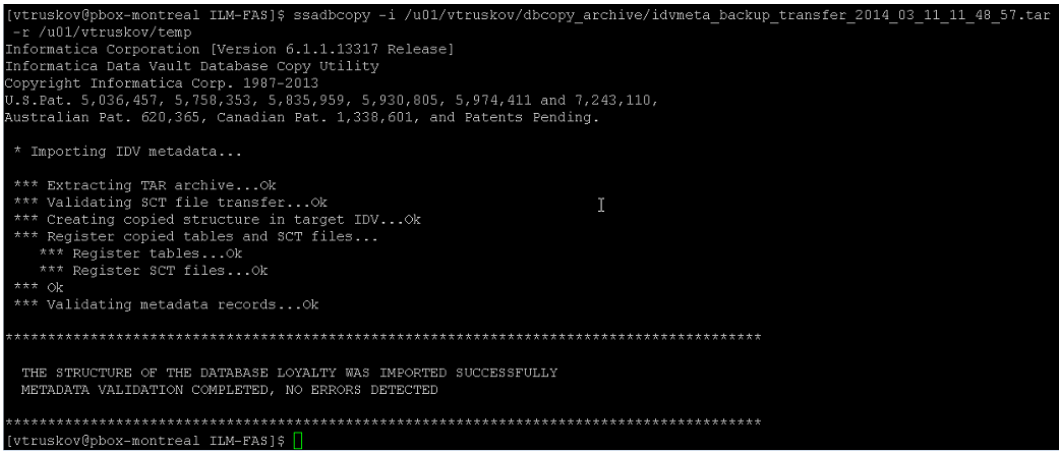
Step 3. Import the Data to the Target System

Run the ssadbcopy command in the target system.

You run the following ssadbcopy command in the target system:

```
ssadbcopy -i /u01/vtruskov/dbcopy_archive/
idvmeta_backup_transfer_2014_03_11_11_48_57.tar -r /u01/vtruskov/temp
```

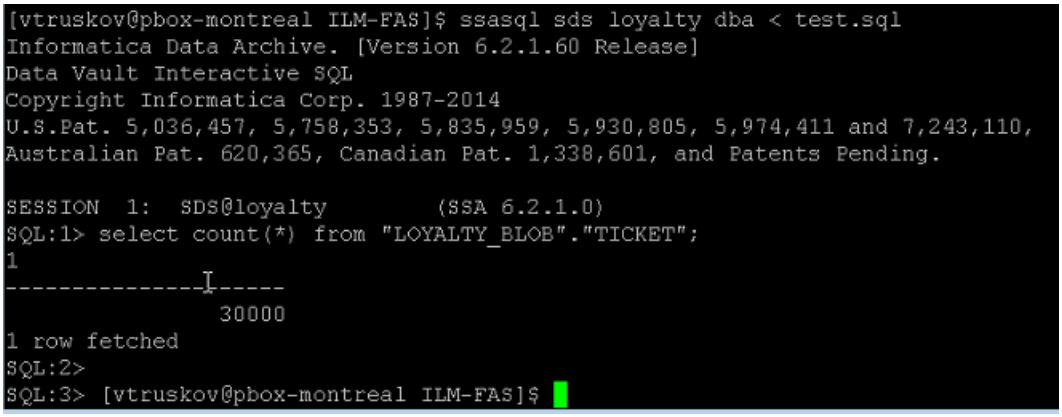
The following figure shows the command and the command output:



The utility extracts the files from the `idvmeta_backup_transfer_2014_03_11_11_48_57.tar` TAR archive file to the `/u01/vtruskov/temp` temporary file directory. The utility validates that all of the data files listed in the data file text file exist in the `/u01/sct_stage` data file directory. The utility creates the table structure in the target database. The utility registers the tables and data files in the Data Vault repository. The utility validates the metadata in the Data Vault repository for all objects in the table.

You want to verify that the target TICKET table includes the same amount of rows as the source TICKET table. The source TICKET table included 30,000 rows. You run a query to count the rows in the TICKET table in the target system. The TICKET table includes 30,000 rows. The tables include the same row count.

The following figures shows an example of the query and the query results:



CHAPTER 9

Archived Data Migration

This chapter includes the following topics:

- Archived Data Migration Overview, 127
- Archived Data Migration Process, 128
- Prerequisites for Archived Data Migration, 128
- ssamigrate Known Limitations, 129
- ssamigrate Syntax, 130
- ssamigrate Commands for Supported Target Systems , 131
- ssamigrate Example Scenarios, 132
- Archived Data Vault Migration Example, 133

Archived Data Migration Overview

Use the ssamigrate utility to migrate archived data from one storage system to another. Both storage systems must be configured within the same Data Vault instance. By default, the ssamigrate utility copies the SCT files to the new location, unregisters the SCT files in the source, and registers the files in the new location. You can specify that you want the utility to delete the SCT files from the source location. You can migrate data from one database, one schema, or one table at a time.

Use the ssamigrate utility to migrate archived data between the following Data Vault systems:

- From a local or network file system to another local or network file system.
- From a local or network file system to an external storage such as EMC Centera.
- From an external storage such as Amazon S3 to a local or network file system.
- From an external storage to another external storage.

For example, you retired an application to the local file system of a Data Vault instance. You now want to move the retired data to an external storage system that uses the Amazon S3 API version 2 storage protocol. Both the local file system and the external storage system are configured within the same Data Vault instance. You can use the ssamigrate command line utility to migrate the retired data without the need to re-archive the data.

When you use the ssamigrate utility to migrate data, the data files retain the original compression. The data is not even read during the migration process and remains immutable. The data migration is directly from the current target to the new target. You do not require an intermediate staging area.

The ssamigrate utility is located in the Data Vault installation folder on the machine where Data Vault is installed. Run the ssamigrage utility from the machine where Data Vault is installed.

When you migrate data from the source Data Vault system, you specify the data that you want to migrate. The utility first creates a temporary location to store and process files and then moves them to the new target Data Vault location.

Note: All the SCT files for a table must reside in the same storage type prior to the migration. For example, if a table has 10 SCT files, all 10 files must be in the local file system or in Hadoop.

If the migration process is interrupted, the ssamigrate utility will resume from the point of interruption. The utility will move any remaining files in the original storage and register the files in the new location.

To revert the migration process, run the ssamigrate command again but specify the original source as the new target.

Archived Data Migration Process

Run the ssamigrate command line utility to migrate archived data to another storage system that is configured within the same Data Vault instance. To migrate archived data, specify the data that you want to move and the location of the target system. The utility creates a log file and transfers the SCT files to the new location.

When you run the ssamigrate command to migrate archived data, the utility performs the following high-level steps:

1. Moves the SCT files from the source to the target locations.
2. Unregisters the SCT files in the source location.
3. Registers the SCT files in the target location.
4. Generates a log, report, and run time files and saves the files in a temporary file location.

Note: Ssamigrate does not change the status of SCT files. After migration, offline files remain as offline.

Prerequisites for Archived Data Migration

Verify the following prerequisites:

- The source and target storage locations are configured within the same Data Vault instance.
- The Data Vault Service is started.
- The ssamigrate command line utility is installed in the Data Vault root installation directory.
- Verify that there are no pending or running retirement and Data Vault loader jobs. Also verify that any compliance jobs (legal hold, retention, purge) are not running.

Additional Prerequisites for Storage Systems that use the Amazon S3 API Version 2 Protocol

If you are migrating archived data to a Data Vault external storage system that uses the Amazon S3 API Version 2 protocol, such as Amazon Web Services S3 or EMC Elastic Cloud Storage, complete the following prerequisites:

1. Install the libcurl API libcurl.so.4.2.0 and the dependent libraries in the following two directories:
   Data Vault Service Directory

   On Windows, install the files in the root folder of the Data Vault Service directory.

   On UNIX, install the files in the `/<File Archive Service Directory>/odbc` directory.

   Data Vault Service Agent Directory

   On Windows or UNIX, install the files in the root folder of the Data Vault Service agent directory.

   If the Data Vault Service agent is installed on multiple machines, install the files on each machine that hosts a Data Vault Service agent.
2. Add the Amazon Web Services S3 parameters in the ssa.ini file of the Data Vault Service.
   Add the following required parameters at the end of the `/<File Archive Service Directory>/ssa.ini` file:
   [AWS_CONNECTION <archive folder name>]

   Example for Amazon Web Services S3 where `test` is the name of the archive folder:

   ```
   [AWS_CONNECTION test]
   ```

   AWS_KEY=<Amazon Web Services key>:<Amazon Web Services secret key>

   Example for Amazon Web Services S3:

   ```
   AWS_KEY=AKIAI62XJDKZAY:MFx94W+ZlGTdEXom+21BKBh4Y4ZM792GiFuR6m
   ```

   Note: For the Data Vault Service to support the migration, the AWS_KEY value must not exceed 128 characters.
3. Optionally, you can validate the connection to the external storage from the Data Vault Service.
   Use the Data Vault Service ssadrv administration command to validate the connection details you added to the `ssa.ini` file.

   On the machine that hosts the Data Vault Service, run the following commands:

   ```
   ssadrv -a aws://<connection
   name>/<URL>/<Amazon Web Services bucket>/
   ```

   Example:

   ```
   ssadrv -a
   aws://test/https://s3.amazonaws.com/awsBucket/
   ```

ssamigrate Known Limitations

The ssamigrate utility has the following known limitations:

- You cannot use wildcards for the database, schema and table name.
- You cannot migrate multiple databases (archive folders) at the same time. You can only one database at a time.
- If the specified database, schema, or table has materialized views, the ssamigrate utility will not migrate the SCT files of the materialized views to the new location.

ssamigrate Syntax

The ssamigrate utility is located in the Data Vault installation folder on the machine where Data Vault is installed. Run the ssamigrage utility from the machine where Data Vault is installed to migrate archived data from one storage system to another within the same Data Vault instance.

The ssamigrate command uses the following syntax:

```
ssamigrate <flags>
```

Use flags to specify the migration task. You can list the flags in any order.

The following table shows the ssamigrate flag options:

The following table describes the ssamigrate flag options and command arguments:

Flag	Example	Description
-h	-	Optional. Displays onscreen help. The help shows the syntax of the command and the possible flags.
-c	-	Optional. Removes the SCT files from the source Data Vault location after the migration is complete. If you do not include -c in the command line, the ssamigrate utility does not remove the files from the source Data Vault location.
-l <file path of target storage location>	-l cas://atmos_cloud1/	Required. The full path of the target storage location that you want to migrate the SCT files to.
-d <database name>	-d RET_LOTYALTY	Required. The name of the database that you want to migrate data from. The database name is the same as the archive folder name in the Data Archive Data Vault target connection. You can migrate data from one database at a time.
-m <number of threads for sct migration>	-m 5	Optional. The number of threads for concurrent SCT file migration. Default is 10.
-n <number of threads for registration>	-n 3	Optional. The number of administrator threads for concurrent SCT file registration. Default is 5.
-r <temporary file location>	-r $SSA_INI_DIR/temp	Optional. The temporary directory on the target system for the log, report, and runtime files. By default, the ssamigrate utility saves the log, report, and runtime files in the same location that the ssamigrate utility is in.
-s <schema name>	- s LOYALTY_BLOB	Optional. The schema that you want to migrate. You can migrate data from one schema at a time. Use the full schema name. If you enter multiple schemas or use wildcards, the command fails. If you do not specify a schema, the utility migrates data from all schemas in the database that you specify in the -d option.

Flag	Example	Description
-t <table name>	-t TICKET	Optional. The table that you want to migrate. You can migrate data from one table at a time. If you enter multiple tables or use wildcards, the command fails. If you do not specify a table, the utility migrates data from all tables from the schema that you specify in the -s option.
-u <user ID>/ <password>	-u dba/dba	Required. User name and password of the Data Vault administrator user account.
-z <number of seconds>	-z 480	Optional. Extra number of seconds allocated for administrator operations such as the registration of SCT files. By default, the ssamigrate utility allocates 120 seconds for an administrator task. If you specify the -z option, then the ssamigrate utility adds the extra number of seconds that you specified to the default number of seconds. For example, if you specify `-z 480`, then the ssamigrate utility allocates up to 600 seconds or 10 minutes for a task to complete. If an administrator task does not complete in the allocated time, the ssamigrate utility creates a time-out.

ssamigrate Commands for Supported Target Systems

Use the ssamigrate commands for the target system that you want to migrate the archived data to.

The following list shows a sample ssamigrate command for each supported target storage platform.

Note: In the sample ssamigrate commands, RET_LOYALTY is the database that you want to migrate data from and dba/dba is the user name and password of the Data Vault administrator user account.

Amazon Web Services Version 2

```
ssamigrate -l aws://aws_v2/https://s3.amazonaws.com/IDVQABUCKET/ -d RET_LOYALTY -u
dba/dba
```

Generic AWS storage URL: aws://<connectionName>/<aws url>/<awsBucket>/

Amazon Web Services Version 4

```
ssamigrate -l aws://aws_v4/https://s3.eu-central-1.amazonaws.com/e2bi-coll-data-
archiving2/ -d RET_LOYALTY -u dba/dba
```

Generic AWS storage URL: aws://<connectionName>/<aws url>/<awsBucket>/

EMC Centera

```
ssamigrate -l cas://centera_cloud1/ -d RET_LOYALTY -u dba/dba
```

Generic EMC Centera URL: cas://<connectionName>/

EMC Atmos

```
ssamigrate -l cas://atmos_cloud1/ -d RET_LOYALTY -u dba/dba
```

Generic EMC Atmos URL: cas://<connectionName>/

Hadoop Distributed File System (HDFS)

```
ssamigrate -l hdfs://1/mnt/data/FASQA/ -d RET_LOYALTY -u dba/dba
```

Generic URL for HDFS storage: hdfs://<connectionName>/<hdfs file system>/

Hitachi Content Platform (HCP)

```
ssamigrate -l hcp://hcp_1/https://ns2.ten2.hcp-demo.hcpdomain.com/rest/ -d
RET_LOYALTY -u dba/dba
```

Generic URL for HCP: hcp://<connectionName>/https://<hcpDomain>/<namspace>/

ssamigrate Example Scenarios

You can run the ssamigrate command with one or more options.

Migrate data from one database

You want to migrate all the data from one database in the source Data Vault system to the target Data Vault system.

Run the following ssamigrate command:

```
ssamigrate -l cas://atmos_cloud1/ -d RET_LOYALTY -u dba/dba
```

The utility migrates the data and structure for database RET_LOYALTY. The utility saves the log, report, and runtime files to the `/data/temp` directory on the source system. The utility transfers the SCT data files to the target location `cas://atmos_cloud1/`.

Migrate data from one schema

You want to migrate data from one schema in the source Data Vault location with a local file system to a target Data Vault location with an external storage system.

Run the following ssamigrate command:

```
ssamigrate -l cas://atmos_cloud1/ -d RET_LOYALTY –s LOYALTY_AIR -u dba/dba
```

The utility migrates the data and structure for database RET_LOYALTY and schema LOYALTY_AIR. The utility saves the log, report, and runtime files to the `/data/temp` directory on the source system. The utility transfers the SCT data files to the target location `cas://atmos_cloud1/`.

Migrate data from one table

You want to migrate data from one table to the target Data Vault system.

Run the following ssamigrate command:

```
ssamigrate -l cas://atmos_cloud1/ -d RET_LOYALTY –s LOYALTY_AIR –t TICKET -u dba/dba
```

The utility migrates the data and structure for database RET_LOYALTY, schema LOYALTY_AIR, and table TICKET. The utility saves the log, report, and runtime files to the `/data/temp` directory on the source system. The utility transfers the SCT data files to the target location `cas://atmos_cloud1/`.

Archived Data Vault Migration Example

You want to migrate the archived data of one table from the Data Vault local file system to the storage system EMC Elastic Cloud Storage (EMC ECS) with Amazon S3 API version 2 storage protocol. Both the source and target storage systems are configured within the same Data Vault instance.

Source System Configuration

Before you begin, list the configuration that you want to use for the source Data Vault system.

The following table lists the configuration values that you want to use for the source Data Vault system:

Description	Value
Database from which you want to migrate data	123SDBCOPY
Schema from which you want to migrate data	LOYALTY_BLOB
Table from which you want to migrate data	TICKET
Directory that stores the log, report, and runtime files.	`/data/idvuser/Jenkins_slave/workspace/IDV64.3_SUSILinux64/temp`

Target System Configuration

Before you begin, list the configuration that you want to use for the target Data Vault system.

The following table lists the configuration values that you want to use for the target Data Vault system:

Description	Value
User name of account that connects to the Data Vault	dba
Password of account that connects to the Data Vault	dba
Directory that stores the data files	`cas://atmos_cloud1/`

Step 1. Migrate the Data from the Source to the Target System

Before you begin, run a query to get the file count and file details of the SCT files for the TICKET table. After you migrate the data to the target Data Vault system, you plan to run the same query on the target to confirm that the migration is successful.

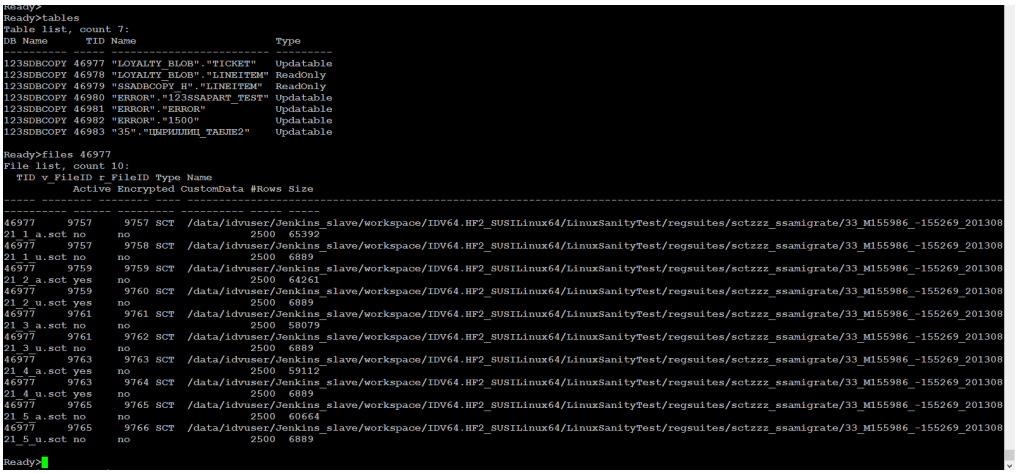
1. Get the list of tables for the database 123SDBCOPY. From the command prompt, run the following command:

   ```
   tables
   ```

2. Note the table ID for the TICKET table.
3. To get the list of files for the TICKET table, run the following command:

   ```
   files <table ID>
   ```

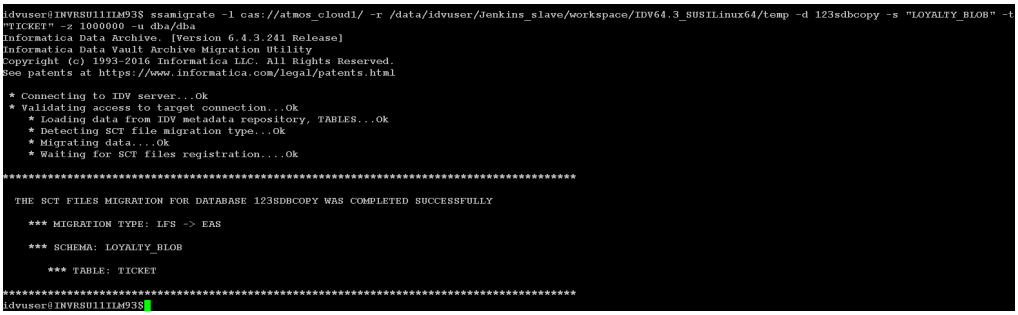
The following image shows the commands and the result:



4. Note the SCT file count and file details such as row count and file size.
5. Run the following ssamigrate command in the source system to migrate the data:

   ```
   ssamigrate -l cas://atmos_cloud1/ -r /data/idvuser/Jenkins_slave/workspace/
   IDV64.3_SUSILinux64/temp -d 123sdbcopy -s "LOYALTY_BLOB" -t "TICKET" -u dba/dba
   ```

   The following image shows the command and the result:



Step 2. Verify the Data in the Target System

Verify that all the SCT files for the table TICKET are now in the target system.

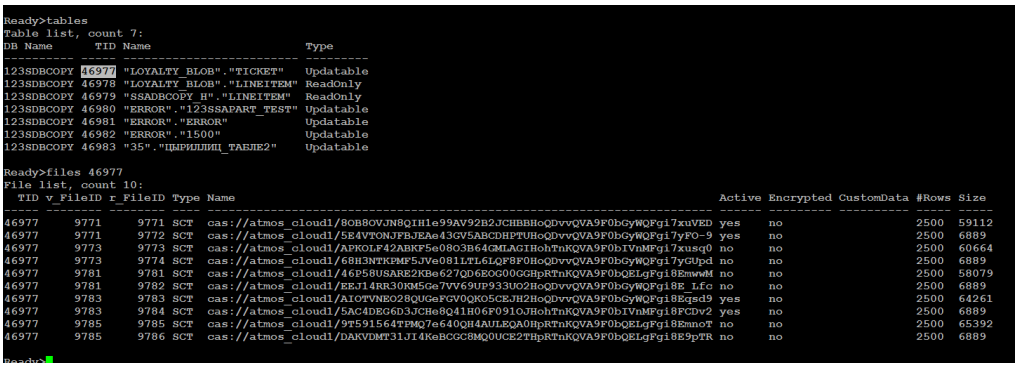
1. Get the list of tables for the database 123SDBCOPY. From the command prompt, run the following command:

   ```
   tables
   ```

2. Note the table ID for the TICKET table.
3. To get the list of files for the TICKET table, run the following command:

   ```
   files <table ID>
   ```

The following image shows the commands and the result:



```
Ready>tables
Table list, count 7:
DB Name      TID Name                        Type
---------- ----- ------------------------- ---------
123SDBCOPY 46977 "LOYALTY_BLOB"."TICKET"    Updatable
123SDBCOPY 46978 "LOYALTY_BLOB"."LINEITEM"  ReadOnly
123SDBCOPY 46979 "SSADBCOPY_H"."LINEITEM"   ReadOnly
123SDBCOPY 46980 "ERROR"."123SSAPART_TEST"  Updatable
123SDBCOPY 46981 "ERROR"."ERROR"            Updatable
123SDBCOPY 46982 "ERROR"."1500"             Updatable
123SDBCOPY 46983 "35"."ЦЫРИЛЛИЦ_ТАБЛЕ2"     Updatable

Ready>files 46977
File list, count 10:
  TID v_FileID r_FileID Type Name                                                                  Active Encrypted CustomData #Rows Size
----- -------- -------- ---- ---------------------------------------------------------------------- ------ --------- ---------- ----- -----
46977     9771     9771 SCT  cas://atmos_cloud1/8OB8OVJN8QIH1e99AV92B2JCHBBHoQDvvQVA9F0bGyWQFgi7xuVED yes    no                  2500  59112
46977     9771     9772 SCT  cas://atmos_cloud1/5E4VTONJFBJEAe43GV5ABCDHPTUHoQDvvQVA9F0bGyWQFgi7yFO-9 yes    no                  2500  6889
46977     9773     9773 SCT  cas://atmos_cloud1/APKOLF42ABKF5e08O3B64GMLAGIHohTnKQVA9F0bIVnMFgi7xusq0 no     no                  2500  60664
46977     9773     9774 SCT  cas://atmos_cloud1/68H3NTKPMF5JVe081LTL6LQF8F0HoQDvvQVA9F0bGyWQFgi7yGUpd no     no                  2500  6889
46977     9781     9781 SCT  cas://atmos_cloud1/46P58USARE2KBe627QD6EOG00GGHpRTnKQVA9F0bQELgFgi8EmwwM no     no                  2500  58079
46977     9781     9782 SCT  cas://atmos_cloud1/EEJ14RR30KM5Ge7VV69UP933UO2HoQDvvQVA9F0bGyWQFgi8E_Lfc no     no                  2500  6889
46977     9783     9783 SCT  cas://atmos_cloud1/AIOTVNEO28QUGeFGV0QKO5CEJH2HoQDvvQVA9F0bGyWQFgi8Eqsd9 yes    no                  2500  64261
46977     9783     9784 SCT  cas://atmos_cloud1/5AC4DEG6D3JCHe8Q41H06F091OJHohTnKQVA9F0bIVnMFgi8FCDv2 yes    no                  2500  6889
46977     9785     9785 SCT  cas://atmos_cloud1/9T591564TPMQ7e640QH4AULEQA0HpRTnKQVA9F0bQELgFgi8EmnoT no     no                  2500  65392
46977     9785     9786 SCT  cas://atmos_cloud1/DAKVDMT31JI4KeBCGC8MQ0UCE2THpRTnKQVA9F0bQELgFgi8E9pTR no     no                  2500  6889

Ready>
```

4. Verify that file count and file details match the file count and file details on the source system before the migration.

CHAPTER 10

Data Validation

This chapter includes the following topics:

- Data Validation Overview, 136
- Running the SSA Data Validation Utility on Servers Connected Over a Network, 136
- Running the SSA Data Validation Utility on Unconnected Servers, 138
- Additional Options for the ssadatavalidation Utility, 141

Data Validation Overview

Use the ssadatavalidation utility to validate data or metadata between two Data Vault servers. For example, after you migrate data from one version of the operating system to another version, you can use the utility to validate data between the two servers.

The utility verifies that the metadata and data in the source and target are the same and that there is no data loss during migration. You can use the utility to validate data between servers that are connected over a network and servers that are not connected. For example, you can migrate data between two on-premise machines that are connected over a network, or between an on-premise machine and cloud storage.

Running the SSA Data Validation Utility on Servers Connected Over a Network

Run the ssadatavalidation utility to validate data and metadata between two Data Vault servers.

Before you run the utility, ensure that you migrate the Data Vault.

1. Verify that the Data Vaults are enabled in the source and target machines.
2. Browse to the <Data Vault installation directory> on either the source or the target machine.
3. Run the ssadatavalidation.sh or ssadatavalidation.exe command depending on the operating system, on either the source or target with the following options:

Note: In the table, server 1 represents the source and server 2 represents the target.

Option	Description
-h	The Informatica Data Vault host name of server 1.
-p	The Informatica Data Vault port of server 1.
-q	The Informatica Data Vault exe port of server 1.
-u UID[/PWD]	The Informatica Data Vault user name and password of server 1.
-H	The Informatica Data Vault host name of server 2.
-P	The Informatica Data Vault port of server 2.
-Q	The Informatica Data Vault exe port of server 2.
-U UID[/PWD]	The Informatica Data Vault user name and password of server 2.
-r <dir>	The temporary directory for log, report, and run time files. All Informatica data validation logs, reports, or deviations are stored under the directory that you enter.
-v [0,1]	Compares based on column values without calculating checksum. Default is 0 or disabled.
-a [0,1]	Performs data validation on all eligible tables without exit on first failure. Set to 1 to continue validation even if errors are found.
-R [0,1]	Resumes the process. Default is 0 or disabled. You can choose to enable the -R option on all repeat utility runs.
-m nnn	The number of threads for data validation. Default is 4.

For example:

```
ssadatavalidation.exe -h INKW12DSG462 -p 3300 -q 3400 -u dba/dba -H INVW16DSG1298 -P
3300 -Q 3400 -U dba/dba -r "C:\IDV" -a 1 -v 0 -R 0 -m 5
```

4. If the utility fails, enable the -R option and run the utility again.

   For example:

   ```
   ssadatavalidation.exe -h INKW12DSG462 -p 3300 -q 3400 -u dba/dba -H INVW16DSG1298 -P
   3300 -Q 3400 -U dba/dba -r "C:\IDV" -a 1 -v 0 -R 1 -m 5
   ```

   The resume.txt file indicates the step from which the utility resumes. The tables_list_results.txt file stored in the temp folder on the source and target machines indicates the table from which the utility resumes.

After the validation is successful, you can use the Data Vault servers.

Running the SSA Data Validation Utility on Unconnected Servers

Run the ssadatavalidation utility to validate data and metadata between two Data Vault servers that are not connected over a network.

Before you run the utility, ensure that you migrate the Data Vault.

1. Verify that the Data Vaults are enabled in the source and target machines.
2. Browse to the <Data Vault installation directory> on the source machine.
3. Run the ssadatavalidation.sh or ssadatavalidation.exe command on the source machine with the following options:

   In the table, Server 1 represents the source machine and Server 2 represents the target machine.

Option	Description
-r <dir>	The temporary directory for log, report, and run time files. All Informatica data validation logs, reports, or deviations are stored under the directory that you enter.
-h	The Informatica Data Vault host name of server 1.
-p	The Informatica Data Vault port of server 1.
-q	The Informatica Data Vault exe port of server 1.
-u UID[/PWD]	The Informatica Data Vault user name and password of server 1.
-i [0,1]	Generates intermediate files. Use the -h, -p, -u, x options or the -H, -P, -Q, -U, -K options with the option.
-v [0,1]	Compares based on column values without calculating checksum. Default is 0 or disabled.
-a [0,1]	Performs data validation on all eligible tables without exit on first failure.
-R [0,1]	Resumes the process. Default is 0 or disabled. You can choose to enable the -R option on all repeat utility runs.
-m nnn	The number of threads for data validation. Default is 4.

   For example:

```
ssadatavalidation.exe -i 1 -h INKW12DSG462.informatica.com
 -p 3300 -q 3400 -r "C:\INKW12DSG462" -u dba/dba -a 1 -v 0 -R 0 -m 5
```

4. Browse to the <Data vault installation directory> on the target machine.

5. Run the ssadatavalidation.sh or ssadatavalidation.exe command on the target machine with the following options:

Option	Description
-r <dir>	The temporary directory for log, report, and run time files. All Informatica data validation logs, reports, or deviations are stored under the directory that you enter.
-H	The Informatica Data Vault host name of server 2.
-P	The Informatica Data Vault port of server 2.
-Q	The Informatica Data Vault exe port of server 2.
-U UID[/PWD]	The Informatica Data Vault user name and password of server 2.
-i [0,1]	Generates intermediate files. Use the -h, -p, -u, x options or the -H, -P, -Q, -U, -K options with the option.
-v [0,1]	Compares based on column values without calculating checksum. Default is 0 or disabled.
-a [0,1]	Performs data validation on all eligible tables without exit on first failure. Set to 1 to continue validation even if errors are found.
-R [0,1]	Resumes the process. Default is 0 or disabled. You can choose to enable the -R option on all repeat utility runs.
-m nnn	The number of threads for data validation. Default is 4.

For example:

```
ssadatavalidation.exe -i 1 -H INVW16DSG1298 -P 3300
-Q 3400 -r "C:\ INVW16DSG1298" -U dba/dba
-a 1 -v 0 -R 0 -m 5
```

Note: It is recommended to run the utility on the source and target in parallel.

6. If the utility fails on either the source or target, enable the -R option and run the utility again.

For example:

```
On Server 1:
ssadatavalidation.exe -i 1 -h INKW12DSG462 -p 3300 -q 3400
 -r "C:\INKW12DSG462" -u dba/dba -a 1 -v 0 -R 1 -m 5

On Server 2:
ssadatavalidation.exe -i 1 -H INVW16DSG1298 -P 3300 -Q 3400
-r "C:\ INVW16DSG1298" -U dba/dba -a 1 -v 0 -R 1 -m 5
```

The resume.txt file indicates the step from which the utility resumes. The tables_list_results.txt file stored in the temp folder on the source and target machines indicates the table from which the utility resumes.

7. Back up the ssadatavalidation_temp directories on the source and target. The directory is deleted when you run the utility to compare results. Backing up the directory helps to avoid running the utility again.
8. Compare the tables_list_results.txt file stored in the ssadatavalidation_temp directory for both the source and target. Verify that the status of all tables is READY.

   If a table fails and has a status other than READY in both files and is the expected state, you can proceed to the next step. For example, the sct file might exist in the source but not in the target.

If a table status is READY in one file but not in the other, run the utility again with the Resume option on the failed tables to correct the issue and then continue.

9. Copy or move the common folder that you specified with the -r option from the source to the target machine or target to the source machine. The folder must be available on the machine where you run the utility to compare results.

   To avoid errors because of disk space issues when you run the utility, run the utility on the machine that has more storage space available. To ensure faster execution and copying of files, copy temp files from a cloud virtual machine to an on-premise machine.

10. Run the ssadatavalidation utility again to compare checksum or result files on the server where you copy the common folder. Use the following options when you run the utility:

Option	Description
-h	The Informatica Data Vault host name of server 1.
-f	The parent directory of the Informatica Data Vault server 1 directory. Do not use this option with -p, -q, -u or -x options.
-H	The Informatica Data Vault host name of server 2.
-F	The parent directory of the Informatica Data Vault server 2 directory. Do not use this option with -P, -Q, -U or -X options.
-v [0,1]	Compares based on column values without calculating checksum. Default is 0 or disabled.
-a [0,1]	Performs data validation on all eligible tables without exit on first failure.
-r <dir>	The temporary directory for log, report, and run time files. All Informatica data validation logs, reports, or deviations are stored under the directory that you enter.

   For example:

```
ssadatavalidation.exe -h INKW12DSG462 -f "C:\INKW12DSG462\ssadatavalidation_temp"
 -H INVW16DSG1298 -F "C:\INVW16DSG1298\ssadatavalidation_temp"
-a 1 -R 0 -r "C:\IDV644_WIN12_WIN16"
```

   Note: Use separate paths for the -r and -f or -F options.

11. If the utility fails when you run the utility to compare the checksum, enable the -R option and run the utility again. This performs the validation only for tables whose checksum comparison failed in the first run.

    For example:

```
ssadatavalidation.exe -h INKW12DSG462 -f "C:\INKW12DSG462\ssadatavalidation_temp"
 -H INVW16DSG1298 -F "C:\INVW16DSG1298\ssadatavalidation_temp\INVW16DSG1298"
-a 1 -R 1 -r "C:\IDV644_WIN12_WIN16"
```

    The resume.txt file indicates the step from which the utility resumes. The tables_list_results.txt file stored in the temp folder on the source and target machines indicates the table from which the utility resumes.

After the validation is successful, you can use the Data Vault servers.

Additional Options for the ssadatavalidation Utility

You can use the following additional options when you run the ssadatavalidation utility:

Option	Description
-d	The name of the database to process. Add the option to perform validation at the database level.
-s	The name of the schema in the database to process. Default is all schemas. Add the option to perform validation at the schema level.

-t	The name of the table in the schema to process. Add the option to perform validation at the table level. Default is all tables.
	Note: To validate at the table level, you must include options -d and -s.
	For example, on connected servers:
	`ssadatavalidation.exe -h INKW12DSG462 -p 3300 -q 3400 -u dba/dba -H INVW16DSG1298 -P 3300 -Q 3400 -U dba/dba -r "C:\IDV" -a 1 -v 0 -R 1 -m 5 -d TEST_DATABASE1 -s TEST_SCHEMA1 -t TEST_TABLE1`
	On unconnected servers:
	`ssadatavalidation.exe -i 1 -h INKW12DSG462-p 3300 -q 3400 -r "C:\INKW12DSG462" -u dba/dba -a 1 -v 0 -R 1 -m 4 -d TEST_DATABASE1 -s TEST_SCHEMA1 -t TEST_TABLE1`

-v[0,1]	Enables or disables validation at the column level. Default is 0. Enter 1 to enable.
	For example, on connected servers:
	`ssadatavalidation.exe -h INKW12DSG462 -p 3300 -q 3400 -u dba/dba -H INVW16DSG1298 -P 3300 -Q 3400 -U dba/dba -r "C:\IDV" -a 1 -v 1 -R 1 -m 5`
	On unconnected servers:
	`ssadatavalidation.exe -i 1 -h INKW12DSG462-p 3300 -q 3400 -r "C:\INKW12DSG462" -u dba/dba -a 1 -v 1 -R 1 -m 4`
	The value of option -v must be the same when you run the utility on each server.

-n nnn	The number of rows in each partition. Requires the -v option. Default is 20000. For example: `ssadatavalidation.exe -h INKW12DSG462 -p 3300 -q 3400 -u dba/dba -H INVW16DSG1298 -P 3300 -Q 3400 -U dba/dba -r "C:\IDV" -a 1 -R 0 -m 5 -n 30000 -v 1`

Option	Description

-k[0,1]	Retain temporary files. Default is 0.
	For example, on connected servers:
	`ssadatavalidation.exe -h INKW12DSG462 -p 3300 -q 3400 -u dba/dba -H INVW16DSG1298 -P 3300 -Q 3400 -U dba/dba -r "C:\IDV" -a 1 -v 0 -R 1 -m 5 -k 1`
	On unconnected servers:
	Server 1
	`ssadatavalidation.exe -i 1 -h INKW12DSG462.informatica.com -p 3300 -q 3400 -r "C:\INKW12DSG462" -u dba/dba -a 1 -v 0 -R 1 -m 5 -k 1`
	Server 2
	`ssadatavalidation.exe -i 1 -H INVW16DSG1298.informatica.com -P 3300 -Q 3400 -r "C:\INVW16DSG1298" -U dba/dba -a 1 -v 0 -R 1 -m 5 -k 1`

-o[0,1]	Records differences in a diff file corresponding to a table. Set the option to 0 to record only the first difference that occurs. Set the option to 1 to record all differences. Default is 0. Use if column data validation identifies deviations. **Note:** To set the option, you must include options -d and -s.

If you run the utility for a table or column that includes special characters in the name on a Windows environment, enter the table and column names in double quotes. Verify that Informatica Data Vault supports the special characters.

For example, on connected servers:

```
ssadatavalidation.exe -h INKW12DSG462 -p 3300 -q 3400
-u dba/dba -H INVW16DSG1298 -P 3300 -Q 3400
-U dba/dba -r "C:\IDV" -a 1 -v 0 -R 1 -m 5 -d TEST_DATABASE1
-s TEST_SCHEMA1
-t "A_B$C~D!E@F^G&H-I+J=K:L<M>N?O|Q{R}"
-c "A_B$C~D!E@F^G&H-I+J=K:L<M>N?O|Q{R}"
```

On unconnected servers:

```
ssadatavalidation.exe -i 1 -h INKW12DSG462-p 3300 -q 3400 -r "C:\INKW12DSG462"
-u dba/dba -a 1 -v 0 -R 1 -m 4 -d TEST_DATABASE1 -s TEST_SCHEMA1
-t "A_B$C~D!E@F^G&H-I+J=K:L<M>N?O|Q{R}" -c "A_B$C~D!E@F^G&H-I+J=K:L<M>N?O|Q{R}"
```

Run the utility on connected servers when SSL is enabled

Add the following options to run the utility:

Option	Description
-x	The absolute path of the certificate file for server 1.
-X	The absolute path of the certificate file for server 2.

If you run the utility from the target machine, the Truststore files must be copied from the source machine to the target machine.

If you run the utility from the source machine, copy the target certificate to the source machine and then import into cacerts on the source machine. If you run the utility from the target machine, copy the source certificate to the target machine and then import into cacerts on the target machine.

For example: `keytool -importcert -file mycertfile.pem -keystore cacerts -alias "Alias"`

```
ssadatavalidation.exe -h INKW12DSG462 -p 3300 -q 3400 -u dba/dba
 -H INVW16DSG1298 -P 3300 -Q 3400 -U dba/dba -r "C:\IDV"
-a 1 -v 0 -R 1 -m 5 -x "C:\OpenSSL\bin\private_cert.pem"
-X "D:\OpenSSL\bin\private_cert.pem"
```

When you create the certificates, you must provide the -h and -H options with the common name.

Run the utility on unconnected servers when SSL is enabled

Add the following options to run the utility:

Option	Description
-x	The absolute path of the certificate file for server 1.
-X	The absolute path of the certificate file for server 2.

For example:

Server 1

```
ssadatavalidation.exe -i 1 -h INKW12DSG462.informatica.com
 -p 3300 -q 3400 -r "C:\INKW12DSG462" -u dba/dba
-a 1 -v 0 -R 1 -m 5 -x "C:\OpenSSL\bin\private_cert.pem"
```

Server 2

```
ssadatavalidation.exe -i 1 -H INVW16DSG1298.informatica.com
-P 3300 -Q 3400 -r "C:\INVW16DSG1298" -U dba/dba
-a 1 -v 0 -R 1 -m 5 -X "C:\OpenSSL\bin\private_cert.pem"
```

CHAPTER 11

Bulk File Uploader

This chapter includes the following topics:

- Bulk File Uploader Overview, 144
- Master XML File, 145
- Source Data Flat Files, 146
- Bulk File Uploader Components, 146
- Logging, 149
- Configuration Parameters, 149
- Running the Bulk File Uploader, 150
- DDL Creation and Limited Alter Table Support, 156
- Discovery and Compliance from Data Archive, 156
- Data Type Mapping, 157
- System Requirements, 158
- Known Limitations, 158

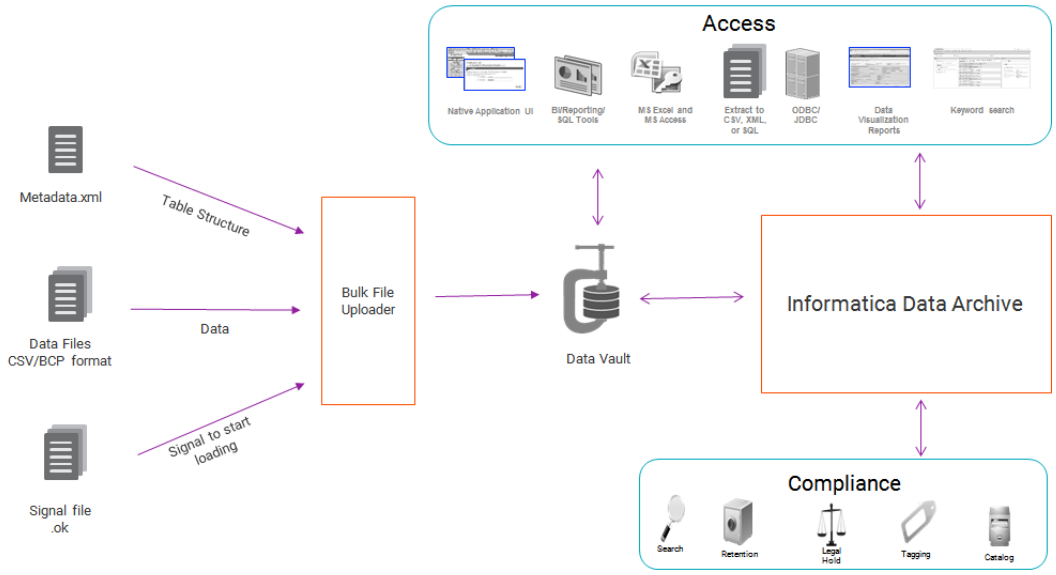
Bulk File Uploader Overview

The bulk file uploader is a high-throughput file archive application for Data Vault. It loads extracted flat files, for example BCP or CSV, to the Data Vault with a lightweight loading mechanism.

When you use the bulk file uploader, you provide the extracted flat files and the source table metadata, in a master XML file. The bulk file uploader generates the table DDL and then loads data from the flat files into Data Vault. After the bulk file uploader loads the data, you can query it through ODBC/JDBC applications. You can also use all of the available Data Archive functionality on the data retired by the bulk file uploader. For an overview of the Data Archive integration process, see the "Discovery and Compliance from Data Archive" topic in this chapter.

Use the bulk file uploader when you want to archive large amounts of data to the Data Vault.

The following diagram illustrates the bulk file uploader architecture:



The bulk file uploader is supported for Linux and Microsoft Windows platforms.

Master XML File

Before you can use the bulk file uploader, you must create a master XML file that contains the metadata information for the source tables.

The master XML file contains the table attributes and a naming pattern for the source files. You can update the XML file to add or change tables irrespective of whether the bulk file uploader is running. Ssaservice, the bulk file uploader scheduler process, reloads the latest metadata from the master XML file whenever you update the file.

You must name the master XML fille "`metadata.xml`" and add the file's path as a value for the TASK_DIRECTORY parameter under the [SERVICE] section in the ssa.ini file. For example:

```
TASK_DIRECTORY = /data/IDV/MASTERXMLDIR
```

Some characters have a special meaning in XML files. Provide the following replacement constants to represent these characters:

Character	Replacement Value
&	&
<	<
>	>
TAB character	

Source Data Flat Files

Before you run the bulk file uploader, save the flat files that contain the source data to a directory accessible to Data Vault. When you are ready to run the uploader, create a signal file in the same directory.

The source data flat files can have any extension. The rows can have any string as the row or column delimiter, as long as the delimiter is supported by Data Vault. The number of rows in each flat file file must be less than 2 billion.

The order of column data in the flat file must match the order of columns in the master XML file, because ssaservice uses the ordering information when it parses the flat file to get column values. For optimal Data Vault performance, the recommended size of a flat file is from 100MB to 5GB and the recommended number of rows is 1 million to 10 million.

You must create a signal file named `<BCP-FILE-NAME>.ok` to signal to the bulk file uploader that the flat file is ready to be loaded to Data Vault. When you are ready to run the bulk file uploader, save the `<BCP-FILE-NAME>.ok` file in the directory where the flat file is located. This directory is specified by the FILE_PATH value in the master XML file. The contents of the file itself are irrelevant and you can leave the file empty.

Bulk File Uploader Components

The bulk file uploader contains four components.

Ssaservice

Scheduling component of the bulk file uploader.

Ssajob

Standalone job executor.

Ssamon

Monitoring utility.

Addjob

Interface for scheduling standalone archive jobs.

Ssaservice Component

Ssaservice is a daemon process. It parses the master XML file to get metadata information for the source tables. It also searches the required paths to find the list of flat files (source files exported from the database) to be loaded into each table, creates an archive job for each flat file, and adds the load job to the executor queue. Ssaservice passes required metadata information to ssajob through a temporary table-level XML file.

Ssaservice stores some internal information in the Data Vault metadata repository tables. During startup, ssaservice checks the current bulk file uploader metadata version and upgrades to the latest version. This metadata information is independent of the Data Vault metadata version.

Ssaservice uses the following command line syntax:

```
ssaservice -h (displays the help information)
ssaservice start (starts the service process)
ssaservice install (installs the daemon script "ssaserviced" in the Linux operating
system)
```

On LINUX, the `ssaservice install` command creates a daemon script called "ssaserviced."

On Microsoft Windows, when you run `ssaservice install`, you are asked if you want to "use system account." If you select yes, the `ssaservice install` command will install ssaservice as a Windows service. To start the Windows service, use the command "`start ssaservice start`."

Informatica recommends that you stop the ssaservice with the ssamon utility. "Shutdown deferred" mode in ssamon terminates ssaservice after all of the running jobs are complete. Alternatively, you can stop ssaservice with "shutdown immediate." This command shuts down ssaservice and terminates any running jobs.

Ssajob Component

Ssajob is a standalone job executor process. It receives the temporary table-level XML file and creates the table DDL and NDL files. Then it starts the Data Vault loader to create SCT files from the source files, and registers the SCT files.

You do not need to directly call ssajob. The bulk file uploader calls ssajob to load the files into Data Vault. To schedule archive jobs, use the addjob interface.

Ssamon Component

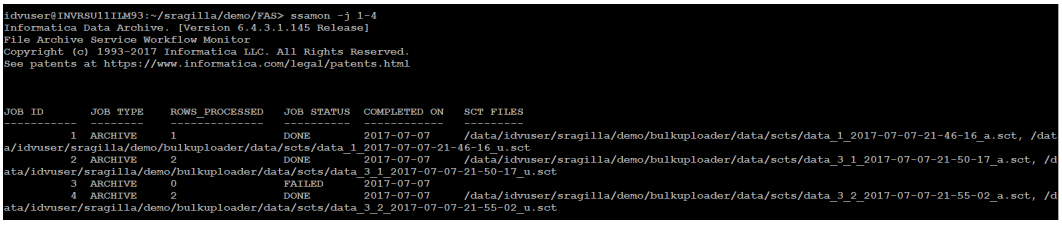
The ssamon monitoring utility tracks the running, completed, and failed upload jobs. It returns the job status based on job ID, date range, or daily runs.

Ssamon uses the following command line syntax:

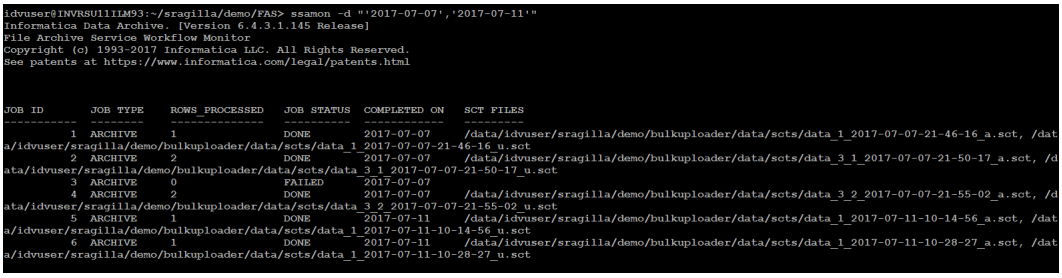
Option	Description
-h	Prints the help screen.
-f	Sends a force kill command to ssaservice.
-j	Prints the current status for the specified job ID, job IDs, or range. For example: ssamon -j 10 ssamon -j 10, 11, 12 ssamon -j 10-80 ssamon -j 10-
-d	Prints the job status for a specified date, dates, or date range in "'yyyy-mm-dd'" format. For example: ssamon -d "'2017-04-19'" ssamon -d "'2017-04-19','2017-04-20','2017-04-21'" ssamon -d "'2017-04-19'-'2017-04-21'" ssamon -d "'2017-04-19'-"
-r	Prints a daily job report from log.
-k	Sends a force kill command to the specified job ID.
-o	Prints a daily report output file.

Option	Description
shutdown immediate	Shuts down ssaservice and terminates running jobs.
shutdown preferred	Shuts down ssaservice after any running jobs are complete.

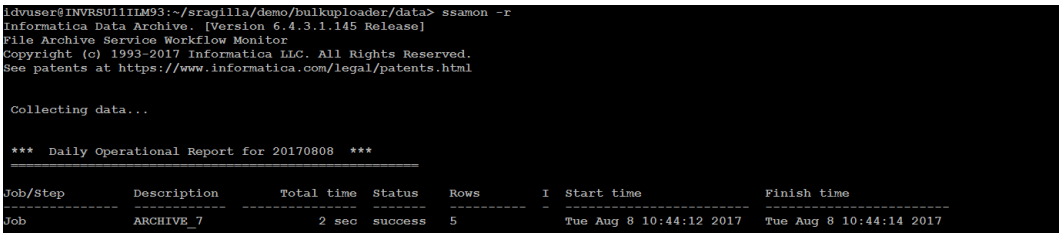
The following graphic is an example of the current status log generated by the `-j` option:



The following graphic is an example of the current status log generated by the `-d` option:



The following graphic is an example of the current status log generated by the `-r` option:



Addjob Component

Use the addjob component to schedule standalone archive jobs.

Addjob requires the metadata database connection details and a configuration file with the NDL file, flat file, and SCT file locations, in addition to the Data Vault connection details. The `-a` option creates the archive job, while the `-s` option allows you to give the required job details.

Addjob does not create DDL. It expects that you have previously created the table structure in Data Vault.

Addjob uses the following command line syntax:

Option	Description
-h	Displays the help screen.
-a	Creates an archive job.
-a -k	Keeps the source data file after the archive job is complete.
-a -s	<ConfFileName> <MetaConnection> <MetaUser/MetaPassword> Allows you to provide the configuration file name and meta database connection details. For example: `addjob -a -s test.conf meta_fb dba/dba`

`-k` is optional, while `-a` and `-s` are required.

Sample Configuration File

The following text is an example of the .conf file required by the addjob utility:

```
NDLFILE=/data/idvuser/gyana/test_model.ndl
FLATFILE=/data/idvuser/gyana/one_million.csv
SCTFILE=/data/idvuser/gyana/one_million.sct
CONNECTION=fas ADM dba/dba
```

Logging

There are 3 levels of logging for the bulk file uploader.

The first level of logging is saved in the directory referenced by the SSASERVICELOG parameter in the ssa.ini file. It creates ssaservice process logs, for example the events that occurr in ssaservice and any errors that occur when ssaservice parses the master XML file. The file name format is `ssaservice-<TIMESTAMP>.log`.

The second level of logging is a summary report of the log in the archive directory. Each job has one summary report file.

The third level of logging is the detailed job-level log files. The files are for DDL, loader, register, and more. The job logs are stored in the directory referenced by the SSAJOBLOG parameter in the ssa.ini file.

Configuration Parameters

To use the bulk file uploader, configure the following parameters in the [SERVICE] section of the ssa.ini file:

TASK_DIRECTORY

Defines the directory that contains the master XML file. You can provide any directory that has read/write permissions.

SSAJOBLOG

Defines the directory in which the job-level logs are created.

SSASERVICELOG

Defines the directory in which the service log files are created.

MAXJOBS

The number of bulk file uploader jobs that can run in parallel. The default value for this parameter is five. If you increase this parameter, consider also increasing the server threads to avoid starvation of archive jobs.

For example:

[SERVICE] MAXJOBS = 10

LOADPARAM

A list of comma separated values that the bulk file uploader uses as command line arguments. The syntax is <ARG1>: <VAL1>, <ARG2>: <VAL2>, ..., <ARGN>: <VALN>

k

The number of parallel threads used for creating the SCT file.

Mem

The maximum memory in MB that the uploader can use for one loading job.

lr

Passes outlier values to the bulk file uploader in the same way that other values are passed to the uploader.

For example:

[SERVICE]

LOADPARAM = k : 2 , lr:'TIMESTAMP:R:99:99:9999:99:99:99
9999999:2016-07-21-00.23.23.000000000000'

Example

The following text is an example of the bulk file uploader configuration parameters in the ssa.ini file:

```
[SERVICE]
SSA_Connection = fas
SSAJOBLOG = E:\6431\fas_logs
ADDRESS =
LOADBUFFER = 32768
SSA_UID = dba
SSA_Database = meta
EMAIL = 0
LOADPARAM = j:2,k:2,lr:'INT:R:-0001,DOUBLE:R:-1.1E+111,TIMESTAMP:R:
0001-01-01-01.09.01.000000000001'
JOBRETRY = 0
TIMEOUT = 15
SSA_PWD = B1E786394FB4762B9A83EE2329162A59
SSASERVICELOG = E:\6431\fas_logs
TASK_DIRECTORY = E:\sct_ssaservice\mdata
```

Running the Bulk File Uploader

To load the data into Data Vault using the bulk file uploader, complete the following steps:

1. Configure the ssa.ini file.
2. Populate the master XML file.

3. Start ssaservice.
4. Submit the source data file for the archive job.
5. Monitor the job status.

Step 1. Configure the Ssa.ini File

When the bulk file uploader starts, it requires the location of the master XML file. Add the TASK_DIRECTORY parameter in the [SERVICE] section of the ssa.ini file and provide the path of the master XML file.

For example:

```
[SERVICE]
......
......
TASK_DIRECTORY = /data/IDV/MASTERXMLDIR
```

Refer to the topic "Configuration Parameters" to review other ssa.ini parameters that impact logging and loading to Data Vault.

Step 2. Populate the Master XML File

Populate the master XML file with the table attributes and a naming pattern for the source files.

Name the master XML file "`metadata.xml`." All tags in the master XML file must be in UPPER case characters.

The following table describes the XML file tags:

XML Tag	Purpose
IDVSERVICE_DOCUMENT	Root tag for the document. Must be present at the beginning and end of the file. If the tag is not present, Data Vault assumes the document is incomplete.
SERVICE_CALL	Specifies the service parameters JOB_TYPE and SOURCE.
JOB_TYPE	The type of job. Currently only the ARCHIVE job type is supported for this tag.
SOURCE	SOURCE can be either XML or ODBC. If the source is XML, ssaservice expects you to provide the table and column attributes in the master XML file. If the SOURCE is ODBC, ssaservice uses the DSN to connect to the source database and get the table attributes such as column name, data type, precision, scale, and nullability information. Currently, only Rainstor is supported as a source for an ODBC connection.
DSN	The DSN for fetching table attributes from the source database. Valid when SOURCE is ODBC.
DSNUID	User name for the source database. Valid when SOURCE is ODBC.
DSNPWD	Password for the source database. Valid when SOURCE is ODBC.
GLOBAL_PARAM	Parent tag for the global parameters effective for each job. These global parameters can be overridden by table level parameters.

XML Tag	Purpose
KEEP_DATA	If the value is 0, the bulk file uploader deletes the flat file after the job is complete. For any other value, the source file will be retained after the job is complete. If the job fails due to any reason, the source file will not be deleted.
TIMESTAMP_FORMAT	Specifies the format for timestamp data in the flat file, if the format is different from the Data Vault default format (yyyy-mm-dd-HH.MM.SS.NNNNNNNNNNN). The timestamp format supports the following special characters as separators: - **:** - **/** - **.** - **-** - **space** For example: yyyy/mm/dd-mm/ss/hh/NNNN or yyyy mm dd-mm ss hh:NNNN
COLUMN_SEPARATOR	String that separates column values in the source flat file.
ROW_SEPARATOR	String that separates rows in the source flat file.
NULL_INDICATOR	String to indicate NULL in the source flat file.
CRYPTO_KEY	Crypto key used for encrypting the SCT file. The crypto key value is visible in the `metadata.xml` file.
TABLES	Parent tag for a list of tables.
TABLE	Tag for the beginning of the table attribute. The value for NAME is the table name, DATABASE is the database name, and SCHEMA is the schema name.
FILE_PATH	Specifies a directory path with a naming pattern for source files.
SCT_PATH	The directory where SCT files will be created.
COLUMNS	List of columns. This information is used for generating the table DDL and NDL.
COLUMN	Parent tag for column attributes. The NAME value signifies the column name.
TYPE	The source data type. Only Data Vault and Rainstor types are currently supported.
PRECISION	Precision or length of the column.
SCALE	Scale value for appropriate data types.
NULLABLE	1 if the column is nullable, 0 if the column is not nullable.

When you define an attribute at the table level and at the global level, the table level value takes priority.

The XML file must contain all attributes of the columns TYPE, PRECISION, SCALE, and NULLABLE. The bulk file uploader uses the precision and scale depending on the data type. Informatica recommends that you create the columns as nullable. For example, for the DECIMAL data type, the uploader uses both precision and scale. For the CHAR data type, the uploader uses only precision. For the INT data type, the uploader ignores both precision and scale.

When you populate the master XML file, you can use either Data Vault data types or Rainstor data types. If you use Rainstor data types, the uploader converts them to Data Vault data types as shown in the topic "Data Type Mapping."

The maximum length supported by Data Vault of CHAR and VARCHAR columns is 32768. If the precision provided for CHAR or VARCHAR columns exceeds 32768 in `metadata.xml`, the uploader adjust the precision in the DDL to 32768. If the actual data is longer than 32768 characters, a truncation error occurs when the bulk file uploader uploads the data.

Data Vault maps wide characters to the CHAR and VARCHAR types. If the precison of WCHAR or WVARCHAR column is N, the internal precison for CHAR or VARCHAR columns is 2N. The maximum length of WCHAR and WVARCHAR column data that you can load to Data Vault is 16384.

Sample Master XML File

The following text is a sample version of the `metadata.xml file`:

```
<?xml version="1.0" encoding="utf-8"?>
<IDVSERVICE_DOCUMENT VERSION="1.0">
   <SERVICE_CALL>
      <JOB_TYPE>ARCHIVE</JOB_TYPE>
      <SOURCE>XML </SOURCE>
   </SERVICE_CALL>
   <GLOBAL_PARAM>
      <KEEP_DATA>1</KEEP_DATA>
      <COLUMN_SEPARATOR>@#@</COLUMN_SEPARATOR>
      <ROW_SEPARATOR>\n</ROW_SEPARATOR>
      <CRYPTO_KEY>12DF</CRYPTO_KEY>
   </GLOBAL_PARAM>
   <TABLES >
      <TABLE NAME="TAB1" DATABASE="DB1" SCHEMA="SCHEMA1">
      <KEEP_DATA>1</KEEP_DATA>
      <COLUMN_SEPARATOR>@#</COLUMN_SEPARATOR>
      <ROW_SEPARATOR>\n</ROW_SEPARATOR>
      <NULL_INDICATOR>null</NULL_INDICATOR>
      <FILE_PATH>C:\IDV\DATA\source*.csv</FILE_PATH>
        <SCT_PATH>C:\IDV\SCT</SCT_PATH>
           <COLUMNS>
              <COLUMN NAME="COL11">
                 <TYPE>DECIMAL</TYPE>
                 <PRECISION>11</PRECISION>
                 <SCALE>4</SCALE>
                 <NULLABLE>1</NULLABLE>
              </COLUMN>
              <COLUMN NAME="COL12">
                 <TYPE>VARCHAR</TYPE>
                 <PRECISION>310</PRECISION>
                 <SCALE>-1</SCALE>
                 <NULLABLE>1</NULLABLE>
              </COLUMN>
              <COLUMN NAME="COL13">
                 <TYPE>INT</TYPE>
                 <PRECISION>15</PRECISION>
                 <SCALE>5</SCALE>
                 <NULLABLE>1</NULLABLE>
              </COLUMN>
              <COLUMN NAME="COL14">
                 <TYPE>FLOAT</TYPE>
                 <PRECISION>26</PRECISION>
                 <SCALE>0</SCALE>
```

```
            <NULLABLE>1</NULLABLE>
         </COLUMN>
         <COLUMN NAME="COL15">
            <TYPE>TIMESTAMP</TYPE>
            <PRECISION>0</PRECISION>
            <SCALE>0</SCALE>
            <NULLABLE>1</NULLABLE>
         </COLUMN>
         <COLUMN NAME="COL16">
            <TYPE>DOUBLE</TYPE>
            <PRECISION>0</PRECISION>
            <SCALE>0</SCALE>
            <NULLABLE>1</NULLABLE>
         </COLUMN>
         <COLUMN NAME="COL17">
            <TYPE>DATE</TYPE>
            <PRECISION>0</PRECISION>
            <SCALE>0</SCALE>
            <NULLABLE>1</NULLABLE>
         </COLUMN>
   </COLUMNS>
</TABLE>
<TABLE NAME="TAB2" DATABASE="DB1" SCHEMA="SCHEMA1">
<KEEP_DATA>1</KEEP_DATA>
<COLUMN_SEPARATOR>|</COLUMN_SEPARATOR>
<ROW_SEPARATOR>|\n</ROW_SEPARATOR>
<NULL_INDICATOR>NULL</NULL_INDICATOR>
<FILE_PATH>C:\DATA\XML\data*.txt</FILE_PATH>
  <SCT_PATH>C:\DATA\SCT</SCT_PATH>
     <COLUMNS>
        <COLUMN NAME="COL21">
           <TYPE>WVARCHAR</TYPE>
           <PRECISION>100</PRECISION>
           <SCALE>0</SCALE>
           <NULLABLE>1</NULLABLE>
        </COLUMN>
        <COLUMN NAME="COL22">
           <TYPE>VARCHAR</TYPE>
           <PRECISION>310</PRECISION>
           <SCALE>-1</SCALE>
           <NULLABLE>1</NULLABLE>
        </COLUMN>
        <COLUMN NAME="COL23">
           <TYPE>INT</TYPE>
           <PRECISION>15</PRECISION>
           <SCALE>5</SCALE>
           <NULLABLE>1</NULLABLE>
        </COLUMN>
     </COLUMNS>
 </TABLE>
<TABLE NAME="TAB1" DATABASE="DB2" SCHEMA="SCHEMA2">
<KEEP_DATA>1</KEEP_DATA>
<COLUMN_SEPARATOR>|</COLUMN_SEPARATOR>
<ROW_SEPARATOR>|\n</ROW_SEPARATOR>
<NULL_INDICATOR>NULL</NULL_INDICATOR>
<FILE_PATH>C:\BACK\file*.bcp</FILE_PATH>
  <SCT_PATH>C:\BACK\SCT</SCT_PATH>
     <COLUMNS>
        <COLUMN NAME="COL31">
           <TYPE>INT</TYPE>
           <PRECISION>0</PRECISION>
           <SCALE>0</SCALE>
           <NULLABLE>1</NULLABLE>
        </COLUMN>
        <COLUMN NAME="COL32">
           <TYPE>WCHAR</TYPE>
           <PRECISION>300</PRECISION>
           <SCALE>0</SCALE>
           <NULLABLE>1</NULLABLE>
        </COLUMN>
```

```
            <COLUMN NAME="COL33">
               <TYPE>SMALLINT</TYPE>
               <PRECISION>0</PRECISION>
               <SCALE>0</SCALE>
               <NULLABLE>1</NULLABLE>
            </COLUMN>
         </COLUMNS>
      </TABLE>
   </TABLES>
</IDVSERVICE_DOCUMENT>
```

The following text is a sample version of the `metadata.xml` file with table name and ODBC source connection details:

```
<?xml version="1.0" encoding="utf-8"?>
<IDVSERVICE_DOCUMENT VERSION="1.0">
   <SERVICE_CALL>
      <JOB_TYPE>ARCHIVE</JOB_TYPE>
      <SOURCE>ODBC</SOURCE>
   </SERVICE_CALL>
   <GLOBAL_PARAM>
      <KEEP_DATA>1</KEEP_DATA>
      <COLUMN_SEPARATOR>,</COLUMN_SEPARATOR>
      <ROW_SEPARATOR>,\r\n</ROW_SEPARATOR>
   </GLOBAL_PARAM>
   <TABLES>
      <TABLE NAME="tab1" DATABASE="gp" SCHEMA="sch1">
        <FILE_PATH>/home/phegde/gyana/abc*.csv</FILE_PATH>
        <SCT_PATH>/home/phegde/gyana/</SCT_PATH>
        <DSN>RAINDSN</DSN>
        <DSNUID>SA</DSNUID>
        <DSNPWD>SA</DSNPWD>
      </TABLE>
   </TABLES>
</IDVSERVICE_DOCUMENT>
```

Step 3. Start Ssaservice

After you populate the master XML file, start ssaservice.

In Linux operating systems, you can start ssaservice with the script "ssaserviced." The start command is "ssaserviced start." You can use "ssaserviced" to stop the service, but Informatica recommends that you use ssamon to stop ssaservice.

In Windows operating systems, you can start ssaservice with the script "start ssaservice start." Alternatively you can start the service in terminal, with the command "ssaservice start."

Step 4. Submit the Source Data File for the Archive Job

The FILE_PATH value in the master XML file specifies the directory where the flat file or files is located. The name pattern specifies which flat files are candidate files. Ssaservice continuously checks this directory for candidate flat files. You can either create the data files in this directory or copy the flat files into this directory.

After you create the signal file `<BCP_NAME>.ok`, ssaservice picks up the data file for processing. For example, if the naming pattern for table TAB1 is `data*.csv`, the flat file `data1.csv` is a candidate file for TAB1. You must create the signal file as `data1.csv.ok`.

If the naming pattern for table TAB1 is `test*.txt`, the flat files `test1.txt`, `test22.txt`, and `test54.txt` are candidate files for TAB1. You must create the signal files as `test1.txt.ok`, `test22.txt.ok`, and`test54.txt`.

When you schedule the first load job, the executor checks whether or not the table exists. If the table exists, the load job continues. If the table is not found in Data Vault, the uploader creates and runs the DDL before the start of the loader process.

If the table attributes do not match with the table definition in the metadata, the bulk file uploader tries to alter the table with the latest metadata information from the master XML file. The bulk file uploader only attempts to alter the table if the newer column attribute is bigger in magnitude than the older type.

Step 5. Monitor the Job Status

Use the ssamon utility to return the status of a job based on the job ID, list of job IDS, or range. Reporting is allowed based on completion date.

DDL Creation and Limited Alter Table Support

The bulk file uploader can create the table DDL if the table does not exist in Data Vault. The uploader also has a limited ability to alter tables.

When you schedule a job to load data into a table, the bulk file uploader creates the DDL, if the table does not exist in Data Vault. If the table exists, the job compares the table attributes in the master XML file with table attributes in the Data Vault metadata.

If you need to add new columns to an existing table, add the column details at the end of the column list in the XML file. The uploader alters the table to add the column.

If the data type in the master XML file is different from the Data Vault metadata and there is no possibility of loss of precision, the uploader alters the existing table attributes to the values in the master XML file. For example, the type CHAR(100) can be altered to CHAR(200), but not the reverse. You cannot alter the column to a different type.

Discovery and Compliance from Data Archive

The bulk file uploader does not migrate compliance information from the source database. If you want to use Data Discovery features, Browse Data, keyword search, or compliance features, you must create discovery entities in the Enterprise Data Manager after the data has been loaded to tables in Data Vault. Then, create a target connection in Data Archive and run a dummy retirement project.

Complete the tasks below to use Data Discovery and apply compliance features, for example retention and legal hold, on the data archived by the bulk file uploader.

In the Enterprise Data Manager:

1. Create a new application version.
2. Create a Data Vault connection.
3. Import metadata from the archived tables, using Data Vault as the source.
4. Create a discovery entity using the tables in Data Vault.

In the Data Archive user interface:

1. Create a target connection using the same Data Vault folder where the data resides.

2. Create a dummy source connection. The source connection can be anything. Data Archive uses the dummy source connection to create the metadata information in the AM_HOME tables. Data Archive also uses this source connection in UI operations, for example Browse Data or keyword search.
3. Run a dummy retirement project. Data Archive uses the dummy retirement project in the same manner that it uses the dummy source connection.
4. Update the nucleus.ini file in the Data Vault Data Archive plugin directory with the Data Vault connection details.

To view Techview data, enable the "informia.dataDiscoveryIgnoreJobIdQuery=Y" parameter in the `conf.properties` file.

Data Type Mapping

The following table describes how the bulk file uploader converts source data types to Data Vault data types:

Source Data Type	Data Vault Data Type
CHAR	CHAR
WCHAR	CHAR
VARCHAR	VARCHAR
WVARCHAR	VARCHAR
REAL	REAL
FLOAT	DOUBLE
DOUBLE/ DOUBLE PRECISION	DOUBLE
DECIMAL	DECIMAL
NUMERIC	DECIMAL
TINYINT	SMALLINT
SMALLINT	SMALLINT
INTEGER/INT	INTEGER
DATE	DATE
TIME	TIME
TIMESTAMP	TIMESTAMP
DATETIME	TIMESTAMP
SMALLDATETIME	TIMESTAMP

BINARY, VARBINARY, CLOB, and BLOB data types are not supported by the bulk file uploader.

System Requirements

The following 32-bit libraries are required for Linux operating systems: libxml, ncurses, libstdc++

For example, libxml2-2.7.6-1.el6.i686, ncurses-libs.i686, libstdc++.i686

Known Limitations

The bulk file uploader has the following limitations:

1. The bulk file uploader does not support all unicode multibyte characters in Data Vault object names.
2. Rainstor is the only source supported for fetching metadata information through an ODBC connection.
3. The bulk file uploader does not migrate table constraints and compliance information from the source database. You can manually create table constraints and apply compliance features after the table has been created.

CHAPTER 12

Data Vault Administration Tool

This chapter includes the following topics:

- Data Vault Administration Tool Overview, 159
- Using the Data Vault Administration Tool, 160
- Managing the Data Vault Repository, 162
- Managing the Data Vault Clients, 168
- Managing the Data Vault Agents, 170
- Viewing Administrator Account Information, 172
- Running a Command Line Program, 173
- Shutting Down the Data Vault Service, 173

Data Vault Administration Tool Overview

The Data Vault Administration Tool is a client application for managing and monitoring the Data Vault repository and the Data Vault components. From the Administration Tool, connect to the Data Vault to view and check the metadata in the Data Vault repository and to manage the Data Vault Agents, administrators, and clients.

You can use the Administration Tool to perform the following tasks:

- View the connection information for the Data Vault Service.
- Check the integrity of the metadata in the Data Vault repository.
- Check the integrity of the Data Vault tables and data files.
- View information about the Data Vault repository and the Data Vault tables and data files
- View information about the Data Vault Agents, clients, and administrator user accounts.
- Stop the Data Vault Agent processes.
- Shut down the Data Vault Service.
- Close client connections.
- Open a console window and run commands to cancel or interrupt queries.

The Administration Tool runs on Windows.

Using the Data Vault Administration Tool

You can install the Data Vault Administrator Tool on Windows to manage the Data Vault repository and the Data Vault components.

The Administration Tool installation is a separate option in the Data Vault installer. Select the **Data Vault Administration Tool and SQL Tool** option when you run the Data Vault installer.

Starting the Data Vault Administration Tool

When you install the Data Vault Administration Tool, the installer adds a shortcut to the application from the Start menu. You must log in to connect to the Data Vault Service.

1. On the Windows start menu, go to **Programs** > **Informatica Data Vault** > **FAS Administrator**.

   The Administration Tool displays a login window. You can ignore the error message that indicates that the installation path is not available.

2. On the login window, enter the login information.

   The following table describes the parameters required to log in to the Administration Tool:

Parameter	Description
User	User name for the user account to connect to the Data Vault Service. You can use the administrator user account created during the Data Vault installation. The user name of the administrator account is *dba*.
Password	Password for the user account to connect to the Data Vault Service.
Host	Host name or IP address of the machine that hosts the Data Vault.
EXE Port	Port number used by the Data Vault Agent and the Data Vault Administration Tool to connect to the Data Vault. Default is 8600.

3. Click **Connect** to log in to the Administration Tool and connect to the Data Vault Service.

Shutting Down the Data Vault Administration Tool

To shut down the Data Vault Administration Tool, click **Main** > **Exit**. The Administration Tool disconnects from the Data Vault Service when it shuts down.

To disconnect from the Data Vault Service without shutting down the Administration Tool, click **Main** > **Disconnect**.

Data Vault Administration Tool Display

The Data Vault Administration Tool user interface has a section for navigation and a section for information display.

The Data Vault Administration Tool has the following sections:

Navigation Pane

Contains a navigation tree where you can navigate to different nodes to view information on different objects.

Information Pane

Displays information for the selected node.

Data Vault Administration Tool Settings

You can set the display behavior of the Data Vault Administration Tool. To set the display behavior, click **Main** > **Settings** and specify the settings to use.

The following table describes the display settings you can specify for the Data Vault Administration Tool:

Option	Description
Auto refresh	Refreshes the information at an interval that you can specify. The default refresh interval is 5 seconds. Default is False.
Branch update	Updates the current node and the parent nodes of the current node. When you modify a node or an object in a node, the Data Vault Administration Tool refreshes the contents of the node and the contents of all the parent nodes in the hierarchy. For example, if you modify the priority of an agent, the Data Vault Administration Tool refreshes the content of the Agent node, the Agent List node, and the Agents node. Default is True.
Scaled values	Displays numeric information in abbreviated rounded format or in units of measurement that can differ from the default in the Data Vault repository. For example, if the Scaled Values option is true, the Data Vault Administration Tool displays 16,685 B as 16.7 KB and 37,001 B as 37.0 KB. Default is True.

Toolbar

The Data Vault Administration Tool provides a toolbar with buttons you can use to perform a task. Instead of selecting from the menu, you can click a button on the toolbar.

To display the toolbar, click **View** > **Toolbar**. You can toggle the toolbar option on or off.

The following table describes the buttons in the Data Vault Administration Tool toolbar:

Option	Description
Connect	Displays the Data Vault Administration Tool login window. Enter the Data Vault Service connection and user account information to connect to the Data Vault Service that you want to manage.
Disconnect	Disconnects from the Data Vault Service.
Console	Opens a console window. You can run commands from the Data Vault Service command line program. To close the console window, click **Console** on the toolbar again or click **Close** on the console window.
Log	Opens a log window that displays the log entries generated by tasks performed in the Data Vault Administration Tool. To close the log window, click **Log** on the toolbar again or click **Close** on the log window.

Option	Description
Refresh	Refreshes the contents of the selected node in the navigation pane. If the **Branch Update** option in the Data Vault Administration Tool settings window is set to True, this button refreshes the contents of the selected node and the parent nodes.
Auto refresh	Enables or disables the automatic refresh setting of the Data Vault Administration Tool display. You can set the refresh frequency in the Data Vault Administration Tool settings window.

Keyboard Shortcuts

You can use keyboard shortcuts to start processes or open windows in the Data Vault Administration Tool.

The following table describes the keyboard shortcuts you can use in the Data Vault Administration Tool:

Function Key	Description
F5	Refreshes the information displayed on the Data Vault Administration Tool.
F7	Opens the Settings window.
F9	Opens the Console window.
F10	Opens the Log window.

You can also use the Alt key with the first letter of the menu item to display the options for the menu. For example, press Alt+A to display the options for the Action menu. The Action menu can have a different list of options based on the node selected in the navigation pane.

Managing the Data Vault Repository

The Data Vault repository stores information about the data archived in the Data Vault.

The Data Vault organizes archived data in databases and tables. A database is equivalent to the Data Vault archive folder in the Data Vault. The database and table names correspond to the database and table names in the archive source. The Data Vault stores the archived data and metadata in data files associated with a table.

You can view information about the databases and tables and their associated data files. You can verify that the data and metadata in the Data Vault are valid and do not contain inconsistencies such as data files not associated with a valid archived table.

Checking Metadata Integrity

You can check the validity and integrity of the metadata in the Data Vault and the Data Vault repository. Open a log window to view the diagnostic messages generated by the metadata integrity check.

You can verify metadata integrity at different levels. On the navigation pane, select the node for the metadata level that you want to check.

The validation process can take a while. The length of time required depends on the metadata level at which you perform the check.

The following table describes the action that you perform to validate the data files and metadata at different levels:

Level	Action
All databases	Select the **Databases** node in the navigation pane and click **Action** > **Check**.
Database	Select a database in the **Databases** node of the navigation pane and click **Action** > **Check**.
Table	Select a table in a database in the navigation pane and click **Action** > **Check**.
Data files in a table	Select the **Files** node for a table in the navigation pane and click **Action** > **Check**.
File	Select a file in the **Files** node of a table in the navigation pane and click **Action** > **Check**.

After the Data Vault Administration Tool runs a check on the selected object, it displays a message confirming whether the metadata integrity check completed successfully or not. The Data Vault Administration Tool sends diagnostic messages to the logs. To view the results of the metadata integrity check, open a log window and review the log messages.

Viewing Database Information

You can view information about an archive database such as the list of tables in the database and their definitions and associated data files. You can drill down to view more detailed information about the data files.

The Data Vault Service uses the decimal kilobyte when it performs data size calculations. The decimal kilobyte is equivalent to 1000 bytes. The Data Vault Administration Tool does not use the binary kilobyte, which is equivalent to 1024 bytes, in the calculations.

Viewing Databases

A database is equivalent to a Data Vault folder and contains the tables with the archived data files. You can view the databases and tables of archived data in the Data Vault repository.

To view the list of databases in the Data Vault repository, select the **Databases** node in the navigation pane. The information pane displays the list of databases.

The following table describes the information displayed for the databases:

Property	Description
ID	Database identifier assigned by the Data Vault Service.
DB	Name of a database in the Data Vault repository. A database is equivalent to a Data Vault archive folder. The database or Data Vault archive folder name is based on the database name in the archive source.

Data Vault Tables

A database or Data Vault archive folder contains Data Vault tables. The Data Vault tables are classified based on the type of data files they contain.

A database can contain the following types of tables:

Updatable

Updatable tables contain data with associated metafield columns. When you archive data and add additional information such as tags and legal holds, the Data Vault Service adds metafields to store the information. The Data Vault Service does not update data columns but can update metafields columns.

Each partition in an updatable table contains the following types of files:

- Archive file. Contains the archived data. The archive file name has a suffix of *_a*.
- Update file. Contains the metadata for the archived data. The Data Vault Service modifies this file when you set tags or legal holds on the archived data or when you delete a logical entity. The file name has a suffix of *_u*.

Archive

Archive tables contain data that do not have associated metafields. When you archive data without adding tag or other metadata, the Data Vault Service does not add metafields to the table. A partition in an archive table contains archive files but does not contain update files.

Viewing Tables

To view the list of tables in a database, select the **Tables** node for a database in the **Databases** node. The information pane displays the list of tables for the database.

The following table describes the information displayed for the list of tables:

Property	Description
ID	Table identifier assigned by the Data Vault Service.
Schema Name	Name of the database schema associated with the table.
Table Name	Name of the listed table.
Type	Type of table based on the data files that the table contains. A table can be one of the following types: - Archive table - Updatable table

Viewing Table Statistics

You can view the archive statistics for a table, including the number of partitions and the compression ratio.

To view the archive statistics for a table, select the name of a table in the **Tables** node. The information pane displays the archive statistics for the selected table.

The following table describes the summary information that is displayed for a table:

Property	Description
Table	Database, schema, and table name of the table in the format "DatabaseName"."SchemaName"."TableName".
ID	Table identifier assigned by the Data Vault Service.
DB	Name of the database that contains the table. The database name in the Data Vault repository is the same as the database name in the archive source.
Schema Name	Name of the schema associated with the database that contains the table.
Table Name	Name of the selected table.
Type	Type of table based on the data files the table contains. A table can be one of the following types: - Archive table - Updatable table

The Data Vault Service displays data file information for Updatable tables but not for Archive tables. For example, the Data Vault Service displays the size of the data files for an Updatable table but not for an Archive table

The following table describes the detailed statistics displayed for a table:

Property	Description
ID	Table identifier assigned by the Data Vault Service.
Name	Schema and table name of the table in the format "SchemaName"."TableName".
Type	Type of table based on the data files the table contains. A table can be one of the following types: - Archive table - Updatable table
Columns	Number of columns in the selected table.
System Columns	Number of system columns in the selected table. The Data Vault Service uses system columns internally for storing metadata. This property does not display for a table of type Archive.
Row Length	Maximum length that a row can have in the selected table. The length of the row is based on the size of the columns in the table. Displayed in bytes.
Partitions	Number of partitions in the table.
Files	Number of data files registered for the selected table.
Total size (Archive)	Size of the archive data portion of the compressed data in the table. Archive files contain the data portion of the table. This property does not display for a table of type Archive.

Property	Description
Total size (Update)	Size of the metadata portion of the compressed data in the table. Update files contain the metadata portion of the table. The Data Vault Service adds to the update files when you perform Data Archive tasks such as setting up legal holds. This property does not display for a table of type Archive.
Total size (Arc/Upd)	Size of archive and update data files as a percentage of the total table size. This property does not display for a table of type Archive.
Total size	Total size of the compressed data in the selected table.
Total rows	Total number of rows in the selected table.
Flat file size (Archive)	Size of the archive data portion of the uncompressed BCP data. The Data Vault Service uses the bulk copy program (BCP) utility to transfer uncompressed source data to the optimized file archive. This property does not display for a table of type Archive.
Flat file size (Update)	Size of the metadata portion of the uncompressed BCP data. This property does not display for a table of type Archive.
Flat file size (Arc/Upd)	Size of the data and metadata portion as a percentage of the total size of the uncompressed BCP data. This property does not display for a table of type Archive.
Flat file size	Size of the uncompressed BCP data for the selected table.
Compression ratio	Estimated degree of data compression in the selected table.

Viewing the Table Definition

You can view the table definition of a selected table.

To view the table definition for a table, select the **Definition** node for a table in the navigation pane. The information pane displays the structure of the table.

The following table describes the information displayed for the columns in the table:

Property	Description
Name	Name of the column in the table.
DataType	Datatype of the column in the table.
Precision	Precision of the column in the table.
Scale	Scale of the column in the table.
Nullability	Indicates whether the column have a Null value.

Viewing the Data Files in a Table

You can view the data files associated with a table.

To view the data files associated with a table, select the **Files** node for a table in the navigation pane. The information pane displays the data files associated with the table.

The following table describes the information displayed for the data files:

Property	Description
TableID	ID of the table to which the Data Vault data file belongs.
v_ID	File identifier generated and used internally by the Data Vault Service.
r_ID	File identifier generated and used internally by the Data Vault Service.
Type	Table type generated and used internally by the Data Vault Service. Default is SCT.
Name	Path and file name of the Data Vault data file.
Active	Status flag generated and used internally by the Data Vault Service. Default is Active.
CustomData	Custom data included in the Data Vault data file, such as legal hold tags.
Rows	Number of table rows included in the Data Vault data file.
Size	Size of the Data Vault data file.

Viewing Data File Information

You can view a list of all the data files associated with a table. A data file can be an archive file or an update file. The Administrator tool displays the same information for an archive or update file.

To view a data file associated with a table, select one of the files listed under the **Files** node for a table in the navigation pane.

The following table describes the information displayed for the archive and update files:

Property	Description
Name	Column name in the data file.
Domain	Name of the domain associated with the column. The domain is the list of unique values that the column contains.
Type	Datatype of the column
Precision	Precision of the column.
Scale	Scale of the column.
Nullability	Indicates whether the column can have a NULL value.

Property	Description
Min	Lowest value that the column contains. For text values, the Min value is the lowest value in alphabetic order.
Max	Highest value that the column contains. For text values, the Max value is the highest value in alphabetic order.

Managing the Data Vault Clients

You can view information about the clients that connect to the Data Vault.

You can also view information about the tasks associated with queries that are sent by the clients. When the Data Vault Service receives a query, it generates a task and a task identifier. The Data Vault Service can generate multiple tasks for a query. The number of tasks associated with a query is equivalent to the number of data files that are accessed by the query.

Viewing Clients

To view the list of clients connected to the Data Vault, select the **Clients** node in the navigation pane. The information pane displays the list of all clients that are connected to the Data Vault and information about the queries that are sent by the client.

The following table describes the information displayed for the clients:

Property	Description
ID	Client identifier that is assigned by the Data Vault Service.
User Name	User name of the client user account connected to the Data Vault.
Start Time	Time when the Data Vault Service started processing the last query received from the client.
Running Time	Total query processing time for the last query received from the client.
Queries	Number of queries from the client processed by the Data Vault Service since the client connected to the Data Vault.
Active Queries	Number of queries from the client that the Data Vault Service is currently running.
Rows	Number of rows fetched from the Data Vault since the client connected to the Data Vault.
Volume	Amount of data fetched from the Data Vault since the client connected to the Data Vault. Displayed in bytes.

Viewing Queries

You can view a list of the queries received from clients connected to the Data Vault.

To view the list of queries received from clients, select the **All Queries** node under the **Clients** node in the navigation pane. The information pane displays the list of all queries received from clients.

The following table describes the information displayed for the list of queries:

Property	Description
ID	Query identifier assigned by the Data Vault Service.
Client ID	ID of the client that sent the query request.
Start Time	Date and time when the Data Vault Service received the query.
Running Time	Amount of time it took to process the query.
Query Type	Type of SQL operation that is performed by the query. The query can have the following types of operation: - SELECT - INSERT - UPDATE - DELETE
Source Type	Type of data queried. The source type can be one of the following types: - Compressed table in the Data Vault (ARC) - Data Vault repository (MDB)
SQL	SQL statement in the query.
Status	Status of the query. The query can have one of the following statuses: - PREPARED. The Data Vault Service is preparing the query. - EXECUTING. The Data Vault Service is currently running the query. - FETCHING. The Data Vault Service has completed running the query and is currently sending the query results to the client. If the client does not close the session, the query remains in FETCHING status. - FAILED. The Data Vault Service failed to complete the query. - TERMINATED. The Data Vault Service terminated the query before the query was completed. - UNKNOWN. Status information for the query could not be retrieved.
Tasks [T/S/C]	Number of tasks for the query. The number of task for a query is equivalent to the number of partition data files that the query must access. For example if the Data Vault Service must access 20 different partition data files to complete a query, the Data Vault Service creates 20 tasks for the query. The Data Vault Administration Tool displays the task information in the following format: *total tasks -> started tasks -> completed tasks* *Total tasks* is the total number of tasks that a query must perform. The total number of tasks include tasks that are not started, tasks that have started, and tasks that are complete. *Started tasks* is the total number of tasks started but not yet completed by the query. *Completed tasks* is the total number of tasks completed by the query.
Rows	Number of rows fetched by the query from the Data Vault.
Volume	Amount of data fetched by the query from the Data Vault.

Closing the Client Connections

You can close the connection for one client or for all clients connected to the Data Vault.

To close the connection of one client, select a client under the **Clients** node in the navigation pane. Then click **Action** > **Close**.

To close the connection of all clients connected to the Data Vault, select the **Clients** node in the navigation pane and then click **Action** > **Close All**.

Managing the Data Vault Agents

You can view information about the Data Vault Agents that are configured for the Data Vault. You can also view information about the tasks associated with the queries that are processed by an agent.

Viewing the Data Vault Agents

To view the list of Data Vault Agents that are configured in the Data Vault, select the **Agent List** node under the **Agents** node in the navigation pane. The information pane displays the list of Data Vault Agents.

The following table describes the information displayed for the Data Vault Agents:

Property	Description
ID	Agent identifier assigned by the Data Vault Service.
Host	Host name of the machine that hosts the agent.
Priority	Priority of the Data Vault Agent. A lower number indicates a higher priority. A priority of zero (0) indicates that the agent is not enabled.
Task ID	ID of the task currently performed by the agent, if the agent is processing a request and the State property shows a status of Running.
State	Status of the Data Vault Agent. The agent can have one of the following statuses: - Ready. The agent is not currently processing a request. The load balancer can assign this agent to process a new request. - Running. The agent is currently busy processing a request.
Start Time	Date and time when the Data Vault Agent was started.

Viewing the Agent Hosts

To view the list of machines that host Data Vault Agents, select the **Hosts** node under the **Agents** node in the navigation pane. The information pane displays the machines that host Data Vault Agents.

The following table describes the information displayed for the agent hosts:

Property	Description
Host	Host name of the machine that hosts the agent.
Agents	Number of agents running on the host.
Frozen Agents	Number of agents that are not enabled. An agent is not enabled when the priority level is set to 0.
Running Tasks	Number of tasks that the agents on the host are currently running.

Property	Description
Start Time	Date and time when the agent host was added to the Data Vault Service.
Idle Time	Amount of time that the host machine has been inactive. An inactive host is a host where no agent is processing a request. When an agent running on the host processes a request, the idle time is greater than zero.
Running Time	Amount of time that the host machine has been active. An active host is a host where one or more agents are processing requests.

Viewing Tasks

To view the list of tasks that the Data Vault Agents on the host are running, select the **Tasks** node under the **Agents** node in the navigation pane. The information pane displays the tasks for all Data Vault Agents on the host.

The following table describes the information displayed for tasks:

Property	Description
Type	The type of task. Tasks can have one of the following types: - Query. Task generated for a query from a client. - Admin. Task generated for an internal process.
Query ID	Query identifier for the query associated with the task.
File ID	File identifier for the data file that the task must access.
Agent ID	Agent identifier for the agent to which the task is assigned.
Command	The command used to run the task.
State	The status or agent information for the task. The task can have one of the following states: - Wait. The task is in the queue ready to run. - Host name and agent ID of the agent that is running the task.
Start Time	Date and time when the task started running.
Wait Time	Amount of time the task has been in the queue. Displays if the task in Wait state.
Running Time	Amount of time that the task has been running. Displays if the task is not in the Wait state.
Rows	Number of rows fetched for the query since the task started.
Volume	Amount of data fetched for the query since the task started. Displayed in bytes.
Pref Hosts	Preferred host for task execution. The Data Vault Service assigns the task to the agents on the preferred host before other agents. If all agents on the preferred host are busy, the Data Vault Service assigns the task to agents on another host.

Setting the Agent Priority Level

You can use the Data Vault Administration Tool to enable and set the priority level of the Data Vault Agents.

1. On the navigation pane, go to the **Agent List** node and select the agent for which you want to set the priority.
2. Set the priority for the agent.
   - To set a priority number from one to ten, click **Action** > **Priority** and select the priority number that you want to set for the agent. Set the priority to zero to disable the agent from processing client requests.
   - To set a priority number greater than ten, click **Action** > **Priority** > **Set**. On the **Agent Priority** window, select **Enable Agent** and enter the priority number that you want to set for the agent. Click **OK**.

   The Data Vault Administration Tool displays a message to confirm that the priority level is changed.
3. Click **OK**.

   The information pane displays the new agent priority level.

Shutting Down All Data Vault Agents

You can use the Data Vault Administration Tool to shut down all agents running in the Data Vault Service.

1. On the navigation pane, select the Agents node.
2. Click **Action** > **Stop all agents**.

   The Data Vault Administration Tool displays a message requesting confirmation that you want to shut down all agents.
3. To get more information about the agents to shut down, click **Details**.

   The Data Vault Administration Tool displays the list of agents with the agent ID and host name.
4. Review the list of agents and verify that you want to shut down all the agents in the list and click **Yes**.

Shutting Down a Data Vault Agent

You can use the Data Vault Administration Tool to shut down a Data Vault Agent.

1. On the navigation pane, go to the **Agent List** node and select the agent that you want to shut down.
2. Click **Action** > **Stop**.

   The Data Vault Administration Tool displays a message to confirm that you want to shut down the agent.
3. Click **Yes** to shut down the agent.

Viewing Administrator Account Information

You can view information about the administrator user accounts that are connected to the Data Vault Service.

To view information about the administrator user accounts connected to the Data Vault Service, select the **Administrators** node in the navigation pane. The information pane displays the list of administrator user accounts connected to the Data Vault Service.

The following table describes the information displayed for an administrator user account:

Property	Description
ID	Windows process ID for the Data Vault Administration Tool that is used by the administrator user account.
Host	Host name of the machine that hosts the Data Vault Administration Tool.
Connect ID	Connection identifier that is assigned by the Data Vault Service to the user account connection.
User	Name of the user account logged in to the Data Vault Administration Tool.
Privilege	Level of privileges assigned to the user account.
Start Time	Date and time when the Data Vault Administration Tool that is used by the administrator user account was started.

Running a Command Line Program

You can use the Data Vault Administration Tool to run commands available in the Data Vault Service command line program. Open a console window to run a command.

1. Click **Main** > **Console** or press F9 to open a console window.
2. On the entry box at the bottom of the console, type the Data Vault Service administration command that you want to run.

   Include any parameter required by the command.
3. Click **Send**.

   The output pane displays the results of the command and any messages that are generated by the command. Use the scroll bar to view text beyond the output pane. You can click **Clear** to delete all text from the output pane.
4. Click **Close** to close the Console window.

Shutting Down the Data Vault Service

You can use the Data Vault Administration Tool to shut down the Data Vault Service to which the Data Vault Administration Tool is connected.

You cannot restart the Data Vault Service from the Data Vault Administration Tool. If you shut down the Data Vault Service from the Data Vault Administration Tool, you must restart it from the command line.

When you shut down the Data Vault Service, the Data Vault Administration Tool continues to run. You cannot reconnect to the Data Vault until you restart the Data Vault Service from the command line.

1. Click **Action** > **Shutdown** .
2. Confirm that you want to shut down the Data Vault Service.

CHAPTER 13

Data Vault Logs

This chapter includes the following topics:

- Overview of the Data Vault Logs, 174
- Standard Log, 174
- Trace Log, 175
- Query Statistics Log, 176

Overview of the Data Vault Logs

When the Data Vault components run, the components generate logs. View the logs to monitor the component processes.

The Data Vault logs contain load balancer warnings, errors, query statistics, and other diagnostic information that can be analyzed for administrative purposes, or to aid in troubleshooting.

The Data Vault logs include the following types of logs:

- Standard log
- Trace log
- Query statistics log

Standard Log

The standard log is a text file where diagnostic messages from the Data Vault system are written.

The type of information written to the standard log file depends on the specified level of detail. At the most basic level, the log file contains all error messages and some other important information, such as the current configuration parameters. At the next highest level, in addition to the basic details, the log file contains some extra information returned from the server.

Finally, at the highest level of detail, system trace information is written to the log file, in addition to the previously mentioned information. This trace information includes the current section of code being executed, the current system layer, parameter values passed to functions, and other similar information.

Because the log file can grow quite large, especially at higher levels of logging detail, there are configurable options for limiting the growth of the log file, on the basis of either file size (the default) or time period. If both options are set, the file size takes precedence. Once the set limit is reached, a new log file is created to

store the latest messages. The old log file is not removed, at least not right away. Since there can be one or more older log files left over, there is an option for specifying the total number of log files to retain, which automates the cleanup of old log files.

The following table describes the parameters related to the standard log file. They are all set in the [SERVER] section of the ssa.ini file:

Parameter	Description
LOG	Enables or disables standard logging.
LOGVERBOSE	The level of detail for log messages.
LOGLIMIT_FILESIZE	The maximum file size for the log file in bytes.
LOGLIMIT_TIME	The time limit for the log file, in number of minutes, hours, days, or months.
LOGLIMIT_MAXFILES	The total number of log files to retain.

A standard log file, located in the path specified by the LOGDIR parameter, has the following naming scheme:

`ssa-yyyy-mm-dd-HH-MM-SS.log`

where:

- *yyyy* is the year
- *mm* is the month
- *dd* is the day
- *HH* is the hour
- *MM* is the minutes
- *SS* is the seconds

Trace Log

The trace log is a text file where diagnostic messages from the Data Vault are written. You can analyze the trace log to troubleshoot a problem in the Data Vault.

The information contained in a trace log is exactly the same as what would be written to the standard log at the highest level of detail, which includes error messages, configuration parameters, server messages, and internal system trace information. The trace log and the standard log are independent of each other, so they can be enabled at the same time.

As with the standard log, the trace log can be limited by file size or time period, and the number of trace logs to retain can be configured.

By default, the trace log is not enabled.

The following table describes the parameters related to the trace log. They are set in the SERVER section of the ssa.ini file:

Parameter	Description
TRACELOG	Enables or disables trace logging.
TRACELOGLIMIT_FILESIZE	The maximum file size for the trace log file.
TRACELOGLIMIT_TIME	The time limit for the trace log, in number of minutes, hours, days, or months.
TRACELOGLIMIT_MAXFILES	The total number of trace log files to retain.

A trace log file, located in the path specified by the LOGDIR parameter, has the following naming scheme:

```
ssa_trace-yyyy-mm-dd-HH-MM-SS.log
```

where:

- *yyyy* is the year
- *mm* is the month
- *dd* is the day
- *HH* is the hour
- *MM* is the minutes
- *SS* is the seconds

Query Statistics Log

The Data Vault can be configured to export query statistics to CSV (Comma-Separated Values) files, from which the information can be loaded into a database table for analysis and manipulation. The exported statistics include such things as query timings, number of data files processed, and the amount of disk space required for temporary files, recorded for each individual query.

The query statistics log can be enabled or disabled, and the maximum log file size can be set, through the ssa.ini file.

The following table describes the query statistics log parameters that are set in the [SERVER] section:

Parameter	Description
STATLOG	Enables or disables the query statistics log. The possible values are **ON** (enable) and **OFF** (disable). By default, the query statistics log is enabled. When enabled, the query statistics are continuously written to CSV files in the directory location specified by the LOGDIR parameter.
STATLOGLIMIT_FILESIZE	Sets the maximum size of each log file in bytes. When the size limit for the current log file is reached, it is renamed, and the query statistics are written to a new log file. By default, the maximum log file size is **1048576** bytes (1 MB).

Example of a Query Statistics Log

The following example of configuration settings enable the query statistics log and sets a maximum log file size of 10485760 bytes (10 MB).

```
[SERVER]
STATLOG=ON
STATLOGLIMIT_FILESIZE=10485760
...
```

When query statistics logging is enabled, the statistics for each query are written to a CSV log file in the LOGDIR directory, whether or not the query completes successfully. The log file name has the following format:

```
ssa_stats-yyyy-mm-dd-HH-MM-SS.sta
```

where *yyyy-mm-dd-HH-MM-SS* is the timestamp from when the CSV file was created.

The initial timestamp used for the log file name will remain the same for the duration of the Data Vault Service session, even if query statistics logging is disabled and then re-enabled. The timestamp will only change if Data Vault Service is restarted.

The initial timestamp used for the log file name will remain the same for the duration of the Data Vault Service session, even if query statistics logging is disabled and then re-enabled. The timestamp will only change if the Data Vault Service is restarted.

When the maximum file size (set by the STATLOGLIMIT_FILESIZE parameter) is reached, the current log file is renamed and a new log file is created for the latest query statistics. The older log file has a three-digit extension appended to its name, to distinguish it from the currently active log file. The numeric extension starts at ".001" and increments by one for each subsequent log file that reaches its maximum size, up to ".999".

For example:

```
ssa_stats-2008-08-15-11-31-02.001
ssa_stats-2008-08-15-11-31-02.002
ssa_stats-2008-08-15-11-31-02.003
...
ssa_stats-2008-08-15-11-31-02.999
```

The LOGDIR location can therefore contain at most **1000** query statistics log files (999 previous log files, plus one currently active CSV file) for the same session.

Note: Query statistics logs usually exceed the maximum file size by a very small amount, since the last query statistics record in a log file is written in its entirety, rather than split across two log files.

Query Statistics Fields

The first line of each query statistics log file is a header containing the field names for the records that are written to the file.

The following table describes query statistics fields:

Field	Description
TimeStamp	The date and time when the query was started.
DBName	The name of the Data Vault repository targeted by the query.
UserID	The user authorization that executed the query.

Field	Description
QueryID	The internal identifier for the query.
RowsFetched	The total number of rows in the query result set.
RowsReturned	The total number of rows from the query result set that were sent to the client.
TimeGetMD	The amount of time (in seconds, rounded from milliseconds) from when the query fetching starts to when the first data file is selected. Between these two points, a list of all the data files affected by the query is assembled from the metadata information.
TimeExec	The amount of time (in seconds, rounded from milliseconds) from when the first compressed data file is selected to when the first row (or block of rows, depending on the manner of fetching) is received by the client.
TimeFetch	The total amount of time (in seconds, rounded from milliseconds) spent fetching data from compressed data files.
TotTimeGetMD	The total amount of time (in seconds, rounded from milliseconds) from when the query fetching starts to when the last data file is selected.
TotTimeExec	The total amount of time (in seconds, rounded from milliseconds) from when the first data file is selected to when the result set is complete.
NumFiles	The total number of data files accessed by the query.
NumTasks	The total number of processes started during the query execution. The number of processes is equal to the number of data files accessed by the query.
MaxDiskSpace	The maximum amount of disk space (in bytes) required by the query for temporary result set data.
SQL1...SQL9	The text of the SQL command for the query, split across multiple SQL*n* fields if necessary. That is, if the SQL statement exceeds the length of the *SQL1* field (4056 bytes), the first 4056 characters are put in *SQL1* and the rest is put in the *SQL2* field; if the rest of the SQL statement does not fit in *SQL2*, that field is filled and the rest is put in *SQL3*; and so on until the *SQL9* field is filled. If the full SQL statement exceeds 36,504 characters, then the query text will be truncated at the end of the *SQL9* field.
Error	The error code returned, if the query did not complete successfully.
ErrorString	The error message associated with the error code above.

When query statistics logging is enabled, each executed query will have a record written to the CSV log file, whether or not the query completes successfully.

Loading Query Statistics into a Database

The contents of the query statistics logs can be loaded into database tables for subsequent analysis and manipulation.

The following scripts are included with Data Vault Service to facilitate setting up a database to receive the query statistics information:

- ssastat.ddl

- ssastat.ndl

To use the scripts, first connect to the database as the user DBA, and run ssastat.ddl through the Data Vault SQL tool to create the domains and table (SSAQSTAT) for the database in a schema called SSASTAT, owned by the userID SSASTAT. Then, using the Data VaultLoader, execute the ssastat.ndl script to load the query statistics records into the SSASTAT.SSAQSTAT table. The ssastat.ndl file will likely have to be edited to point to the appropriate statistics log (**.sta**) files, since the loader script points to "ssa_stats.sta" in the current directory by default.

Important: When invoking the Data Vault Loader, include the argument **-s 1** to start processing from the second row of data. The first row in every query statistics log consists of column headers.

CHAPTER 14

User Account Privileges

This chapter includes the following topics:

- User Account Privileges Overview, 180
- GRANT, 181
- REVOKE, 185
- ALTER AUTHORIZATION, 187
- DBA Privileges, 188
- Data Vault Passwords, 189
- DBA User Passwords, 191

User Account Privileges Overview

While the user DBA has unrestricted privileges to create, alter, manipulate, and drop any object in the database, non-DBA users must have certain privileges in order to execute SQL statements involving objects in the database. Non-DBA users, once created, can connect to the database and proceed to create new schemas (which they will own), and then tables, domains, and so on within them. A new user also receives ownership privileges on their default schema, if it did not exist prior to the creation of the user authorization; if the default schema is already owned by another user, the new user receives the OWNER privilege on it.

Ownership of a database object entails the ability to give privileges on the object to other users using the GRANT command; these privileges may furthermore be given WITH GRANT OPTION, which allows the grantee to give the privileges to yet another user. Privileges granted to another user can later be taken away using the REVOKE command. As a "super-user", the DBA can GRANT or REVOKE any privilege on any database object, including DBA privileges, which confer upon the recipient the same abilities as the user DBA.

Further details about specific privileges can be found in the following sections, which describe the syntax and usage of the privilege-related SQL commands.

- GRANT

  Grants privileges to a user authorization on the database or database objects such as schemas, tables, and so on.

- REVOKE

  Nullifies specified user privileges on the database or particular database objects.

- ALTER AUTHORIZATION

  Changes a non-DBA user password or default schema.

- DBA privileges

  Includes DBA-specific privileges, such as creating and dropping users.

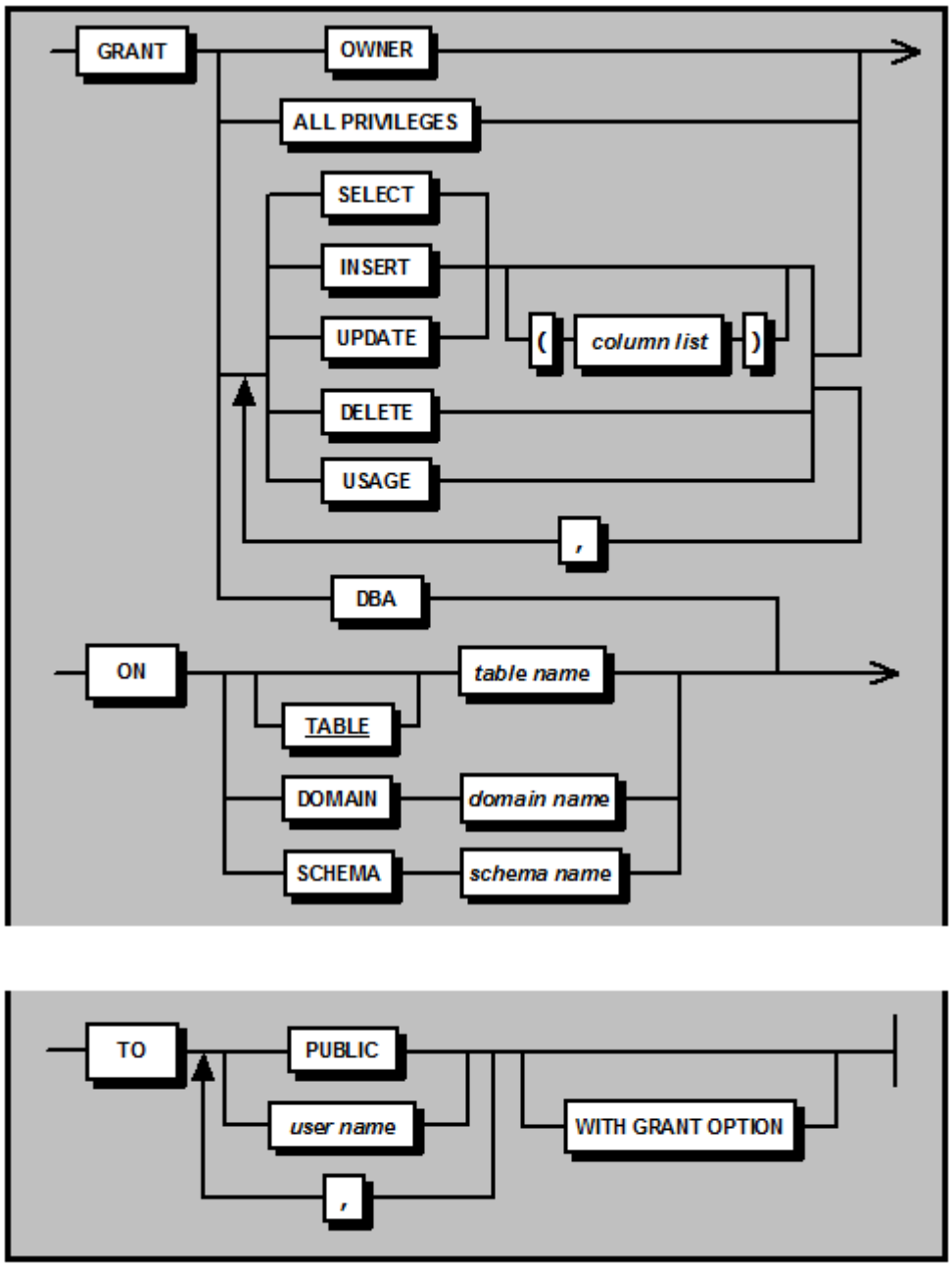
GRANT

The GRANT command confers database, table, and other privileges to database user authorization IDs. All relevant objects must exist in the database. Granted privileges are recorded in the appropriate privileges tables in the SYSTEM schema for the database. Specific privileges can be revoked subsequent to their granting, using the REVOKE statement.

Required Privileges

To grant a privilege on a database object, the user authorization must own the object, possess the privilege WITH GRANT OPTION on the object, or possess DBA privileges.

Syntax



The following list provides the syntax options for GRANT command:

column list

This specifies an optional list of columns in the table on which SELECT, INSERT, or UPDATE is being granted. The column list, if specified, must be enclosed in parentheses, with each column name separated from the next by a comma. If the column list is omitted, all columns in the target table are updatable by the grantee.

table name

domain name

schema name

These specify the name of an existing table/view, domain, or schema, respectively. If the table, view, or domain does not belong to the current schema, preface the object name with the schema name and a period (.), that is, *schema-name.object-name*. If privileges are being granted on a schema or domain, the object type keyword (SCHEMA or DOMAIN) must be included before the object name. If no object type is specified, the target object is assumed to be a table or view, and privileges are granted on the table/view having the specified name.

user name

The *user name* argument identifies one or more user authorization IDs (separated by commas) being granted the privileges specified. The PUBLIC keyword can be included in place of (or along with) an authorization ID (or a list of authorization IDs): this will grant the specified privileges to all current and future authorization IDs.

Description to GRANT

This statement format is used to grant privileges on specified database objects. One or more of the privilege option keywords may be specified after GRANT (separated by commas), allowing multiple privileges on a single database object to be granted in one statement.

The privileges resulting from the execution of a GRANT command statement are recorded as one or more individual *grants* in the privileges tables in the SYSTEM schema for the database. Each individual grant involves the granting of one privilege by a *grantor* to one *grantee,* except in the case of ALL PRIVILEGES, which is a shorthand designation for multiple privileges on a schema or table. A grantee can be any authorization ID (as recorded in the system catalog for that database), or the PUBLIC designation.

The same privilege can be granted to a single grantee by several different grantor authorization IDs. The grantee retains the privilege as long as one (or more) of these grants remains recorded in the appropriate privilege table in the SYSTEM schema. Grants (individual or groups) can be removed from the system tables by executing the appropriate REVOKE commands.

OWNER Privileges

This allows the recipient to create, alter, manipulate, drop, and grant/revoke privileges on objects in a schema that the user does not own. The OWNER privilege does *not* confer the ability to GRANT schema privileges or to drop the schema. This privilege is only valid with schemas; attempting to grant the OWNER privilege on a table or domain will produce an error message.

SELECT

This allows the user to retrieve data from the specified table, or all tables in the specified schema. The privilege can be granted on specific columns in the specified table by listing their names (within parentheses, and separated by commas) after the SELECT keyword. If a column list is not included, the privilege applies to all columns in the table.

INSERT

This allows the user to insert data into the specified table, or all tables in the specified schema. The privilege can be granted on specific columns in the specified table by listing their names (within parentheses, and separated by commas) after the INSERT keyword. If a column list is not included, the privilege applies to all columns in the table.

UPDATE

This allows the user to update data in the specified table, or all tables in the specified schema. The privilege can be granted on specific columns in the specified table by listing their names (within parentheses, and separated by commas) after the UPDATE keyword. If a column list is not included, the privilege applies to all columns in the table.

DELETE

This allows the user to delete data from the specified table, or all tables in the specified schema.

USAGE

This allows the user to specify *domain name* as the domain argument in a column definition.

ALL PRIVILEGES

The following table lists the individual privileges granted on a database object with ALL PRIVILEGES:

	Table	Schema	Domain
SELECT	Yes	Yes	No
INSERT	Yes	Yes	No
UPDATE	Yes	Yes	No
DELETE	Yes	Yes	No
USAGE	No	Yes	No

When a privilege is granted on a schema, that privilege applies to all objects belonging to the schema. For instance, granting SELECT on a schema allows the grantee to SELECT from any table in the schema.

WITH GRANT OPTION clause

If the WITH GRANT OPTION clause is included, the grantee can, in turn, grant the privileges specified in the GRANT command to other valid authorization IDs.

Note: The WITH GRANT OPTION clause cannot be included in a GRANT statement if either the privilege being granted is OWNER, or the recipient of the privilege is PUBLIC.

PUBLIC

To grant privileges to *all* users in the database, specify PUBLIC as the grantee. Users who have not been specifically granted any privileges have only those privileges granted to PUBLIC. When a privilege is granted to PUBLIC, the system continues to maintain the list of user names that have been specifically granted the privilege. Privileges granted to PUBLIC are granted to all new user authorizations subsequently created in the database.

Examples of Owner Privileges

The following example grants all authorization IDs the ability to update two specific columns (“col1” and “col3”) in a table called “my_table”:

```
GRANT UPDATE (col1, col3) ON TABLE my_table TO PUBLIC;
```

In this case, users without DBA privileges, who possess no other privileges on *my_table*, cannot update any other columns.

The following example grants DBA privileges to an authorization ID called “marketing”:

```
GRANT DBA TO marketing;
```

The following example grants to authorization IDs “clive”, “bertrand”, and “nigel” the ability to specify domain “d_char” as a domain argument in a column definition (and also to GRANT USAGE on the domain to other users):

```
GRANT USAGE ON DOMAIN d_char
     TO clive, bertrand, nigel WITH GRANT OPTION;
```

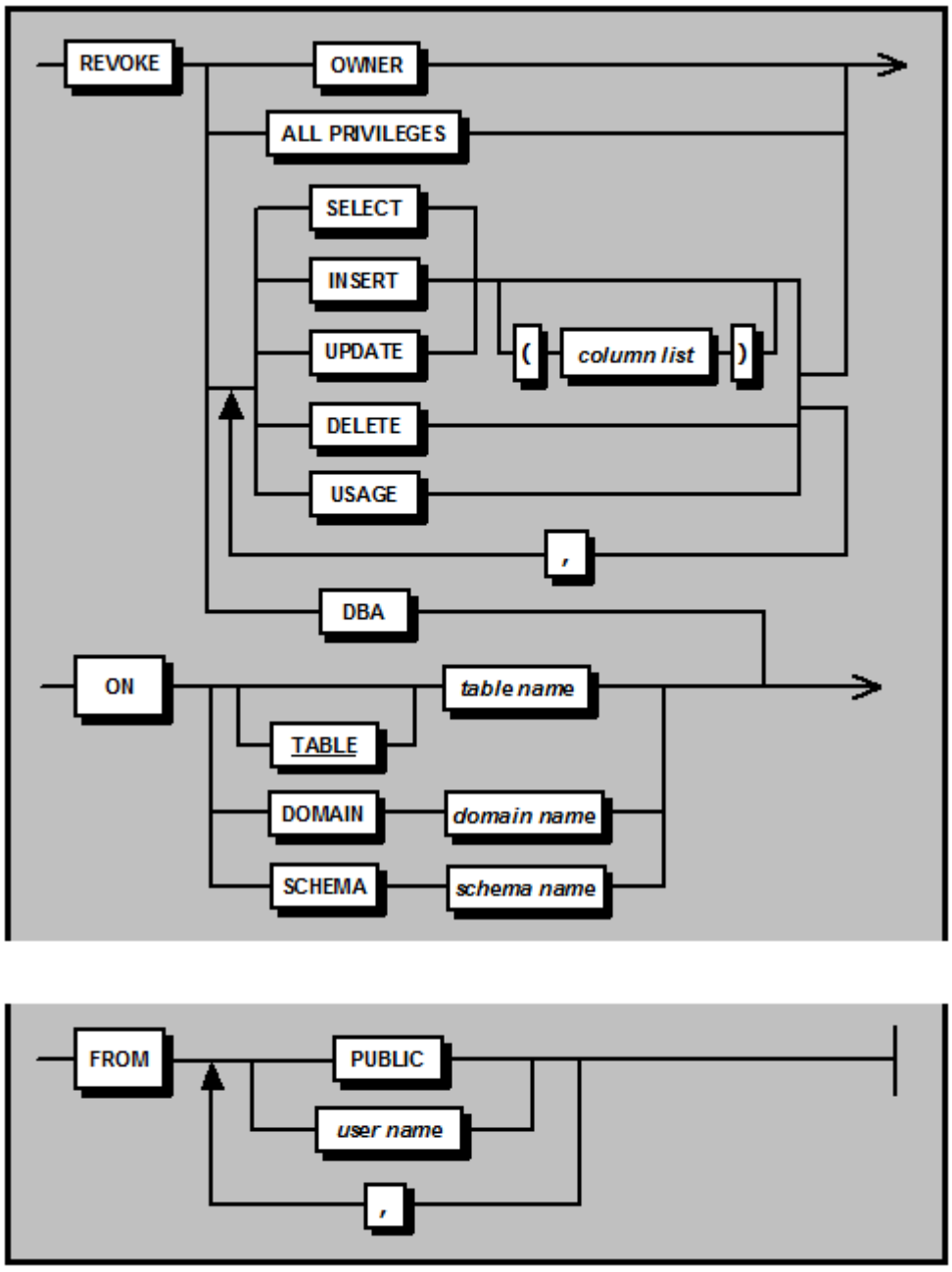
REVOKE

The REVOKE command cancels database privileges held by specified authorization IDs. All relevant objects must exist in the database. Revoked privileges are deleted from the privilege tables in the SYSTEM schema for the database.

Required Privileges

In most instances, the authority needed to revoke a specific privilege must be identical to the authority that originally granted the privilege — which means that privileges individually granted by a specific authorization ID should be revoked individually by the same authorization ID. However, a user authorization with DBA privileges may revoke any privilege regardless of whether that user granted the privilege originally.

Syntax



The following list provides the syntax options for GRANT command:

column list

The *column list* argument specifies an optional list of columns in the table on which the SELECT, INSERT, or UPDATE privilege is being revoked. The column list, if specified, must be enclosed in parentheses, with each column name separated from the next by a comma.

table name

domain name

schema name

These specify the name of an existing table/view, domain, or schema, respectively. If the table, view, or domain does not belong to the current schema, preface the object name with the schema name and a period (.), that is, *schema-name.object-name*. If privileges are being revoked from a schema or domain, the object type keyword (SCHEMA or DOMAIN) must be included before the object name. If no object type is specified, the target object is assumed to be a table or view, privileges are revoked from the table/view having the specified name.

user name

The *user name* argument identifies one or more user authorization IDs (separated by commas) from which the specified privilege or privileges are being revoked. The PUBLIC keyword option can be specified along with or in place of an authorization ID (or a list of authorization IDs), revoking the specified privileges from all valid authorization IDs.

Description to REVOKE

This statement format is used to revoke privileges on specified database objects. One or more of the privilege option keywords may be specified after REVOKE (separated by commas), allowing multiple privileges on a single database object to be revoked in one statement.

Individual grants (or groups of grants) are removed from the privilege tables in the SYSTEM schema for the database by executing REVOKE command statements. Each individual REVOKE action involves the removal of one privilege by one *revoker* from one *revokee*, except when ALL PRIVILEGES is revoked on a table or schema.

If ALL PRIVILEGES on a table or schema is revoked from a user authorization, each of the individual privileges that constitute ALL PRIVILEGES for the object is removed (regardless of whether they were originally granted with the ALL PRIVILEGES option). Conversely, if ALL PRIVILEGES was granted on a table or schema, the individual privileges that make up ALL PRIVILEGES may be subsequently revoked one at a time.

ALL PRIVILEGES on a domain consists solely of the USAGE privilege, so this privilege may be removed by revoking either ALL PRIVILEGES or USAGE on the domain.

DBA privileges cannot be revoked partially by revoking individual object privileges; they must be revoked altogether.

See the description of GRANT above for more information about the privileges that may be revoked.

Note: Since the same privilege can be granted to a single grantee by several different grantors, the grantee retains the privilege as long as one (or more) of these grants remains recorded in the privilege tables in the database SYSTEM schema.

ALTER AUTHORIZATION

Use the ALTER AUTHORIZATION command to change a non-DBA user password or default schema.

Any user can run the ALTER AUTHORIZATION...SET PASSWORD command to change their own password. Any user can also run the ALTER AUTHORIZATION...SET DEFAULT SCHEMA to change their own schema, as long as the user owns or has OWNER privileges on the default schema.

If you have DBA privileges, you can run the ALTER AUTHORIZATION command to change another user's default schema or password.

To run an ALTER AUTHORIZATION command, form a statement with the user name and the desired password or default schema. The specified user name must exist in the database. The password you enter can be up to 128 characters long, is case-sensitive, and must be surrounded by single quotes.

If you specify a schema in the ALTER AUTHORIZATION...SET DEFAULT SCHEMA clause that does not exist in the database, Data Vault creates the schema and assigns the user OWNER privileges with the GRANT option on the schema. If the default schema you specify exists in the database but the user does not own it, Data Vault assigns the user OWNER privileges without the GRANT option on the schema.

You can run one ALTER AUTHORIZATION command at a time. To change both the password and default schema for a non-DBA user, run the ALTER AUTHORIZATION...SET PASSWORD and ALTER AUTHORIZATION...SET DEFAULT SCHEMA commands separately.

Example

The following statement changes the default schema to *vip* for *user1*:

```
ALTER AUTHORIZATION user1 SET DEFAULT SCHEMA vip;
```

The following statement changes the password to *pass123* for *user1*:

```
ALTER AUTHORIZATION user1 SET PASSWORD 'pass123';
```

DBA Privileges

The DBA role gives a user all privileges over all objects in the database. Only a user with DBA privileges can execute the following SQL commands:

CREATE AUTHORIZATION

Creates a database user authorization identifier, which consists of a user name and optional password. When you form the CREATE AUTHORIZATION statement, you can also assign the new user to a default schema. After you commit the CREATE AUTHORIZATION statement, you can connect to the database with the new user.

The user name argument must be unique in the database. The user name can be up to 128 characters long and cannot be "PUBLIC" or "DBA."

The password argument is optional. If you include a password argument in the CREATE AUTHORIZATION statement, the password must be preceded by `PASSWORD`. The password can be up to 128 characters long and is case-sensitive. You must enter the password every time the user connects to the database.

If you do not include a password argument in the CREATE AUTHORIZATION statement, the password has a null value and you will not need to enter a password when you connect to the database.

Optionally, you can assign the user to a default schema. If you include a DEFAULT SCHEMA clause in the CREATE AUTHORIZATION statement, the specified schema is always the initial schema for the user when the user connects to the database. If you do not include a DEFAULT SCHEMA clause, the initial schema for the user is the PUBLIC schema.

If you specify a schema in the DEFAULT SCHEMA clause that does not already exist in the database, Data Vault creates the schema and assigns the new user OWNER privileges with the GRANT option on the schema. If the DEFAULT SCHEMA you specify already exists in the database, Data Vault assigns the new user OWNER privileges without the GRANT option.

Example

The following statement creates a user authorization called *user1* with the password *pass*:

```
CREATE AUTHORIZATION user1 PASSWORD 'pass';
```

The following statement creates the user authorization *u1* with a null password, and makes the default schema for the user *vip*:

```
CREATE AUTHORIZATION u1 DEFAULT SCHEMA vip;
```

DROP AUTHORIZATION

Drops the specified user authorization and all associated privileges. If any database objects belong to the user authorization that you try to drop, the DROP command will fail. You cannot drop the DBA or PUBLIC users.

Example

The following statement drops the user authorization *user1* from the database:

```
DROP AUTHORIZATION user1;
```

ALTER AUTHORIZATION...SET PASSWORD

Changes the password for a non-DBA user. The user name you specify in the ALTER AUTHORIZATION statement must exist in the database. The password you enter can be up to 128 characters long and is case-sensitive.

Example

The following statement sets the password for *user1* to "pass123":

```
ALTER AUTHORIZATION user1 SET PASSWORD pass123;
```

Note: Do not use the ALTER AUTHORIZATION statement to change the DBA user password.

ALTER AUTHORIZATION...SET DEFAULT SCHEMA

Changes the default schema for a non-DBA user. If you specify a schema in the DEFAULT SCHEMA clause that does not exist in the database, Data Vault creates the schema and assigns the user OWNER privileges with the GRANT option on the schema. If the DEFAULT SCHEMA you specify exists in the database but the user does not own it, Data Vault assigns the user OWNER privileges without the GRANT option on the schema.

Example

The following statement changes the default schema for *user1* to *vip*:

```
ALTER AUTHORIZATION user1 SET DEFAULT SCHEMA vip;
```

SHUTDOWN

Shuts down the Data Vault service.

SET TRANSACTION...ISOLATION LEVEL EXCLUSIVE

Data Vault Passwords

You can configure optional policies for the user passwords used to access the Data Vault, either through the Data Vault SQL Tool or ODBC/JDBC clients. Select and configure the policies that you want to enforce based on company IT security policies. When a user changes their Data Vault password, they must separately update the password in the ILM repository.

The following Data Vault password policies are available:

Minimum password length

Specifies the minimum number of characters that a user's Data Vault password must contain.

Password complexity

When enabled, the password used to access Data Vault must contain at least one upper-case letter and at least one of the following special characters: @ # _ = , . ? ~ { }

Password duration

Requires a user to change their Data Vault password after a certain number of days.

Password blacklists

Prevents a user from re-using a specified number of previous passwords.

To enable a password policy, you must add the corresponding policy parameter to the [SERVER] section of the ssa.ini file. For more information, see the topic "SERVER Section" in the chapter "Data Vault Configuration."

The following characters are valid characters for a Data Vault password:

- Upper and lower-case letters A-Z
- Numbers 0 through 9
- Special characters: @ # _ = , . ? ~ { }

Do not use the asterisk (*) character in the password.

Users can change a password themselves, or the DBA user can change a password for another user. The DBA user password never expires, and has a maximum of 30 characters.

Data Vault Passwords in the ILM Repository

When a Data Vault password expires, users must first update the password in Data Vault and then separately update the password in the ILM repository. If a user does not update their Data Vault password in the ILM repository, they are unable to connect to Data Vault from Data Archive. They will not be able to browse data, retire data, create a target connection, or perform other operations involving Data Vault.

You can update a Data Vault password in the ILM repository with the "PasswordChangeEnv.bat" or " PasswordChangeEnv.sh" utilities, which are available in the `optional` directory of the Data Archive installation. You can also use the utilities to update the password for the Data Vault connection by giving the connection name and password.

If you have enabled SSL in Data Archive, you must also enable SSL in the PasswordChangeEnv.sh or PasswordChangeEnv.bat utilities. Add the trust store parameter to the file in same manner as the `startApplimation.sh` or `startApplimation.bat` files. Open PasswordChangeEnv.sh or PasswordChangeEnv.bat in any text editor and add the following information:

-Djavax.net.ssl.trustStore=<full path and name of .jks truststore file> -
Djavax.net.ssl.trustStorePassword=<password of .jks file>

For example:

java -Dfile.encoding=UTF-8 -Djavax.net.ssl.trustStore=/data/idvuser/Jenkins_slave/workspace/644/DA/
irs11ilm02.jks -Djavax.net.ssl.trustStorePassword=password
com.applimation.util.RepositoryDatabaseUserPasswordChange $location

To change a Data Vault password or a Data Vault password in the ILM repository, first configure the `PasswordChangeSampleProperty.properties` file, which is also available in the `optional` directory of the Data Archive installation. When you configure the file you provide the ILM repository name, username, and new Data Vault password. You can also designate a separator value if you are providing multiple user names

and passwords. You can choose any value for the separator as long as it is not used in the password. For example:

```
amhome.values.separator=##
    amhome.rep.names=demo2_expdays2
    amhome.rep.usernames=demo1
    amhome.rep.newpasswords= D@@123
```

After you configure and save the `PasswordChangeSampleProperty.properties` file, run the PasswordChangeEnv.bat or PasswordChangeEnv.sh utility, depending on the operating system. The utility asks for the full path of the properties file that you configured, and returns either an error or success message.

Updating Passwords in the JReport Catalog

When the Data Vault password expires, you will receive a "password expired" message when you try to run Data Visualization reports. To run Data Visualization reports, update the Data Vault password in both Data Vault and the ILM repository, then update the Data Vault password in the JReport catalog.

1. Use the Data Vault SQL Tool or ODBC/JDBC client to update the Data Vault password.
2. Use the `PasswordChangeEnv.bat` or `PasswordChangeEnv.sh` utility to update the Data Vault password in the ILM repository.
3. Open JReport Designer.
4. Click **File** > **Publish and Download** > **Download from Server**.
5. Enter the required connection details for the Data Archive server. Give the following servlet path: `/visualization/jrserver`.
6. Click **OK**.

   The **Download Resource From** window appears.
7. Select the public report and download the catalog.
8. Edit the password in the catalog and save it.
9. Export the catalog to the Data Archive server using the correct "Publish Resource To" path.

You can now run existing Data Visualization reports.

DBA User Passwords

To change the DBA user password, you must use the ChangeDBAPassword utility. The utility is in the Data Vault installation directory. On UNIX systems the file name is `ChangeDBAPassword.sh` and on Microsoft Windows systems the file name is `ChangeDBAPassword.bat`.

The following characters are valid for a DBA user password:

- Upper and lower-case letters A-Z
- Numbers 0 through 9
- Special characters: @ # _ = , . ? ~ { }

On Microsoft Windows, if you use special characters in a password created with the ChangeDBAPassword utility, enclose the password string within double quotation marks. For example:

```
ChangeDBAPassword.bat demo1 "A=,B=,"
```

In the example above, the new password contains special characters so it is enclosed within double quotation marks.

If both the old and new passwords contain special characters, enclose both passwords within double quotation marks. For example:

```
ChangeDBAPassword.bat "A=,B=," "DEMO?#="
```

If either the new or old password begins with the number sign (#), you must enclose the password within double quotation marks. If you do not, the utility fails in UNIX environments.

Do not use the asterisk (*) character in the password.

Prerequisites

Before you use the ChangeDBAPassword utility, complete the following prerequisites:

1. Set the JAVA_HOME environment variable and add the bin directory inside JAVA_HOME to the PATH environment variable.
2. Install Data Vault.
3. Verify the original password and the new password that you want to set.

Changing the DBA User Password

To change the DBA user password, run the ChangeDBAPassword utility and provide both the old and new passwords.

1. Change directories to the Data Vault folder.
2. Run the following command:
   - On Microsoft Windows: `ChangeDBAPassword.bat <old password> <new password>`
   - On Unix: `ChangeDBAPassword.sh <old password> <new password>`

CHAPTER 15

ssasql Command Line Program

This chapter includes the following topics:

- ssasql Command Line Program Overview, 193
- ssasql Operating Modes , 193
- ssasql Example, 194
- ssasql Arguments, 194
- ssasql Session Mode , 196
- ssasql Session Mode Commands, 196
- ssasql SQL Mode , 198
- ssasql SQL Mode Command Syntax , 198
- ssasql SQL Mode Commands , 199
- Display Formats, 204

ssasql Command Line Program Overview

ssasql allows users to connect to a Data Vault repository database and execute SQL statements interactively.

Start ssasql from the operating system command prompt with the following syntax:

```
ssasql [ flags ] [ connection-name instance-name user-name[/user-password ] ]
```

ssasql Operating Modes

ssasql can operate in SQL mode or in Session (SESS)mode. Each mode has its own set of commands. Additionally, there is a set of system-level commands. Session mode enables users to connect and disconnect database sessions. SQL mode enables users to execute SQL statements interactively against a database. The system-level commands include commands for switching between the operating modes.

The ssasql prompt always reflects the mode currently in use. When in Session mode, ssasql displays a prompt of the form **SESS:*n*>**; when in SQL mode, the prompt is **SQL:*n*>** (where *n* is a number reflecting the command number for each mode). Each time a command is issued within a particular mode, the command number displayed in the prompt increases by one.

When the ssasql invocation connects to the database successfully, the user is automatically placed in SQL mode. If the ssasql invocation does not connect successfully, either because the invocation arguments are invalid or are not included, the user is automatically placed in Session (SESS) mode.

The standard redirection character (**<**) may be used in conjunction with the ssasql invocation to execute SQL commands contained in a batch file, as in the following example:

```
ssasql server1_vip dba <batch.sql
```

This ssasql invocation connects to a Data Vault repository database or Nucleus database instance called **vip**, using the connection **server1_vip**, as the user DBA. The SQL statements contained in the ASCII file **batch.sql** are executed automatically. To return control to the operating system shell when command execution is finished, include the **.EXIT** system command at the end of the batch file.

ssasql Example

The ssasql invocation in this example connects to a Data Vault repository database or Nucleus database instance called **vip** as **user1**, using a connection definition called **server1_vip**. The user authorization password is **PaSs1**. Session logging is turned on with the **-o1** flag. Following a successful connection, user1 is automatically placed in SQL mode (in the default schema associated with that user authorization).

```
ssasql -o1 server1_vip vip user1/PaSs1
```

ssasql Arguments

connection-name

When connecting to the Load Balancer, *connection-name* must refer to a CONNECTION section in the client nucleus.ini file that contains the Host and Port information used by the Load Balancer (specified in the ssa.ini file). When connecting to a running Nucleus database instance or Data Vault repository database, *connection-name* must be the name of the connection used to start *instance-name*.

The *connection-name* argument is optional, but must be entered in order to connect automatically to a Nucleus instance or Data Vault repository database. If ssasql is being used to connect to a database instance running on a remote server, the client-side nucleus.ini file must contain a [CONNECTION *connection-name*] section that specifies the port number and host name used to connect to the database instance, and these must match the values specified for the same connection in the server-side nucleus.ini file. The connection name is not case-sensitive.

Note: By default, the maximum number of client programs that may use a particular connection is **10** (this limit may be changed by including a MaxUsers entry in the appropriate CONNECTION section of the server-side nucleus.ini file).

instance-name

The name of the Data Vault repository database or Nucleus database instance to which to connect. The *instance-name* argument is optional. It must be entered, however, if a *connection-name* argument has been specified. *Instance-name* must have already been started with the specified connection. The instance name is not case-sensitive.

user-name

The *user-name* argument is optional. It is required, however, if *connection-name* and *instance-name* arguments are entered. The *user-name* argument must specify a valid user authorization in the specified database. The user name is not case-sensitive.

/user-password

The *user-password* argument is required only if a *user-name* argument is entered, and if the specified user authorization has an associated password. The *user-password* argument must be separated from the *user-name* argument only by a slash character (**/**); no “white space” may appear between the two arguments. The password must precisely match the password associated with the specified user authorization.

Note: User passwords are case-sensitive.

flags

The optional flags described below may be specified individually, or in combination (if multiple flags are specified, they must be separated by spaces). When the flag includes an argument, these elements cannot be separated by any “white space”.

-h or -?

The **-h** (or **-?**) flag displays a help screen which describes the ssasql invocation flags and arguments.

-f *file-name*

The **-f** flag lets the user specify a name and location for the session log file created through the .OUTPUT system command. By default, the log file is called **inter.out** and is created in the current working directory. If the full path to the log file contains any spaces, put quotation marks around the path. If only the file name is specified (that is, the path location is omitted), the file is created in the current directory.

-o{0 | 1}

The -**o** flag turns the session logging off (0) or on (1). The default setting is “off” (that is, **-o0**). The **-o1** flag turns session logging on, and creates a file with the name **inter.out**, in which all ssasql input and output will be recorded. The inter.out file is always created in the current directory and overwrites any existing inter.out file.

-p{0 | 1}

The **-p** flag turns the ssasql command prompt off (0) or on (1). The default setting is “on”, that is,

-p1. Include the **-p0** flag in the invocation in order to turn off the prompt display.

-t{0 | 1}

The **-t** flag turns the title list display off (0) or on (1). The default setting is “on”, that is, **-t1**. Include the **-t0** flag in the invocation in order to suppress the display of the column headings returned by SQL SELECT statements.

-w{0 | 1}

The **-w** flag turns the SQL warning message display off (0) or on (1). The default setting is “on”, that is, **-w1**. Include the **-w0** flag in the invocation in order to suppress the display of SQL warning messages.

ssasql Session Mode

Session mode enables users to establish database sessions (that is, connect to a Nucleus database instance), change from one session to another, and disconnect database sessions. The .SESS system command enables users currently in SQL mode to access ssasql Session mode and execute Session mode commands.

When ssasql is invoked without connection name, database name, and user name arguments, or if any of the specified arguments is invalid, the user is automatically placed in Session mode. The CONNECT command can subsequently be used to establish a database session.

ssasql Session Mode Commands

The following commands are supported in Session mode. Each must be terminated by a semicolon (**;**). When in Session mode, users may enter the HELP command followed by a semicolon (**;**) in order to display all available Session mode commands.

CONNECT

The CONNECT command enables a user to establish a database session by connecting to a running Nucleus database instance or Data Vault repository database. Up to 10 sessions may be established from a single client. When ssasql successfully connects to a database/instance, the user is automatically placed in SQL mode (in the default schema associated with the specified user authorization).

The CONNECT command has the following form (single quotation marks must be typed literally as they appear):

```
CONNECT 'connection-name' 'instance-name' 'user-name[/password]' ;
```

- connection-name

  The *connection-name* argument must match the connection name used when the Data Vault repository database or database instance was started. If ssasql is being used to connect to a database/instance running on a **remote** server, the client-side nucleus.ini file must contain a [CONNECTION *connection-name*] section that specifies the port number and host name used to connect to the database/instance, and these must match the values specified for the same connection in the server-side nucleus.ini file. The connection name is not case-sensitive.

  Note: By default, the maximum number of client programs that may use a particular connection is **10** (this limit may be changed by including a **MaxUsers** entry in the appropriate CONNECTION section of the server-side nucleus.ini file).

- instance-name

  The *instance-name* argument specifies the name of the Data Vault repository database or database instance to which to connect. It must have been started using the specified connection name. The instance name is not case-sensitive.

- user-name

  The *user-name* argument must specify a legal user authorization in the database specified by the *instance-name* argument. The user name is not case-sensitive.

- password

  The *password* argument is required only if there is a password associated with the user authorization specified in the *user-name* argument. The password argument is case-sensitive, and must match the user's password exactly. The password argument must be preceded by a slash (/).

  Note: User passwords are case-sensitive.

The CONNECT command in this example connects as user dba to a Data Vault repository database or database instance called vip, started using the connection server1_vip. No password has been assigned to the user.

```
CONNECT 'server1_vip' 'vip' 'dba';
```

DISCONNECT

The DISCONNECT command disconnects database sessions established from the current client (ssasql). The DISCONNECT command performs an implicit ROLLBACK on the current transaction in the specified session. The DISCONNECT command has the following form:

```
DISCONNECT [ session | ALL ] ;
```

session

The *session* argument is optional. If no session is not specified, the current session (that is, the one denoted by the asterisk (*) in the ssasql status display) will be disconnected. If included, the session argument is the number corresponding to the session to be disconnected.

ALL

The ALL keyword may be entered instead of the *session* argument in order to disconnect all database sessions established from the current client.

This example illustrates the use of the DISCONNECT command to disconnect a database session identified as session 3.

```
DISCONNECT 3;
```

In this example, the current database session is disconnected.

```
DISCONNECT;
```

DURATION

The DURATION command specifies a maximum duration to be imposed on the processing of all SQL SELECT statements issued from the current session. After the DURATION command is issued, any SQL query issued within the session is subject to the specified limit on duration and will expire upon exceeding it. Query expiration restores the SQL prompt, and the program proceeds as if the query had never been issued. Other sessions established from the same client are unaffected. The DURATION command has the following general form:

```
DURATION minutes ;
```

minutes

The *minutes* argument is required and must be an integer corresponding to the desired maximum SQL query duration, specified in minutes. Specifying a minutes argument of 0 (zero) results in an infinite duration (that is, no limit on query processing time). All sessions have a default duration of 0.

The DURATION command in this example specifies a maximum SQL query duration of fifteen minutes for the current session.

```
DURATION 15;
```

USE

The USE command changes the current database session. That is, the session specified as the argument for the USE command becomes the new current session, and all subsequent SQL statements are executed in the context of that current database session. The USE command has the following general form:

```
USE session-number ;
```

session

The *session* argument is required, and must be a number corresponding to an existing database session. The session identified by the session argument will become the current session. Only sessions established by the current client may be targeted by the USE command.

In this example, the USE command changes the current session to session 2.

```
USE 2;
```

ssasql SQL Mode

ssasql SQL mode enables users to execute SQL statements interactively against a particular database. Any legal Data Vault Service SQL statement may be executed in SQL mode once a database session has been established. All SQL statements must be terminated by a semicolon (**;**). There can be a single SQL statement per line. If multiple SQL statements, separated by semicolons, are included on the same line, only the first statement will be processed; the rest will be ignored.

When a connection name, database instance name, and user name are included in the ssasql invocation, the user is automatically placed in SQL mode (in the default schema associated with the specified user) upon successful connection to the database instance. If the user switches to Session mode once a database session has been established, the **.SQL** system command can be used to switch back to SQL mode.

In SQL mode, pressing <Ctrl+C> while a query is executing will halt the query. Pressing <Ctrl+C> *twice* in this situation will cause ssasql to shut down.

When connected to a Data Vault repository database, please note that only SELECT statements can be executed against an archived table, with the following additional restrictions:

- Only one table can be queried
- Subqueries and joins (including self-joins) are not permitted
- the FETCH FIRST...ONLY clause is not supported
- The WHERE clause can contain only Boolean value expressions, excluding the EXISTS predicate.

Note: By default, the maximum number of concurrently executing SQL commands for a single client process is **128**. This limit may be changed for the connection by including a **MaxQueries** entry in the appropriate CONNECTION section of the server-side nucleus.ini file.

ssasql SQL Mode Command Syntax

Ssasql SQL mode commands are prefixed with a period (**.**) in order to distinguish them from the command set for the current mode. Any command that is not prefixed by a period is executed within the context of the

current mode; ssasql issues an error message if the command does not belong to the current mode's command set. Ssasql commands are not case-sensitive.

In order to execute a command belonging to the current operating mode, the command must not be prefixed by a period, and must be terminated by a semicolon (**;**).

To execute a command in another mode without leaving the current mode, precede the command with the system command normally used to change to the desired mode, as in the following example:

```
SQL:3> .sess disconnect 3;
```

Here, the DISCONNECT statement is executed from SQL mode by prefixing it with the system command normally used to change to SESS mode, that is, **.SESS**. Note that the statement is terminated with a semicolon (;) — in this case, however, it is not necessary to include the semicolon, since the command is executed with the mode prefix.

ssasql SQL Mode Commands

ssasql system commands may be executed from either operating mode. Unlike modal commands, system commands must fit entirely on the current prompt line. Any command prefixed by a period that is not recognized as a system command is passed to the operating system shell for execution.

.DATA

The .DATA command switches the display of fetched data off or on. When switched off, ssasql performs query fetches but does not show the retrieved data. Only the number of rows returned and the column headers are displayed.

.EXIT

The .EXIT command returns the user to the operating system shell. Ssasql automatically disconnects all currently active sessions initiated from the client.

.EXPORT [BULK] { *SQL-statement* } INTO { |*pipe* | '*file*' }

CSV ['*delimiter-character*'

'*column-separator-character*'

'*row-separator-character*'

'*null-character*']

The .EXPORT CSV command exports the results of a query to a flat file or pipe in the comma-separated value (CSV) format. The full SELECT statement is included as a parameter for this command, enclosed by curly braces **{ }**. A terminating semicolon is optional for the SELECT statement. If exporting to a file, the specified file path/name must be contained in single quotation marks. If exporting to a pipe, the pipe number (from **0** to **255**) must be specified without quotes, preceded by a pipe symbol (|).

The optional special characters (value delimiter, column separator, row separator, and null indicator) are each represented by a single character in either ASCII or hexadecimal form, and enclosed by single quotation marks. These special characters are defined as follows:

delimiter-character

The delimiter character is used to delimit a value if it contains a double quotation mark ("), comma (**,**), or newline character (**\n**). By default, this delimiter character is a double quotation mark ("). For example, if the source value is:

```
oh,my
```

by default the value will be written to the CSV file as:

```
"oh,my"
```

Note: that the .EXPORT CSV operation will escape a double quotation mark by preceding it with another double quotation mark, if it appears in a string value. So if the source value is this:

```
xyz""123
```

and the default delimiter (") is used, the following is the string that will be exported:

```
"xyz""""123"
```

column-separator-character

The column separator character is used to separate the fields in each record. By default, this character is a comma (**,**).

row-separator-character

The row separator character is used to delimit rows (records). By default, this character is a newline (**\n**).

null-character

The null character is used to represent a null value in the exported data. There is no default null character.

Note: that the special characters can be omitted only if there are no null values in the exported data, as there is no default null character.

There is a "bulk fetch" option that allows data to be exported at a faster rate than with the standard .EXPORT. To enable bulk fetching, simply include the BULK keyword after the .EXPORT keyword and before the SQL clause.

Example:

```
.EXPORT BULK { SELECT id, ddate, price, qty, sku
FROM s1.t1 WHERE ddate > '2004-01-01' }
INTO 'C:\exp.dat'
CSV '\x22' ',' '\x0a' '‡'
```

In the above .EXPORT statement, the results of the embedded SELECT statement are to be written to the specified file (*exp.dat*), via bulk fetching from the server. Here, the escape character is **\x22** (the hexadecimal representation of the double quotation mark character), the column separator is a comma (**,**), the row separator is **\x0a** (the hexadecimal representation of the linefeed character), and the null character is the ASCII "double dagger" symbol

(**‡**).

.HELP

The .HELP command displays a list of all system commands and their syntax.

.HEX

The .HEX command switches the hexadecimal display of binary data on or off. When turned off, unprintable characters are output as periods (‘**.**’). This command is useful when querying BLOB columns.

.INDEX

The .INDEX command toggles the display of a row index. When turned on, returned rows will have an extra column called **ID**, which contains the numerical order in which the rows were fetched starting from 1. By default, row indexing is turned off.

.INTERVAL [REPEAT | NO REPEAT] [*n*] *SQL-statement* ;

The .INTERVAL command executes the specified SQL statement and displays *n* rows, sampled at regular intervals from the result set. If the *n* argument is not supplied, the value is assumed to be **100**. When the REPEAT option is included (which is also the default), the number of rows specified is the number of rows displayed, even if it is greater than the total number of rows in the result set. If necessary, some or all of the displayed rows are duplicated. When the NO REPEAT option is specified, there is no row duplication, and the number of rows displayed can be less than n.

The SQL statement must end with a semicolon (**;**).

.MAXLENGTH

The .MAXLENGTH command switches full display of character data on or off. By default, ssasql limits the data display to **40** characters.

.MAXROWS *n*

The .MAXROWS command specifies the maximum number of rows (*n*) returned by a query. That is, if *n* is less than the number of rows normally returned by a query, an arbitrary subset of *n* rows will be returned instead. The default is **0** (unlimited).

Note: While this system command is enabled, the true number of rows returned by a query will be obscured by the MAXROWS limitation. To disable the command, use **.MAXROWS 0**.

.MEASURE

The .MEASURE command switches the timing of query fetches on or off. When switched on, query output will also display the elapsed time for command execution, each fetch stage (512 KB of data per fetch stage), and the query as a whole.

.NULLS [*string*]

The .NULLS command establishes a display string for null values returned by SQL SELECT statements. Ssasql will display the character string provided in the string argument in place of any null values returned by SQL queries. Consider this example:

SQL:1> .NULLS N/A

After this command is executed, ssasql will return the string 'N/A' in place of any nulls returned by SELECT statements. To clear the null string setting, execute the .NULLS command without an argument.

.OUTPUT

The .OUTPUT command switches session logging on or off. When session logging is on, all ssasql input and output is recorded in a file named **inter.out** in the current working directory. If logging is turned off and then on again within a single ssasql session, ssasql continues to append to the output file without overwriting its contents. Turning on logging in a subsequent session, however, will overwrite any preexisting output file. A session can be started with logging turned on by including the **-o** flag in the ssasql invocation.

Warning: When session logging is enabled, executing queries that produce large result sets can cause the log file to grow quite large. Eventually, disk space can be exhausted, and Data Vault Service will terminate abnormally. If possible, avoid logging large queries, or else use the .QUIET or .DATA command to hide the results of such queries.

.PROMPT

The .PROMPT command switches the SQL prompt display (that is, SQL:*n*>) on and off. A session can be started with the prompt display turned off by including the **-p** flag in the ssasql invocation.

.QUIET

The .QUIET command switches SELECT statement output off or on. After execution of the .QUIET command, ssasql returns only a count of the rows constituting the SELECT statement result, and not the actual data.

.RUN *file*

The .RUN command executes a batch of commands, either contained in a host file, or entered from the console. The *file* argument must be the name of an ASCII text file containing valid commands for the current mode, or system commands.

Consider the following example:

```
SQL:1> .RUN sql.txt
```

The contents of the file **sql.txt** must be valid Data Vault Service SQL statements, since ssasql is in SQL mode (unless system mode change commands are included in the file). Ssasql executes the statements in the order of their placement in the file. If a hyphen (**-**) is specified for the file argument, ssasql accepts valid commands from the console until **Ctrl+D** is entered (indicating end of input).

.SESS [command]

The .SESS command, when executed in SQL mode, changes the current operating mode to Session mode. If the optional *command* argument is included, it must be a valid Session mode command. The Session mode command specified is executed without leaving the current ssasql mode.

.SQL [command]

The .SQL command, when executed in Session mode, changes the current operating mode to SQL mode. If the optional *command* argument is included, it must be a valid Data Vault Service SQL statement. The specified SQL command is executed without leaving the current ssasql mode.

.STARTROW *n*

The .STARTROW command specifies the row in the result set from which to start fetching. For example, if *n* is 5, the first four rows in the result set are not displayed by ssasql. By default, all rows are displayed (*n* = 1).

.STATS

The .STATS command displays the current values of ssasql system variables as well as the database connection status for the client.

ssasql STATS Display

The .STATS display has the following appearance:

```
SQL:3> .STATS
current status
NULL string:    N/A
OUTPUT flag:    ON
PROMPT flag:    ON
TITLES flag:    ON
DATA flag:      ON
```

Warning: WARNINGS flag: OFF

```
QUIET flag:     OFF
INDEX flag: OFF
MEASURE flag:   OFF
```

```
MAXLENGTH flag: OFF
HEX flag: OFF
MAXROWS:        DISABLED
INTERVAL REPEAT: TRUE
INTERVAL rows: 100
STARTROW: 1
CONNECTIONS:
*   1: CN1@db01  (Nucleus 4.1.1994.0)  SERIALIZABLE
    2: CN1@db01  (Nucleus 4.1.1994.0)  SERIALIZABLE
```

- NULL string. Displays the NULL string specified using the **.NULLS** system command. Here, the string has been set to 'N/A.'
- OUTPUT flag. Displays OUTPUT flag status as set by the **-o** option flag or the **.OUTPUT** system command (ON or OFF).
- PROMPT flag. Displays PROMPT flag status as set by the **-p** option flag or the **.PROMPT** system command (ON or OFF).
- TITLES flag. Displays TITLES flag status as set by the **-t** option flag or the **.TITLES** system command (ON or OFF).
- DATA flag. Displays the DATA flag status as set by the **.DATA** system command (ON or OFF).

Warning: WARNINGS flag. Displays WARNINGS flag status as set by the **-w** option flag or the **.WARNINGS** system command (ON or OFF).

- QUIET flag. Displays QUIET flag status as set by the **.QUIET** system command (ON or OFF).
- INDEX flag. Displays INDEX flag status as set by the **.INDEX** system command (ON or OFF).
- MEASURE flag. Displays the MEASURE flag status as set by the **.MEASURE** system command (ON or OFF).
- MAXLENGTH flag. Displays MAXLENGTH flag status as set by the **.MAXLENGTH** system command (ON or OFF).
- HEX flag. Displays HEX flag status as set by the **.HEX** system command (ON or OFF).
- MAXROWS. Displays the MAXROWS value specified using the .MAXROWS system command. Here, the maximum number of rows has not been set explicitly, so it defaults to unlimited.
- INTERVAL REPEAT. Displays whether the **.INTERVAL** option is set to REPEAT or not.
- INTERVAL rows. Displays the number of rows set by the **.INTERVAL** command (100 by default).
- STARTROW. Displays the starting row for fetches as set by the **.STARTROW** system command (1 by default).
- CONNECTIONS. Displays the database sessions (that is, connections) maintained by the current client. The server and database instance name are separated by the “at” symbol (**@**), and an asterisk (*****) indicates the current session. As well, the server software version and the transaction mode (SERIALIZABLE or EXCLUSIVE) are displayed.

.SYSTEM [*command*]

The .SYSTEM command passes control to the operating system for execution of the command specified by the *command* argument. This is equivalent to typing *.command*, since all commands prefixed by a period that are unrecognized by ssasql are passed to the operating system for execution.

.TIME

The .TIME command displays the current time. If executed subsequently during the same ssasql session, it displays the time elapsed since the previous invocation of the command.

.TITLES

The .TITLES command switches the display of column names in the headings of query results on and off. A session can be started with the .TITLES display turned off by including the **-t** flag in the ssasql invocation.

.WARNINGS

The .WARNINGS command switches the display of SQL warning messages on or off. A session can be started with SQL warning messages suppressed by including the **-w** flag in the ssasql invocation.

Display Formats

Limiting the Display of Results from a SELECT Statement

The display of results returned by a query can be limited to a specified number of records by including the LIMIT option with the SELECT statement. When the LIMIT keyword and a numeric parameter are appended to a SELECT statement, at most that number of records will be displayed from the actual result set.

Note: that, like the .MAXROWS system command, the LIMIT option does not affect the query results returned from the server; it serves only to limit what is displayed on the client side. Therefore, using the LIMIT option with CREATE TABLE...SELECT or INSERT...SELECT will not filter the data in any way.

The LIMIT syntax is the following:

```
<SELECT statement> LIMIT n ;
```

where *n* is a positive integer.

For example:

```
SELECT t1 FROM salestab LIMIT 10;
```

If the above query returns more than 10 records, only the first 10 records from the result set will be displayed. Otherwise, if the query returns 10 records or less, the LIMIT option is essentially ignored and the full result set is displayed.

Date Picture Display Format

The default format for the display of date values by Data Vault SQL clients (including ssasql) is as follows:

```
yyyy-mm-dd
```

That is, a ten-character string consisting of a four-digit year, a two-digit month number (1-12), and a two-digit day number (1-31), with components separated by hyphens (-). For example, for the date July 1, 2002, ssasql will display 2002-07-01.

The date picture can be customized by including a **DatePic** entry in the [CLIENT] section of the client-side nucleus.ini file.

Time Picture Display Format

The default format for the display of time values by Data Vault SQL clients (including ssasql) is as follows:

```
hh:mm:ss
```

That is, an eight-character string consisting of a two-digit hour (0-23), a two-digit minutes figure (0-59), and a two-digit seconds figure (0-31), with components separated by colons (:). For example, for 10 minutes past 2 pm, ssasql will display 14:10:00.

The time picture can be customized by including a **TimePic** entry in the [CLIENT] section of the client-side nucleus.ini file.

CHAPTER 16

Data Vault Audit Log

This chapter includes the following topics:

- Data Vault Audit Log Overview , 206
- Audit Log Process, 207
- Audit Log Configuration Process, 207
- Dynamic Data Masking Rules, 208
- Configuring the Data Vault SSA.ini File, 208
- Configuring the Data Vault Connection in the Management Console, 210
- Creating the Connection Rules, 211
- Creating the Security Rules, 213
- Generate the Audit File, 216
- Sample Audit Log, 217

Data Vault Audit Log Overview

You can create an audit log that records information about the tasks that a user performs in the Data Vault. The audit log contains information such as the user that issued the SQL query to the Data Vault and the exact SQL statement issued.

The audit log records information about user access to the Data Vault, so it can help you comply with organizational policies. For example, if you are a compliance officer, you might need to know when an application user queries Data Vault columns that contain sensitive information.

To create a Data Vault audit log, configure Informatica Dynamic Data Masking to connect to the Data Vault. Dynamic Data Masking is a data security product that operates as a proxy between the Data Vault and the applications or client tools that query the Data Vault. You configure connection rules and security rules in the Dynamic Data Masking Management Console to identify which incoming Data Vault SQL requests trigger the audit.

When an SQL request meets the criteria of the rules that you created, the Dynamic Data Masking Rule Engine rewrites the request to append an audit tag. Then the Dynamic Data Masking Server sends the rewritten request to the Data Vault. The Data Vault server recognizes the appended audit tag and writes the audit information to the audit log.

To extract the audit information and write it to a formatted CSV file, run the `EXTRACT FROM AUDIT` command.

Audit Log Process

When an application or client queries the Data Vault, the Dynamic Data Masking rules that you configure determine if Data Vault audits the request.

When an application or client sends an SQL request statement to the Data Vault, the Dynamic Data Masking Rule Engine applies the connection and security rules that you configure to the SQL request.

If the SQL request queries any columns that you configured to trigger the audit, the Dynamic Data Masking Rule Engine rewrites the SQL request to append an audit tag.

After the Rule Engine appends an audit tag to the SQL statement, the Dynamic Data Masking service sends the rewritten statement to the Data Vault. Data Vault recognizes the appended audit tags, and the Data Vault server writes the audit information to the audit log.

To extract the audit information from the audit log, use the Data Vault SQL Worksheet or a third-party SQL tool to run the `EXTRACT FROM AUDIT` command. The `EXTRACT FROM AUDIT` command extracts the audit information that you specify in the command and writes the information to a formatted CSV file.

The following figure shows the audit log components:

Audit Log Configuration Process

To create an audit log, first configure the Data Vault `ssa.ini` file. Then, in the Dynamic Data Masking Management Console, configure the Data Vault to connect to Dynamic Data Masking. Last, create connection rules and security rules in the Management Console.

To configure the audit log, perform the following tasks:

1. Configure the Data Vault `ssa.ini` file. The audit log parameters turn the audit log on and off, and control how the audit log files grow. To limit the size of the audit log files, configure either the AUDITLOGLIMIT_FILESIZE or AUDITLOGLIMIT_TIME property. To set the maximum number of audit log files that Data Vault retains, configure the AUDITLOGLIMIT_MAXFILES property.
2. Configure the Data Vault connection in the Management Console. First, add the Dynamic Data Masking service for Data Vault. Then enter the Data Vault connection details to create a connection to the Data Vault.
3. To process incoming connection requests from the application or client that issues SQL requests to the Data Vault, create connection rules. The first connection rule that you create identifies incoming connections and routes them to the Data Vault. The second connection rule that you create sends requests to the security rule set.
4. After you create the connection rules, create a security rule set and security rules. The first security rule that you create logs whenever a connection rule sends an incoming SQL request to the security rule set. The second security rule that you create identifies the columns that you want to trigger the audit. If an incoming SQL request queries any of the columns that you configured to trigger the audit, the rule rewrites the SQL request statement to append an audit tag.

Dynamic Data Masking Rules

When you configure the audit log, you create both connection rules and security rules in the Management Console.

A connection rule defines the connection criteria that the Dynamic Data Masking Rule Engine uses to identify a connection and the target database. Connection rules use a matcher and a rule action to identify and route a connection. The matcher defines the criteria that the Rule Engine uses to identify a match. The rule action determines how the Rule Engine processes the matched connection.

A security rule defines the criteria that the Rule Engine uses to parse and rewrite an SQL request. Security rules consist of a matcher, a rule action, and a processing action. The Rule Engine applies security rules to incoming SQL statements to identify a match to the criteria that you define. If a match occurs, the Rule Engine applies the security action to the SQL statement.

For more information about Dynamic Data Masking rules, see the *Informatica Dynamic Data Masking User Guide*.

Create the following connection and security rules to identify, route, and rewrite incoming SQL requests to the Data Vault:

Connection rule folder

Connection rule that routes all incoming connections to the rules in the folder. The folder contains the switch to database and use rule set rules.

Switch to database rule

Connection rule that routes incoming connections to the Data Vault database.

Use rule set rule

Connection rule that sends incoming Data Vault SQL requests to the security rule set that contains the log and rewrite rules.

Log rule

Security rule that logs whenever an incoming SQL request is sent to the security rule set. The Rule Engine writes this information to the `rule.log` file.

Rewrite rule

Security rule that identifies the columns that you want to trigger the audit. If these columns are included in an SQL request sent to the Data Vault, the rewrite rule also rewrites the SQL statement to append an audit tag.

Configuring the Data Vault SSA.ini File

To enable audit logs, configure the Data Vault `ssa.ini` file. The audit log properties turn the audit log on or off and allow you to specify the directory where Data Vault stores the audit log files. The properties also control how the audit logs grow.

Configure the following properties in the SERVER section of the Data Vault `ssa.ini` file:

AUDITLOG=

Enables or disables audit logging. Valid inputs are ON and OFF.

AUDITLOGDIR=

The location of the audit logs that the Data Vault server creates. Enter the file path of the directory where you want to save the audit logs.

For example, `AUDITLOGDIR=C:\ILM-IDV\fas_logs\`

AUDITLOGLIMIT_FILESIZE=

Sets the maximum size in bytes for the audit log. When the audit log reaches this size, one of two events occurs. New log information might replace the oldest log entries, keeping the audit log below or at the specified maximum size. Or, if you set AUDITLOGLIMIT_MAXFILES to a value greater than 1, the Data Vault server creates another audit log to store the latest log information. If the Data Vault server creates another audit log, it leaves the previous log intact.

If you set the AUDITLOGLIMIT_FILESIZE parameter to 0, Data Vault uses the AUDITLOGLIMIT_TIME parameter instead. If the AUDITLOGLIMIT_TIME parameter is also missing, Data Vault uses the default value for AUDITLOGLIMIT_FILESIZE to limit the log.

The AUDITLOGLIMIT_FILESIZE parameter takes precedence over the AUDITLOGLIMIT_TIME parameter. If you enter a value for both parameters in the `ssa.ini` file, the Data Vault server limits the audit log by size instead of time period.

Default is 1048576.

This property is optional.

AUDITLOGLIMIT_TIME=

Sets a time limit on the audit log. The audit log time limit determines how long the Data Vault server writes to an audit log. When the time interval between the first and latest message exceeds the time limit, the Data Vault server creates another audit log. The Data Vault server removes previous audit logs based on the maximum files setting, AUDITLOGLIMIT_MAXFILES.

Specify one of the following values for the AUDITLOGLIMIT_TIME parameter:

n MINUTE or *n* MINUTES

n HOUR or *n* HOURS

n DAY or *n* DAYS

n MONTH or *n* MONTHS

If you do not specify the unit of time, Data Vault uses DAYS.

The AUDITLOGLIMIT_TIME and AUDITLOGLIMIT_FILESIZE parameters are mutually exclusive. Both parameters set limits on the audit log, but AUDITLOGLIMIT_TIME constrains by time, while AUDITLOGLIMIT_FILESIZE constrains by file size. If you configure both parameters in the `ssa.ini` file, the AUDITLOGLIMIT_FILESIZE parameter takes precedence.

This property is optional.

AUDITLOGLIMIT_MAXFILES=

Specifies the maximum number of audit logs that Data Vault retains.

After an audit log reaches its configured size or time limit, the Data Vault server creates another audit log to store the latest messages. Data Vault does not delete the old audit log. Data Vault retains at least two audit logs, the current audit log and the previous one. Data Vault might delete earlier logs based on the value for AUDITLOGLIMIT_MAXFILES.

If you enter a value of 0 for the AUDITLOGLIMIT_MAXFILES parameter, Data Vault sets the maximum number of log files to unlimited. If you enter a value of 1, Data Vault uses the default value instead.

Default is 2.

Configuring the Data Vault Connection in the Management Console

In the Dynamic Data Masking Management Console, configure a connection to the Data Vault.

To create a connection to the Data Vault, add the Dynamic Data Masking service for Data Vault. Then define the Data Vault database properties.

For more information about Dynamic Data Masking services, see the *Informatica Dynamic Data Masking Administrator Guide*.

Step 1. Add the Dynamic Data Masking Service for Data Vault

Add the Dynamic Data Masking service for Data Vault in the Management Console.

1. In the Management Console, right-click the Dynamic Data Masking Server node and select **Add DDM Services**.

   The **Add DDM Services** window appears.
2. Select **DDM for FAS** and click **OK**.
3. In the Management Console, right-click the DDM for FAS service and select **Edit**.

   The **Edit** window appears.
4. Add an unused port number. Enter 8500 to use the default Data Vault listener port. If you use the default Data Vault listener port, you do not have to configure client connections.
5. Click **OK**.

Step 2. Create a Connection to the Data Vault

To create a connection to the Data Vault, add the Data Vault and define the database properties.

Verify that no firewall prohibits the Dynamic Data Masking Server from connecting to the database server.

1. In the Management Console, right-click the root domain node and then **Add Database**.

   The **Add Database** window appears.
2. Select the **FAS** database type.
3. Define the following properties for the Data Vault:

   DDM Database Name

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

   Server Address

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

Server Port

TCP/IP listener port for the Data Vault. Informatica recommends that you do not use the default Data Vault listener port. If you enter the default Data Vault listener port, clients might bypass the Dynamic Data Masking Server.

FAS Database Name

Database name for the Data Vault.

DBA Username

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

DBA Password

Password for database user.

4. Click **Test Connection** to validate the connection to the database.
5. Click **OK**.

   The Data Vault database appears in the Management Console tree.

Creating the Connection Rules

The following information is an example of a simple connection rules configuration.

Create a connection rule folder and insert two connection rules. The first rule that you create identifies incoming connections and routes the connections to the Data Vault database. The second rule that you create directs the incoming requests to the security rule set that you create before you create the security rules.

Step 1. Create the Connection Rule Folder

Create a connection rule folder to contain the connection rules that you create in the next steps.

1. In the Management Console, right-click the DDM for FAS service and select **Connection Rules**.

   The **Rule Editor** window appears.
2. In the **Rule Editor** window, right-click the **DDM for FAS** service and select **Append Rule**.

   The **Append Rule** window appears.
3. Enter a name for the connection rule folder in the **Rule Name** box. For example, IDV Connection Rules.
4. Define the following parameters:

Parameter	Description
Matcher	Defines the criteria that the Rule Engine uses to identify a match. Select **All Incoming Connections**. The All Incoming Connections matcher applies the matching action to all incoming connections.
Action	Defines the action that the Rule Engine applies to the request. Select **Folder**. The Folder action creates a folder and processes the connection through the contents of the folder until a processing action stops the Rule Engine.

Parameter	Description
Processing Action	Defines the action that the Rule Engine applies to the request after the Rule Engine applies the rule. Select **Stop if Applied**. The Rule Engine will not continue to the next rule in the rule tree.

5. Click **OK**.

   The connection rule folder appears in the rule tree under the root rule tree node.

6. To save the folder, select **File** > **Update Rules**.

Step 2. Create a Rule to Switch to the Data Vault Database

To create a rule that routes incoming connections to the Data Vault database, give the rule a name and define the rule parameters.

1. Right-click the connection rule folder that you created in the previous step and select **Append Rule**.

   The **Append Rule** window appears.

2. Enter a rule name, such as Switch to Data Vault Database.
3. Define the following parameters:

Parameter	Description
Matcher	Defines the criteria that the Rule Engine uses to identify a match. Select **All Incoming Connections**. The all Incoming Connections matcher applies the matching action to all incoming connections.
Action	Defines the action that the Rule Engine applies to the request. Select **Switch to Database**. The Switch to Database action forwards incoming connections to the database that you specify.
Database	Enter the name of the Data Vault database that you defined.
Processing Action	Defines the action that the Rule Engine applies to the request after the Rule Engine applies the rule. Select **Continue**. The Rule Engine continues to the next rule in the rule tree.

4. Click **OK.**

   The rule appears under the DDM for FAS service in the **Rule Editor** window.

5. Select **File** > **Update Rules**.

Step 3. Create a Rule that Applies the Security Rule Set

Create a rule that applies a specified security rule set to the incoming SQL request statement. You must name the security rule set in this step, but you create the security rule set in the next step.

1. In the **Rule Editor** window, right-click the connection rule folder and select **Append rule**.

   The **Append Rule** window appears.

2. Enter a rule name, such as Apply Security Rule Set.

3. Define the following parameters:

Parameter	Description
Matcher	Defines the criteria that the Rule Engine uses to identify a match. Select **All Incoming Connections**. The All Incoming Connections matcher applies the rule action to all incoming connections.
Action	Defines the action that the Rule Engine applies to the request. Select **Use Rule Set**. The Use Rule Set action applies the specified security rule set to the SQL statement request.
Rule Set Name	Enter a name for the rule set. For example, Data Vault Rule Set.
Processing Action	Defines the action that the Rule Engine applies to the request after the Rule Engine applies the rule. Select **Stop if Applied**. The Rule Engine will not continue to the next rule in the rule tree.

4. Click **OK.**

   The rule appears under the DDM for FAS service in the rule tree under the root rule tree node.

5. Select **File** > **Update Rules**.

Creating the Security Rules

The following information is an example of a simple security rules configuration.

Create a security rule set so that you can append the two security rules. The security rules that you create log, identify, and rewrite SQL requests that an application or a client issues to the Data Vault.

Create a rule that logs every incoming request to the security rule set. When the rule engine applies this rule, the resulting information appears in the Dynamic Data Masking `rule.log` file. Use the rule to identify incoming requests that cause the audit to fail. For more information about the `rule.log` file, see the *Informatica Dynamic Data Masking Administrator Guide*.

Create a rule that identifies specific strings of text in the SQL requests that an application or a client issues to the Data Vault. Then, the Rule Engine rewrites the SQL statement to append the `AUDIT USING` and `FROM RESULTSET` tags.

After the Dynamic Data Masking Rule Engine identifies a match and rewrites the SQL statement to append the `AUDIT USING` and `FROM RESULTSET` tags, it sends the rewritten request to the Data Vault. The Data Vault server recognizes the `AUDIT USING` tag and writes the results of the audit to the audit log file. The audit log file is located in the directory that you configured in the `SSA.ini` file for the **AUDITLOGDIR** property.

Step 1. Create the Security Rule Set

To create the security rule set, enter the name that you previously defined for the set.

1. In the Management Console, right-click a domain node and select **Add Rule Set**.

   The **Add Rule Set** window appears.

2. Enter the name that you defined for the security rule set in the previous step and click **OK**.

   The security rule set appears under the domain node in the Management Console.

Step 2. Create a Rule that Logs Incoming Requests to the Security Rule Set

To create a rule that logs every incoming request to the security rule set, give the rule a name and define the parameters.

1. In the Management Console, right-click the security rule set and select **Security Rule Set**.

   The **Rule Editor** window appears.
2. In the Rule Editor window, right-click the security rule set and select **Append Rule**.

   The **Append Rule** window appears.
3. Enter a name for the rule. For example, Log All.
4. Define the following parameters:

Parameter	Description
Matcher	Defines the portion of the SQL request that the Rule Engine must match to apply an action. Select **Any**. The Any matcher applies the rule action to all incoming SQL statements.
Action	Defines the action the Rule Engine applies to an SQL statement. Select **Nothing**. The Nothing rule action does not apply an operation to the processed SQL request.
Processing Action	Defines how the Rule Engine processes the connection after the Rule Engine applies the security rule action. Select **Continue**. The Rule Engine continues to the next rule in the rule tree.

5. Select the check box next to "Log when rule is applied."
6. Click **OK**.

   The rule appears under the rule set in the **Rule Editor** window.

Step 3. Create a Rule to Rewrite SQL Request Statements

Create a rule that identifies specific strings of text in an SQL request statement and then rewrites the SQL request to append an audit tag.

1. In the **Rule Editor** window, right-click the security rule set and select **Append rule**.

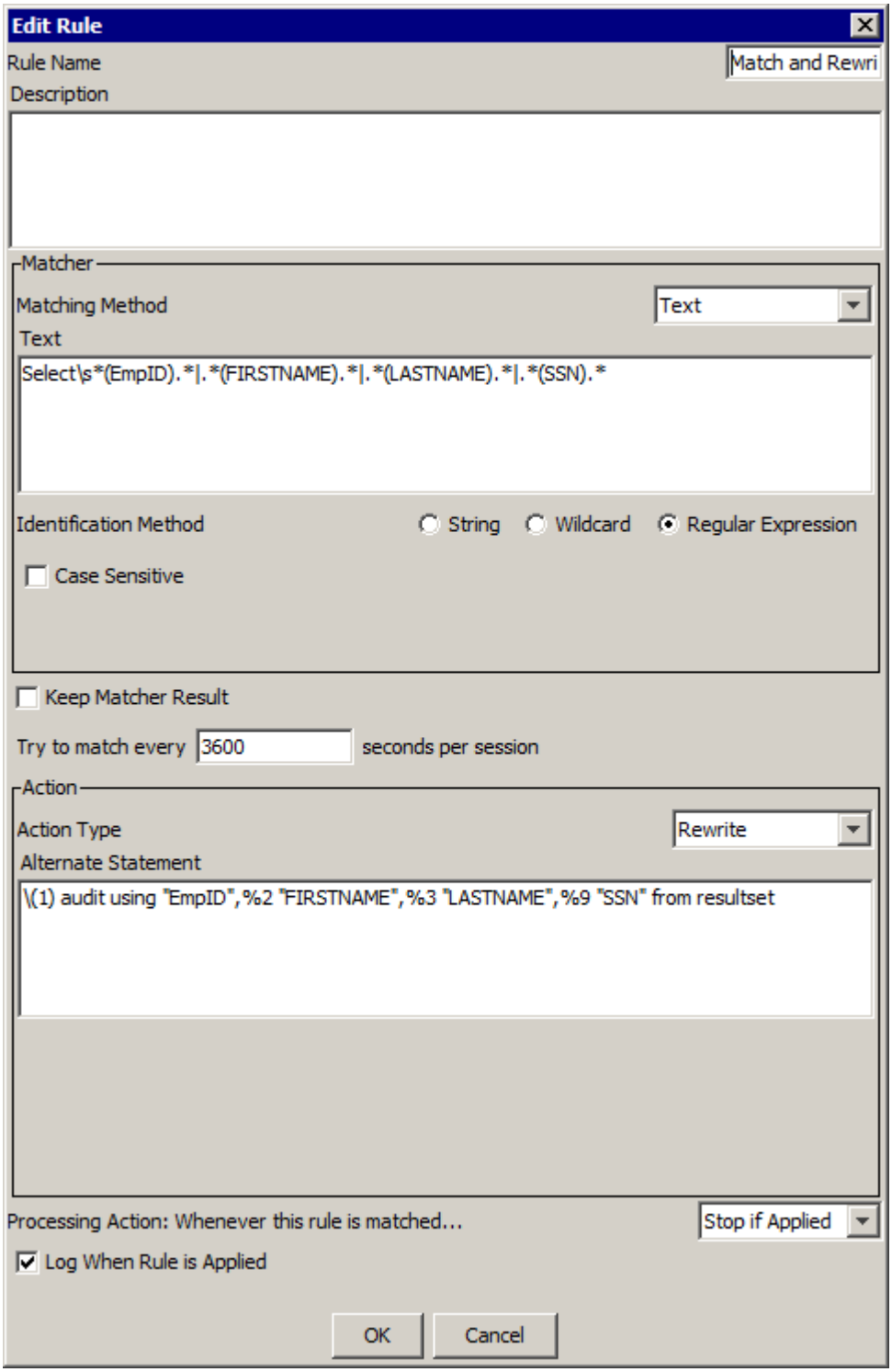
   The **Append Rule** window appears.
2. Enter a name for the rule, such as Match and Rewrite.
3. To configure the matcher, select **Text** from the menu next to Matching Method.

   The text matcher identifies a specific string of text within an SQL request sent to the Data Vault.
4. In the text box, enter a statement in a regular expression that identifies the archived database columns that you want to trigger the audit functionality when the columns are included in an SQL request statement issued to the Data Vault.

   For example, if you want Data Vault to audit all SQL request statements that query the columns named EmpID, FIRSTNAME, LASTNAME, and SSN, enter the following statement:

   ```
   Select\s*(EmpID).*|.*(FIRSTNAME).*|.*(LASTNAME).*|.*(SSN).*
   ```
5. Next to Identification Method, select **Regular Expression**.
6. To configure the Action, select **Rewrite** from the menu.
7. In the Alternate Statement text box, enter an SQL statement that includes the column names that you want to trigger the audit functionality. The statement must begin with "\(1)`audit using`" and end with "`from resultset`". The "%" sign followed by a number designates column position.

8. To configure the processing action, select **Stop if applied** from the menu.
9. Select the check box next to "Log When Rule Is Applied."



10. Click **OK.**

11. Click **File** > **Update Rules**.

Generate the Audit File

To extract the audit information from the Data Vault audit log and write the information to a CSV file, run the `EXTRACT FROM AUDIT` command.

In the Data Vault SQL Worksheet or a third-party SQL tool, run the following command: `EXTRACT FROM AUDIT <"predefined attribute">, <column name>, INTO '<file path>';`

Replace <predefined attribute> with the type of audit information that you want to extract from the audit log.

The following table describes valid inputs for the `EXTRACT FROM AUDIT` command:

Attribute	Description
CURRENT TIMESTAMP	Time when the SQL request statement was received by the Data Vault.
CURRENT USER	User name on the client connection that issued the SQL request statement to the Data Vault.
CURRENT SERVER	The host and port that the client server is running on.
CURRENT SQL	The SQL request statement issued to the Data Vault.
CURRENT SQLID	A string that contains a unique query identifier that is issued by the Data Vault server.
ROW TIMESTAMP	The time when a specific row is sent to the client.

Replace <column name> with the names of the audited columns that you want to extract from the audit log and write to the CSV file. These columns are the columns that you configured to trigger the audit when you created the rule to rewrite SQL requests.

Replace <file path> with the name of the directory where you want to create the CSV file in addition to the file name. For example, `C:\ILM-IDV\fas_logs\Audit1.log`

For example, the following query extracts information from the columns LASTNAME, FIRSTNAME, and SOCIAL, in addition to the time stamp and client user name:

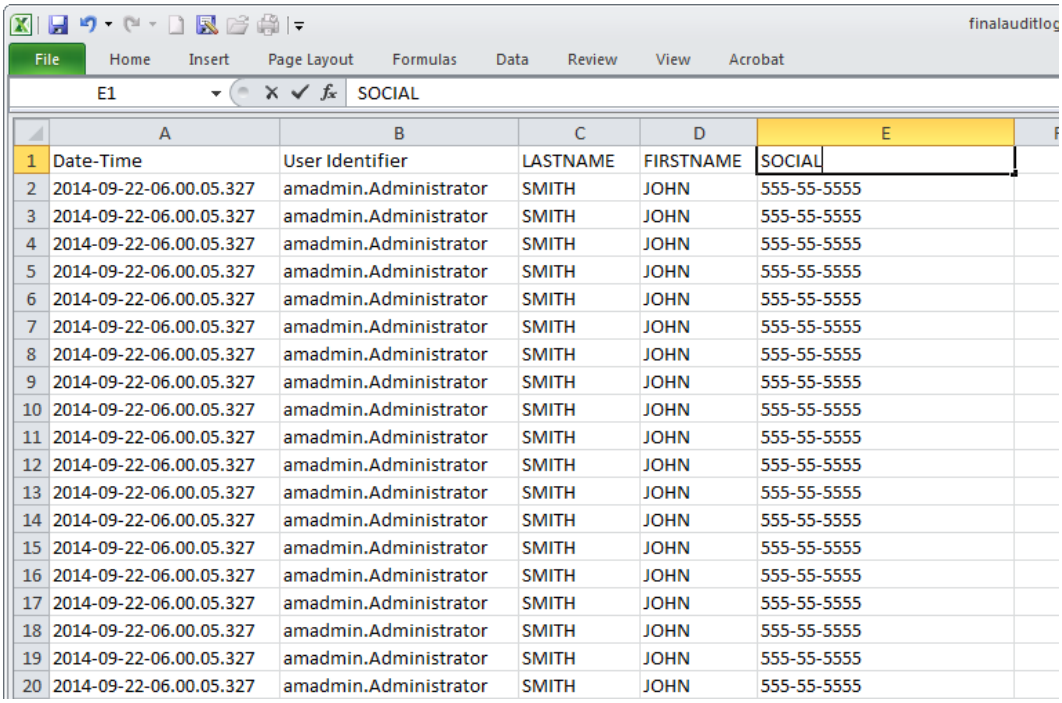
```
EXTRACT FROM AUDIT "CURRENT TIMESTAMP", "CURRENT USER", LASTNAME, FIRSTNAME, SOCIAL INTO 'C:
\ILM-IDV\fas_logs\Audit1.log';
```

When you run the `EXTRACT FROM AUDIT` command, the command extracts the information that you specify in the SQL statement and writes it to a formatted CSV file. The file is located in the directory that you specify in the command.

Sample Audit Log

The `EXTRACT FROM AUDIT` command extracts the audit information from the audit log and writes it to a formatted CSV file.

The following figure shows an example of the formatted CSV file opened in Microsoft Excel:



	A	B	C	D	E
1	Date-Time	User Identifier	LASTNAME	FIRSTNAME	SOCIAL
2	2014-09-22-06.00.05.327	amadmin.Administrator	SMITH	JOHN	555-55-5555
3	2014-09-22-06.00.05.327	amadmin.Administrator	SMITH	JOHN	555-55-5555
4	2014-09-22-06.00.05.327	amadmin.Administrator	SMITH	JOHN	555-55-5555
5	2014-09-22-06.00.05.327	amadmin.Administrator	SMITH	JOHN	555-55-5555
6	2014-09-22-06.00.05.327	amadmin.Administrator	SMITH	JOHN	555-55-5555
7	2014-09-22-06.00.05.327	amadmin.Administrator	SMITH	JOHN	555-55-5555
8	2014-09-22-06.00.05.327	amadmin.Administrator	SMITH	JOHN	555-55-5555
9	2014-09-22-06.00.05.327	amadmin.Administrator	SMITH	JOHN	555-55-5555
10	2014-09-22-06.00.05.327	amadmin.Administrator	SMITH	JOHN	555-55-5555
11	2014-09-22-06.00.05.327	amadmin.Administrator	SMITH	JOHN	555-55-5555
12	2014-09-22-06.00.05.327	amadmin.Administrator	SMITH	JOHN	555-55-5555
13	2014-09-22-06.00.05.327	amadmin.Administrator	SMITH	JOHN	555-55-5555
14	2014-09-22-06.00.05.327	amadmin.Administrator	SMITH	JOHN	555-55-5555
15	2014-09-22-06.00.05.327	amadmin.Administrator	SMITH	JOHN	555-55-5555
16	2014-09-22-06.00.05.327	amadmin.Administrator	SMITH	JOHN	555-55-5555
17	2014-09-22-06.00.05.327	amadmin.Administrator	SMITH	JOHN	555-55-5555
18	2014-09-22-06.00.05.327	amadmin.Administrator	SMITH	JOHN	555-55-5555
19	2014-09-22-06.00.05.327	amadmin.Administrator	SMITH	JOHN	555-55-5555
20	2014-09-22-06.00.05.327	amadmin.Administrator	SMITH	JOHN	555-55-5555

APPENDIX A

Sample Configuration Files

This appendix includes the following topics:

- ssa.ini, 218
- nucleus.ini, 219

ssa.ini

The following example shows the content of the ssa.ini file:

```
[AGENT]
PRIORITY=10
LOGDIR=C:\Informatica\ILM-DV\fas_agent_logs

[QUERY]
THREADS=2
MAXVMEM=4096
MEMORY=512
TEMPDIR=C:\Informatica\ILM-DV\temp
SHAREDIR=C:\Informatica\ILM-DV\share
INDEXDIR=C:\Informatica\ILM-DV\index

[COMMON]
CRYPTOLEVEL=0
TEMPDIR=C:\Informatica\ILM-DV\temp
SHAREDIR=C:\Informatica\ILM-DV\share
LOGDIR=C:\Informatica\ILM-DV\fas_logs

[EXPORT]
BUFFSIZE=512

[SERVICE]
SSAJOBLOG=C:\Informatica\ILM-DV\fas_logs
ADDRESS=
SSA_UID=dba
LOADBUFFER=32768
SSASERVICELOG=C:\Informatica\ILM-DV\fas_logs
JOBRETRY=0
LOADPARAM=j:2,k:2
SSA_PWD=B1E786394FB4762B9A83EE2329162A59
SSA_Connection=fas
SMTP=
SSA_Database=meta
EMAIL=0
TIMEOUT=15

[STARTER]
AGENT_CONTROL=1
AGENT_COUNT=2
VERBOSE=2
```

```
SERVER_CONTROL=1
AGENT_CMD=ssaagent
SERVER_CMD=ssaserver
#EXE0=ssaservice start
LOGDIR=C:\Informatica\ILM-DV\fas_logs

[META]
PWD=B1E786394FB4762B9A83EE2329162A59
STARTUPHOST=localhost
AUTOSTARTUP=1
UID=dba
Database=meta
STARTUPCOMMAND=fb_inet_server -a
Connection=meta

[SERVER]
THREADS=10
HOST=myhost
EXEPORT=8600
PORT=8500
ROWSETBUFFER=64
TEMPDIR=C:\Informatica\ILM-DV\temp
LOGDIR=C:\Informatica\ILM-DV\fas_logs
INDEXDIR=C:\Informatica\ILM-DV\index
```

nucleus.ini

The following example shows the content of the nucleus.ini file:

```
[CLIENT]
TimePic=hh:mm:ss
DatePic=yyyy-mm-dd

[CONNECTION meta]
Port=50501
Host=myhost
STACK=20971520
MaxUsers=157

[CONNECTION fas]
Port=8500
Host=myhost
```

INDEX

- (hyphen) 43
-o (ssasql flag) 193, 199
-p (ssasql flag) 199
-t (ssasql flag) 199
-w (ssasql flag) 199
.DATA (system command) 199
.EXIT (system command) 193, 199
.HELP (system command) 199
.HEX (system command) 199
.INDEX (system command) 199
.MAXLENGTH (system command) 199
.MAXROWS (system command) 199, 204
.MEASURE (system command) 199
.NULLS (system command) 199
.OUTPUT (system command) 199
.PROMPT (system command) 199
.QUIET (system command) 199
.RUN (system command) 199
.SESS (system command) 199
.SQL (system command) 199
.STARTROW (system command) 199
.SYSTEM (system command) 199
.TIME (system command) 199
.TITLES (system command) 199
.WARNINGS (system command) 199

A

ADMIN
  ssa.ini parameter 34
AGENT section
  ssa.ini 31
AGENT_CMD
  ssa.ini 19
AGENT_CONTROL
  ssa.ini 19
AGENT_COUNT
  ssa.ini 19
ALL PRIVILEGES clause 183, 187
Alter Index (ssasql) 98
AUTOSTARTUP
  ssa.ini parameter 32

B

BLOCKFETCHBUFFER
  ssa.ini parameter 25
Bourne shell, setting the NUCLEUS environment variable in 43
BUFFSIZE
  ssa.ini parameter 25
BULKLOADDIR
  ssa.ini parameter 25

C

C shell, setting the NUCLEUS environment variable in 43
case sensitivity
  ssasql commands 198
CASTING
  ssa.ini parameter 32
changing database sessions 196
CLIENT (nucleus.ini section) 204
client session, disconnecting a 193, 196, 199
client-side nucleus.ini file 204
columns
  domains, relationship with 183
  privileges 181, 183, 185
compacted table (SCT) 198
Compacted Table (SCT) 198
configuration file
  nucleus.ini 23, 219
  ssa.ini 23, 218
connecting to a database instance 193, 196
CONNECTION
  ssa.ini parameter 32
CONNECTION (nucleus.ini section) 198
connection name 198
CREATE AUTHORIZATION 188
Create Index (ssasql) 97
CSV 199

D

data files
  calculation of 106
  creation of 107
  orphaned 109
  registration 109
  repartitioning 105
  row count 105
data indexing
  create index command 97
  description 95
  drop index command 99
  example 96
  renew index command 98
  rules and guidelines 96
data repartitioning
  data file registration 109
  description 105
  example 110
  logs 110
  ssapart command 111
  use cases 105
Data Vault
  components 15
  Data Vault Administration Tool 16
  Data Vault Data Archive plug-in 16

Data Vault *(continued)*
Data Vault repository 15
Data Vault Service 15
load balancer 15
logs 174
ssa_starter 18
startup 18
Data Vault Administration Tool
Data Vault 16
Data Vault Agent
Data Vault Service 15
shutdown 22
startup 21
Data Vault Data Archive plug-in
Data Vault 16
Data Vault Loader
Data Vault Service 15
Data Vault repository
Data Vault 15
Data Vault Service
Data Vault 15
Data Vault Agent 15
Data Vault Loader 15
shutdown 22
startup 21
database
changing sessions 196
connecting to 193, 196
disconnecting from 193, 196, 199
instance 198
qualified object names 181
DATABASE
ssa.ini parameter 32
database instance 38
DatePic (nucleus.ini date picture string) 204
DBA privileges 181, 185, 187
default values (nucleus.ini file) 43
DELETE privileges 183
DELIMITER
ssa.ini parameter 25
disconnecting a client session 193, 196, 199
domains
privileges 181, 183, 185, 187
DROP 188
Drop Index (ssasql) 99

E

environment variable
SSA_INI_DIR 23
SSA_STARTER_USER 23
EOL
ssa.ini parameter 25
ERHF
ssa.ini parameter 25
EXE
ssa.ini 19
EXEHOST
ssa.ini parameter 31, 34
EXEPORT
ssa.ini parameter 25, 31, 34
.EXPORT...CSV (system command) 199
EXPORT (ssa.ini section) 25
EXPORT section
ssa.ini 25

H

HOST
ssa.ini parameter 25
Host (nucleus.ini parameter) 38
hyphen (-) 43

I

indexes
altering 98
creating 97
description 95
dropping 99
example 96
renewing 98
rules and guidelines 96
INSERT 183
INSERT privileges 181, 185
instance of a database 38
instance, database
name 198
IP address.
Host (nucleus.ini parameter). 38

K

keywords, SAND CDBMS Nearline SQL 181, 183, 185, 187
Korn shell, setting the NUCLEUS environment variable in 43

L

LIMIT (ssasql option to limit SELECT results) 204
load balancer
Data Vault 15
LOG
ssa.ini parameter 25
LOGAPPEND
ssa.ini parameter 31
LOGDIR
ssa.ini 19
ssa.ini parameter 25, 31
LOGINTIMEOUT
ssa.ini parameter 32
LOGLIMIT_FILESIZE
ssa.ini parameter 25
LOGLIMIT_MAXFILES
ssa.ini parameter 25
LOGLIMIT_TIME
ssa.ini parameter 25
logs
query statistics logs 174
data repartitioning 110
standard logs 174
trace logs 174
LOGVERBOSE
ssa.ini parameter 25

M

MaxQueries (nucleus.ini parameter) 198
MAXTEMPSIZE
ssa.ini parameter 25

MEMORY
ssa.ini parameter 24
META
ssa.ini 21
META section
ssa.ini 32
Metadata Repository 193, 198
metadata repository database
connecting to 193
MODE
ssa.ini parameter 25

N

names
connection 198
instance 198
user 198
nucleus.ini
configuration file 23, 219
nucleus.ini file
client-side 204
parameters
DatePic 204
Host 38
MaxQueries 198
Port 38
TimePic 204
sections
CLIENT 204
CONNECTION 198
server-side 198
NULL
ssa.ini parameter 25
null values 199

O

orphaned data files
registration 109
OWNER privileges 183
ownership privileges 181

P

performance tuning
queries 95, 105
picture string (nucleus.ini)
time 204
PORT
ssa.ini parameter 25
Port (nucleus.ini parameter) 38
PRIORITY
ssa.ini parameter 31
privileges
ALL PRIVILEGES clause 183, 187
DBA 181, 185, 187
DELETE 183
granting 181
INSERT 181, 185
OWNER 183
ownership 181
revoking 181
SELECT 181, 183, 185
UPDATE 181, 183, 185
privileges *(continued)*
USAGE 183, 187
WITH GRANT OPTION 181, 184
Public user 181, 183–185
PWD
ssa.ini parameter 32

Q

qualified object names 181
queries
performance tuning 95, 105
QUERY
ssa.ini parameter 24

R

Renew Index (ssasql) 98
REVOKE 181, 183
row count
data files 105, 106
ROWSETBUFFER
ssa.ini parameter 25

S

SAND CDBMS Nearline
SQL keywords 181, 183, 185, 187
schemas
granting privileges on 183
OWNER privilege 183
System schema 181, 183, 185, 187
SCT 198
sections, nucleus.ini
CLIENT 204
CONNECTION 198
SELECT (SQL command) 198, 199, 204
SELECT privileges 181, 183, 185
SERVER
ssa.ini parameter 25
SERVER_CMD
ssa.ini 19
SERVER_CONTROL
ssa.ini 19
server-side nucleus.ini file 198
SESS (Session) mode, ssasql
about 193, 199
SET TRANSACTION 188
setting the NUCLEUS environment variable
in Bourne shell 43
in C shell 43
in Korn shell 43
SHAREDIR
ssa.ini parameter 24, 25
shutdown
Data Vault Agent 22
Data Vault Service 22
SHUTDOWN 188
SQL commands
CREATE AUTHORIZATION 188
DROP 188
GRANT 181
INSERT 183
REVOKE 181, 183
SELECT 198, 199, 204

SQL commands *(continued)*
SET TRANSACTION 188
SHUTDOWN 188
SQL keywords, SAND CDBMS Nearline 181, 183, 185, 187
SQL mode, ssasql 193, 198, 199
SSA_INI_DIR
environment variable 23
ssa_starter
Data Vault 18
options 19
syntax 19
SSA_STARTER_USER
environment variable 23
ssa.ini
AGENT_CMD 19
AGENT_CONTROL 19
AGENT_COUNT 19
configuration file 23, 218
EXE 19
LOGDIR 19
META 21
SERVER_CMD 19
SERVER_CONTROL 19
STARTER 19
USER 19
VERBOSE 19
ssa.ini parameter
MEMORY 24
SHAREDIR 24
TEMPDIR 24
THREADS 24
VFS 24
ssaagent
startup 21
ssapart
data repartitioning 111
ssaserver
startup 21
ssasql
Alter Index 98
Create Index 97
Drop Index 99
establishing a database session 193
Renew Index 98
SQL mode 198
ssasql (SAND CDBMS Nearline Interactive SQL client program)
batch file, executing a 193
changing database sessions 196
column name display, turning off 199
command prompt display, turning off 199
disconnecting a database session 193, 196, 199
establishing a database session 193, 196
flags
-o 193, 199
-p 199
-t 199
-w 199
limiting SELECT results 204
SESS (Session) mode
about 193, 199
session logging 199
SQL mode 193, 199
system commands
.DATA 199
.EXIT 193, 199
.EXPORT...CSV 199
.HELP 199
.HEX 199
ssasql (SAND CDBMS Nearline Interactive SQL client program) *(continued)*
system commands *(continued)*
.INDEX 199
.MAXLENGTH 199
.MAXROWS 199, 204
.MEASURE 199
.NULLS 199
.OUTPUT 199
.PROMPT 199
.QUIET 199
.RUN 199
.SESS 199
.SQL 199
.STARTROW 199
.STATS 199
.SYSTEM 199
.TIME 199
.TITLES 199
.WARNINGS 199
warning message display, turning off 199
STARTER
ssa.ini 19
startup
Data Vault 18
STARTUPCOMMAND
ssa.ini parameter 32
STARTUPHOST
ssa.ini parameter 32
.STATS (system command) 199
switching between database sessions 196
syntax
ssa_starter 19
system commands, ssasql
ssasql (SAND CDBMS Nearline Interactive SQL client program) 193
system commands 193
System schema 181, 183, 185, 187
System tables
about 181, 183, 185, 187

T

tables
data file registration 109
indexing 96
privileges 185
repartitioning 105
System tables, about 185
TEMPDIR
ssa.ini parameter 24, 25
temporary files
data repartitioning 107
THREADS
ssa.ini parameter 24, 25
TimePic (nucleus.ini time picture string) 204
TRACELOG
ssa.ini parameter 25
TRACELOGLIMIT_FILESIZE
ssa.ini parameter 25
TRACELOGLIMIT_MAXFILES
ssa.ini parameter 25
TRACELOGLIMIT_TIME
ssa.ini parameter 25
TRUNCATION
ssa.ini parameter 32

U

UID
  ssa.ini parameter 32
UPDATE privileges 181, 183, 185
USAGE privileges 183, 187
user
  name 198
USER
  ssa.ini 19
user authorizations 181, 183, 185
user Public 181, 183–185

V

VERBOSE
  ssa.ini 19
VERBOSE *(continued)*
  ssa.ini parameter 25
VFS
  ssa.ini parameter 24
views
  privileges 181, 185

W

WITH GRANT OPTION 181, 184